

# SIEMENS

**SINUMERIK  
840D sl/840Di sl/840D/840Di/810D**

**Commissioning CNC Part 2 (HMI)**

**Commissioning Manual**

**Expand User  
Interface** **BE1**

**Online Help** **HE1**

**Installing HMI  
Embedded** **IM2**

**Commissioning HMI  
Advanced** **IM4**

**Creating Foreign  
Language Texts  
with Windows 2000 /  
XP** **TX2**

# SINUMERIK® Documentation

## Printing history

Brief details of this edition and previous editions are listed below.

The status of each version is indicated by the code in the "Remarks" columns.

*Status code in the "Remarks" column:*

- A ....** New documentation.
- B ....** Unrevised reprint with new order number
- C ....** Revised edition with new status

<b>Edition</b>	<b>Order No.</b>	<b>Comments</b>
12.98	6FC5297-5AE20-0BP0	A
08.99	6FC5297-5AE20-0BP1	C
02.01	6FC5297-6AE20-0BP0	C
11.01	6FC5297-6AE20-0BP1	C
11.02	6FC5297-6AE20-0BP2	C
03.04	6FC5297-6AE20-0BP3	C
08.05	6FC5397-3AP10-0BA0	C
01/2006	6FC5397-3AP10-1BA0	C

## Registered Trademarks

All designations with the trademark symbol ® are registered trademarks of Siemens AG. Other designations in this documentation may be trademarks whose use by third parties for their own purposes may infringe the rights of the owner.

## Liability disclaimer

We have checked that the contents of this document correspond to the hardware and software described. Nonetheless, differences might exist and therefore we cannot guarantee that they are completely identical. The information contained in this document is, however, reviewed regularly and any necessary changes will be included in the next edition.

# Preface

## SINUMERIK® Documentation

The SINUMERIK documentation is organized in 3 parts:

- General documentation
- User documentation
- Manufacturer/service documentation

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

<http://www.siemens.com/motioncontrol>

Follow the menu items "Support" → "Technical Documentation" → "Overview of Publications".

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

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

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

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

## Target group

This documentation is intended for service personnel.

## Benefits

Based on the commissioning manual, the intended target group can test and commission the system correctly without any danger.

## Standard version

The functionality of the standard version is described in this commissioning manual. Extensions or changes made by the machine tool manufacturer are documented by the machine tool manufacturer.

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

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

## Technical Support

If you have any questions, please contact our hotline:

### Europe and Africa time zone:

A&D Technical Support

Phone: +49 (0) 180 / 5050 - 222

Fax: +49 (0) 180 / 5050 - 223

Internet: <http://www.siemens.com/automation/support-request>

E-mail: <mailto:adsupport@siemens.com>

### Asia and Australia time zone:

A&D Technical Support

Phone: +86 1064 719 990

Fax: +86 1064 747 474

Internet: <http://www.siemens.com/automation/support-request>

E-mail: <mailto:adsupport@siemens.com>

### America time zone:

A&D Technical Support

Phone: +1 423 262 2522

Fax: +1 423 262 2289

Internet: <http://www.siemens.com/automation/support-request>

E-mail: <mailto:adsupport@siemens.com>

---

### Note

Country telephone numbers for technical support are provided under the following Internet address:

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

---

## Questions about the manual

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

Fax: +49 (0) 9131 / 98 - 63315

E-mail: <mailto:motioncontrol.docu@siemens.com>

Fax form: See the reply form at the end of the brochure.

## SINUMERIK Internet address

<http://www.siemens.com/sinumerik>

## EC declaration of conformity

The EC Declaration of Conformity for the EMC Directive can be found/obtained

- in the internet:  
<http://www.ad.siemens.com/csinfo>  
under product/order no. 15257461
- at the relevant branch office of the A&D MC group of Siemens AG.

## Safety Instructions

This Manual contains information which you should carefully observe to ensure your own personal safety and the prevention of material damage.

The notices referring to your personal safety are highlighted in the manual by a safety alert symbol, notices referring to property damage only, have no safety alert symbol. The warnings appear in decreasing order of risk as given below.



### 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 warning triangle indicates that minor personal injury can result if proper precautions are not taken.

---

### Caution

Without a warning triangle this indicates that property damage may occur 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 several hazards of different degrees are present, the warning notice representing the highest degree of danger will be used. A warning notice accompanied by a safety alert symbol indicating a risk of bodily injury can also indicate a risk of property damage.

## Qualified personnel

The associated device/system may only be set up and operated using this documentation. The device/system must only be commissioned and operated by **qualified personnel**. Qualified persons are defined as persons who are authorized to commission, ground, and tag equipment, systems, and circuits in accordance with established safety standards.

## Correct usage

Please note the following:



---

### Warning

The unit may be used only for the applications described in the catalog or the technical description, and only in combination with the equipment, components and devices of other manufacturers where recommended or permitted by Siemens. Correct transport, storage, installation and assembly, as well as careful operation and maintenance, are required to ensure that the product operates safely and without faults.

---

## Notes

The following notes with a special meaning are used in the documentation:

---

### Note

This symbol always appears in this documentation where further, explanatory information is provided.

---



---

### Ordering Data Option

In this documentation you will find the symbol shown on the left with a reference to an ordering data option. The described function is only executable on the control if the control has the designated option.

---

## When changes take effect

After data is modified (e.g. machine data) you must also observe when the data goes into effect (e.g. after Power On or immediately). The time is therefore always specified as well.



# SINUMERIK 840D sl/840Di sl SINUMERIK 840D sl/840Di sl/840D/840Di/810D

## Commissioning CNC Part 2 (HMI)

### Expand User Interface (BE1)

#### Valid for

##### *PLC*

SINUMERIK 840D sl/840DE sl  
SINUMERIK 840Di sl / 840DiE sl  
SINUMERIK 840D powerline/840DE powerline  
SINUMERIK 840Di powerline/840DiE powerline  
SINUMERIK 810D powerline/810DE powerline

<i>Software</i>	<i>Software Version</i>
HMI Advanced	7.2
HMI-Embedded	7.1
ShopMill	7.1
ShopTurn	7.1





## Contents

<b>1 Introduction</b> .....	<b>BE1/1-7</b>
1.1 Objective .....	BE1/1-8
1.2 Performance range and mode of operation.....	BE1/1-8
1.3 Boundary conditions for use.....	BE1/1-9
1.4 Syntax rules in this documentation .....	BE1/1-10
<b>2 First Steps</b> .....	<b>BE1/2-13</b>
2.1 Fundamentals of Configuration .....	BE1/2-14
2.1.1 Configuration and start files .....	BE1/2-15
2.1.2 Search sequence for configuration files .....	BE1/2-15
2.1.3 Calling your own screen forms from PLC.....	BE1/2-16
2.2 Creating configuration file as ASCII file.....	BE1/2-16
<b>3 Syntax</b> .....	<b>BE1/3-17</b>
3.1 Structure of a configuration file .....	BE1/3-19
3.2 Defining a start softkey .....	BE1/3-20
3.3 Syntax of screen form descriptions .....	BE1/3-21
3.3.1 Defining a screen form .....	BE1/3-21
3.3.2 Screen form properties .....	BE1/3-24
3.3.3 Screen form elements.....	BE1/3-27
3.3.4 Multi-column screen forms .....	BE1/3-34
3.3.5 Differences in screen form appearance .....	BE1/3-35
3.3.6 Integer expansions of the variable parameter group 1).....	BE1/3-37
3.3.7 Binary, hexadecimal and exponential notation.....	BE1/3-39
3.3.8 More information about variable parameters.....	BE1/3-40
3.3.9 Variable properties .....	BE1/3-43
3.3.10 Display images.....	BE1/3-45
3.3.11 Arrays.....	BE1/3-49
3.3.12 Table element (grid).....	BE1/3-54
3.4 Syntax of softkey descriptions.....	BE1/3-61
3.5 Methods .....	BE1/3-65
3.5.1 PRESS method .....	BE1/3-66
3.5.2 LOAD method .....	BE1/3-67
3.5.3 UNLOAD method .....	BE1/3-67
3.5.4 CHANGE method.....	BE1/3-67
3.5.5 FOCUS method .....	BE1/3-68
3.5.6 OUTPUT method .....	BE1/3-69
3.5.7 Version management in conjunction with OUTPUT blocks .....	BE1/3-69
3.6 Functions .....	BE1/3-71
3.6.1 Screen form or softkey change .....	BE1/3-71
3.6.2 PLC/NC variable, write/read MD .....	BE1/3-75
3.6.3 General screen form variables .....	BE1/3-78
3.6.4 General functions.....	BE1/3-81
3.6.5 Subprograms .....	BE1/3-84

3.6.6	File functions.....	BE1/3-87
3.6.7	NC Functions (PI services).....	BE1/3-93
3.6.8	External functions (only HMI Advanced) .....	BE1/3-93
3.6.9	Operators and functions.....	BE1/3-95
3.6.10	Use of strings .....	BE1/3-98
3.6.11	Functions for start softkeys .....	BE1/3-103
3.7	Error handling, log book .....	BE1/3-103
<b>4</b>	<b>Programming Support.....</b>	<b>BE1/4-105</b>
4.1	What does the programming support do? .....	BE1/4-106
4.2	Creating new screen forms .....	BE1/4-106
4.3	Special functions .....	BE1/4-107
4.4	General cycle support .....	BE1/4-109
4.5	Activating a dialog screen from the NC program.....	BE1/4-111
4.5.1	Structure of instructions .....	BE1/4-112
4.5.2	Instruction examples .....	BE1/4-113
4.5.3	Example 1 MMC command without acknowledgment.....	BE1/4-116
4.5.4	Example 2 Hold time and optional text variables .....	BE1/4-117
4.5.5	Example 3 MMC command with synchronous acknowledgment mode .....	BE1/4-118
4.5.6	Example 4 Positioning of input/output field .....	BE1/4-119
4.5.7	Example 5 Displaying graphics in the dialog screen .....	BE1/4-121
4.5.8	Example 7 Displaying OPI variables .....	BE1/4-122
4.5.9	Example 8 Asynchronous acknowledgment mode with softkeys .....	BE1/4-123
<b>5</b>	<b>Configuring OP Hotkeys and PLC Keys .....</b>	<b>BE1/5-127</b>
5.1	Introduction .....	BE1/5-128
5.1.1	OP hotkeys .....	BE1/5-129
5.1.2	Assigning functions to keys.....	BE1/5-130
5.1.3	Configuring the M key and Menu Select as HK7, HK8.....	BE1/5-134
5.1.4	Expansions and special cases .....	BE1/5-135
5.2	Selecting displays created with 'Expand user interface' .....	BE1/5-136
5.2.1	Assigning INI files to operating areas.....	BE1/5-136
5.2.2	Configuring 'Expand user interface' functions .....	BE1/5-137
5.3	PLC interface .....	BE1/5-138
5.3.1	Description of the interface .....	BE1/5-138
5.3.2	Requirements for the PLC programm .....	BE1/5-138
5.3.3	Screen selection from PLC .....	BE1/5-139
5.4	Expanded functions for PLC keys .....	BE1/5-142
5.5	List of selectable states in HMI Advanced.....	BE1/5-143
5.6	List of selectable states in HMI Embedded sl.....	BE1/5-144
5.7	List of selectable states in ShopMill on NCU.....	BE1/5-145
5.8	List of selectable states in ShopTurn on NCU.....	BE1/5-147
5.9	Hotkey assignment on the MFII keyboard.....	BE1/5-150

5.10	Examples .....	BE1/5-151
5.11	Configuration overview .....	BE1/5-151
5.12	Terms.....	BE1/5-153
<b>6</b>	<b>Custom Operating Area.....</b>	<b>BE1/6-155</b>
6.1	Introduction .....	BE1/6-156
6.1.1	Delivery .....	BE1/6-156
6.1.2	Activation during startup.....	BE1/6-157
6.1.3	Activation by keys .....	BE1/6-157
6.1.4	Characteristics of the Custom area .....	BE1/6-157
6.1.5	Integration of the Custom operating area.....	BE1/6-158
6.2	Layout and functions .....	BE1/6-159
6.2.1	Starting screen form layout .....	BE1/6-159
6.3	Behavior during operating area change .....	BE1/6-160
<b>7</b>	<b>Configuring and Working Environment .....</b>	<b>BE1/7-161</b>
7.1	General information about the configuring environment .....	BE1/7-162
7.2	COMMON.COM control file.....	BE1/7-163
7.2.1	Control file function .....	BE1/7-163
7.2.2	Storage location of COMMON.COM .....	BE1/7-164
7.2.3	Configuration files .....	BE1/7-164
7.2.4	Structure of the COMMON.COM control file .....	BE1/7-165
7.3	Storing the configuration files.....	BE1/7-172
7.3.1	HMI Embedded sl .....	BE1/7-172
7.3.2	HMI Advanced .....	BE1/7-173
7.4	Common HW platform for both HMI systems.....	BE1/7-174
7.4.1	General search strategy.....	BE1/7-175
7.4.2	Search strategy for COMMON.COM.....	BE1/7-177
7.4.3	Search sequence for display images .....	BE1/7-177
<b>8</b>	<b>Terms, Tables, Lists, Examples.....</b>	<b>BE1/8-179</b>
8.1	Terms.....	BE1/8-180
8.2	Table of start softkeys .....	BE1/8-183
8.3	Color table.....	BE1/8-184
8.4	List of accessible system variables .....	BE1/8-186
8.5	List of PI services .....	BE1/8-195
8.6	Examples .....	BE1/8-199
<b>I</b>	<b>Index .....</b>	<b>BE1/I-203</b>
I.1	Subject Index .....	BE1/I-203



# 1

## 1 Introduction

1.1 Objective .....	BE1/1-8
1.2 Performance range and mode of operation .....	BE1/1-8
1.3 Boundary conditions for use .....	BE1/1-9
1.4 Syntax rules in this documentation .....	BE1/1-10

## 1.1 Objective

**Expand user interface** is implemented with an interpreter and configuration files that describe the user interfaces.

The **Expand user interface** tools can be used to create user interfaces that represent functional expansions for specific machine manufacturers or end users, or simply to implement your own screen form layout. User environments configured by Siemens or non-Siemens machine manufacturers can be modified or replaced.

The interpreter is available for HMI Embedded sl, ShopMill and ShopTurn on NCU as well as HMI Advanced.

Parts programs, for example, can be edited on user interfaces created by users. Screen forms can be created directly on the control system.

### Prerequisites

An additional graphics program is needed to produce graphics/display images. For HMI-Embedded sl you also need the application disk and the PaintShopPro tool (<http://www.jasc.com>).

The tool box supplied contains configuration examples for new screen forms. You can also use these examples as a template for creating your own forms.

## 1.2 Performance range and mode of operation

1. Open screen forms containing the following elements:
  - Softkeys
  - Variables, tables
  - Texts and Help texts
  - Graphics and Help displays
2. Initiate operations in response to the following actions:
  - Open a screen form
  - Input values (variables)
  - Select a softkey
  - Exit a screen form
3. Restructure screen forms dynamically:
  - Edit and delete softkeys
  - Define and design variable fields
  - Insert, exchange and delete display texts (language-dependent and independent)
  - Insert, exchange and delete graphics
4. Apply protection levels according to user classes

5. Variables
  - Read (NC, PLC and user variables)
  - Write (NC, PLC and user variables)
  - Combine with mathematical, comparison or logic operators
6. Execute functions:
  - Subprograms
  - File functions
  - PI services
  - External functions (HMI Advanced)
7. Data exchange between screen forms
8. The screen form can be invoked by:
  - Pressing the (start) softkeys
  - Selecting the PLC

**Expand user interface** is configured using ASCII files stored on the HMI.

These configuration files, which combine to describe the user interface as a whole, are created using the ASCII Editor.

The syntax that must be applied in creating these files is described in the following chapters.

---

#### Note

The programming support functions described in Chapter 4 and the user interfaces for Siemens cycles have been created with the system tools for **Expand user interface**. As a result, they can be modified as required by the machine manufacturer or end user within the scope described in this chapter.

---

## 1.3 Boundary conditions for use

- It is only possible to switch between screen forms within one HMI operating area.
- In the case of HMI Advanced, user, setting and machine data are initialized on request.
- User variables may not have the same names as system or PLC variables.
- The screen forms activated by the PLC form a separate operating area for HMI Advanced (similar to measuring cycle displays).
- Chapter 7 contains system-specific information concerning the number and size of configuration files.

## 1.4 Syntax rules in this documentation

<b>Keywords</b>	Keywords are recognized as part of the syntax in the configuration file. Key words are in <b>bold</b> .
<i>Parameters</i>	Parameters can be modified and are replaced by other words/symbols in the configuration file. Parameters are in <i>italics</i> .
<i>Par.xy</i> = value	Parameters with corresponding abbreviation (ID parameters) can be modified during runtime. Abbreviations are in <b>bold</b> .

---

### Note

When an ID parameter is modified during operation, it is not necessary to specify the parameter if it is unambiguous:

**xy** = value

---

[optional syntax] Optional syntax can, but need not, be configured. Optional syntax is represented in square parentheses.

---

### Note

If the square parentheses are in **bold** ([...]), then the entry is not optional syntax, but square parentheses, which form part of the syntax, i.e., square parentheses in bold must be configured.

---

*Example* Configuration examples are shown in a different font.

*; Comment* Comments are preceded by a semicolon (;):

- in the configuration examples in this documentation and
- as syntax elements in the configuration files

### *Positional*

*parameters* Positional parameters are parameters that are defined on the basis of their position.

The slash "/" separates the parameter groups. *A parameter group comprises one or more individual parameters.*

A comma (",") is used to separate the individual parameters of in a parameter group.

Undefined parameter groups are replaced by a single forward slash ("/"), which acts as a place holder.

Undefined single parameters are replaced by a single comma (","), which acts as a place holder.

"/" place holders are omitted after the last defined parameter group.

"," place holders are used after the last defined individual parameter in a parameter group.



## Example syntax

```
//M(Identifier/[Header]/[Graphic]/[Dimension]/[System
or user variable]/[Graphic position]/[Attributes])
```

---

### Note

All possible values, which can be assumed by a parameter, are listed for some positional parameters. Only **one** of the specified values can be assigned.

*Identifier*=(Text[,Status][,Access level])

with status           se0  
                          se1  
                          se2  
                          se3

HS2= ("thread" , se1)

If alternative parameters are possible, the separator | is placed between the alternatives in the syntax description.

---

## Example call

```
//M(Mask1 / "CYC83" / "c83.bmp" / , , 300 , 200 // 10)
```

;Name of identifier: Form1

;Header: CYC83

;Graphics: c83.bmp

;Dimension: Distance from left and top: Standard positions are system-dependent, width 300 pixels, height 200 pixels

;System or user variables: none

;x-position of Help display: 10 pixels

;Attribute: none

---

### Note

The positions and size of display elements are specified in pixels with reference to standard displays with 640x480 pixels. In displays with higher resolutions, these positions are converted automatically.

---

`$8xxxx` Call a text from a separate text file. Texts used in screen forms can be specified explicitly in configuration files or, particularly in cases where screen forms are to be displayed in different languages, stored separately in text files.

Language-dependent texts can be used for:

- Softkey labels
- Headings
- Help texts
- Any other texts

The screen forms refer to texts in different text files via text identification numbers (number after the \$ character). Text files are usually language-dependent. Entries in control files determine which text files must be activated.

The structure of the configuring environment in Chapter 7 describes how selected text files can be activated for the target systems HMI Embedded sl and HMI Advanced.

The syntax for creating screen form texts in separate files is also to be found in Chapter 7.

`"text"` *text* is text defined by the user and can be entered directly as such in the configuration file.

Notations `$8xxxx` and `"text"` can be used as alternatives.



# 2

## 2 First Steps

2.1 Fundamentals of Configuration.....	BE1/2-14
2.1.1 Configuration and start files .....	BE1/2-15
2.1.2 Search sequence for configuration files .....	BE1/2-15
2.1.3 Calling your own screen forms from PLC .....	BE1/2-16
2.2 Creating configuration file as ASCII file .....	BE1/2-16

## 2.1 Fundamentals of Configuration

### Editor

An ASCII editor (e.g., Notepad or the HMI editor) is used to create configuration files.

### Files

The defining data for new user interfaces are stored in configuration files. These files are automatically interpreted and the result displayed on the screen. Configuration files are not stored in the software supplied and must be set up by the user.

### Principle of the screen form tree

Several interlinked screen forms create a screen form tree. A link exists if you can switch from one screen form into another. You can use the newly defined horizontal/vertical softkeys in your form to call the preceding or any other screen form.

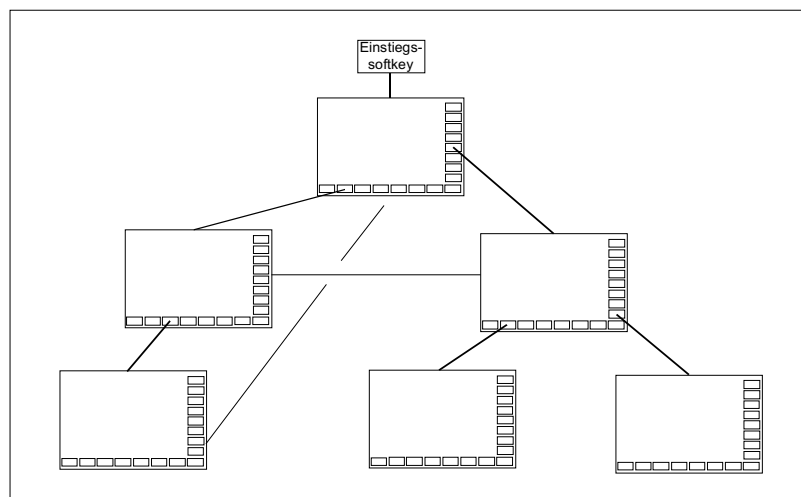


Fig. 2-1: Screen form tree

A screen form tree can be created behind each start softkey.

### Reverting to the standard application

You can exit the newly created user interfaces and return to the standard application.



You can use the <RECALL> key to close new user interfaces if you have not configured this key for any other task.

## 2.1.1 Configuration and start files

Every application uses permanent (HMI Embedded sl, ShopMill, and ShopTurn on NCU) or preset (HMI Advanced) start softkeys, which can be selected to open newly configured screen forms. Additional start softkeys can be configured with HMI Advanced.

A table containing the start softkeys with the assigned configuration files (default setting) can be found in the section "Table of start softkeys".

### Other files

In the event of a call to load a screen form (LM) or a softkey menu (LS) in a configuration file, a new file name containing the object called can be specified. This makes it possible to structure the configuration, e.g., all functions in one operation level in a separate configuration file.

## 2.1.2 Search sequence for configuration files

### HMI Embedded on NCU

For HMI Embedded sl as well as ShopMill and ShopTurn on NCU, the system searches the relevant directories of the CompactFlash card when accessed.

A setting can be made in file COMMON.COM in the standard cycle directory/user cycle directory of HMI Embedded sl to specify whether a configuration file search should be carried out each time access is attempted (only relevant when the screen forms are being set up directly on the control) or if the file already located and buffered should be reused (corresponds to the standard operating scenario).

---

#### Note

Further information about setting the configuring and operational environments can be found in Chapter 7.

---

### HMI Advanced

With HMI Advanced, the system first attempts to access the configuration files in the user cycle directory followed by the standard cycle directory and the manufacturer cycle directory.

### 2.1.3 Calling your own screen forms from PLC

Screen forms can be selected via the PLC as well as via softkeys. An interface is available for signal exchange between the PLC and HMI.

For more information about display selection from the PLC, see section "PLC interface".

## 2.2 Creating configuration file as ASCII file

### Start softkeys

One or more softkeys (start softkeys), which are used to initiate your own operating sequences, are defined in one of the specified configuration files.

The loading of a dedicated screen form is associated with a softkey definition or another softkey menu. These are then used to perform the subsequent actions.

### Screen form definition to define the screen form layout

Some examples of screen form content include:

- Short text
- Graphics text
- Input/output fields (variables)
- Unit text
- Images/displays
- Tables
- Softkey menus, etc.

Definition of the methods/functions that are used for the purpose of processing the form screen elements.

### Function

Pressing the start softkey loads the assigned screen form. This will also activate the softkeys associated with the screen form. Variables will be output to the standard positions unless specific positions have been configured.

---

#### Note

Further information about the configuration file syntax can be found in Chapter 3.

---

# 3

## 3 Syntax

3.1 Structure of a configuration file .....	BE1/3-19
3.2 Defining a start softkey .....	BE1/3-20
3.3 Syntax of screen form descriptions.....	BE1/3-21
3.3.1 Defining a screen form.....	BE1/3-21
3.3.2 Screen form properties .....	BE1/3-24
3.3.3 Screen form elements.....	BE1/3-27
3.3.4 Multi-column screen forms.....	BE1/3-34
3.3.5 Differences in screen form appearance.....	BE1/3-35
3.3.6 Integer expansions of the variable parameter group 1).....	BE1/3-37
3.3.7 Binary, hexadecimal and exponential notation .....	BE1/3-39
3.3.8 More information about variable parameters .....	BE1/3-40
3.3.9 Variable properties.....	BE1/3-43
3.3.10 Display images .....	BE1/3-45
3.3.11 Arrays.....	BE1/3-49
3.3.12 Table element (grid).....	BE1/3-54
3.4 Syntax of softkey descriptions .....	BE1/3-61
3.5 Methods .....	BE1/3-65
3.5.1 PRESS method.....	BE1/3-66
3.5.2 LOAD method .....	BE1/3-67
3.5.3 UNLOAD method .....	BE1/3-67
3.5.4 CHANGE method .....	BE1/3-67
3.5.5 FOCUS method .....	BE1/3-68
3.5.6 OUTPUT method .....	BE1/3-69
3.5.7 Version management in conjunction with OUTPUT blocks..	BE1/3-69
3.6 Functions .....	BE1/3-71
3.6.1 Screen form or softkey change.....	BE1/3-71
3.6.2 PLC/NC variable, write/read MD.....	BE1/3-75
3.6.3 General screen form variables.....	BE1/3-78
3.6.4 General functions.....	BE1/3-81
3.6.5 Subprograms .....	BE1/3-84
3.6.6 File functions.....	BE1/3-87
3.6.7 NC Functions (PI services) .....	BE1/3-93
3.6.8 External functions (only HMI Advanced) .....	BE1/3-93
3.6.9 Operators and functions .....	BE1/3-95
3.6.10 Use of strings .....	BE1/3-98

3.6.11 Functions for start softkeys.....	BE1/3-103
3.7 Error handling, log book.....	BE1/3-103



### 3.1 Structure of a configuration file

A configuration file consists of the following elements:

- Description of the start softkeys
- Definition of the screen form
- Description of the blocks
- Definition of the softkey menu(s)

Below is a simplified example of this structure:

```
; Definition of the start softkey (optional)
//S (START)
....
//END

; Definition of the screen form
//M (.....)
Definition of the variable
DEF .....

; Description of the blocks:
LOAD
...
END_LOAD

UNLOAD
...
END_UNLOAD

ACTIVATE
...
END_ACTIVATE
...
//END

; Definition of a softkey menu
//S (...)
//END
```

## 3.2 Defining a start softkey

Start softkeys are softkeys that are not called from a screen form, but which have been configured **before** the first new screen form.

Start softkeys can be used to initiate certain functions. There are various ways to define start softkeys (see "Syntax Methods").

In order to access the start screen or a start softkey menu, the start softkey must be defined.

The definition block for a start softkey is structured as follows:

Start identifier of start softkey	//S(Start)
Define start softkey	HS1=(...) ;horiz. SK 1
Method LM or LS function End of method	PRESS(HS1) LM... END_PRESS
End identifier of start softkey	//END

**//S (Start)** ;Start of identifier softkey...  
**//END** ;End identifier

The softkey is defined in the start softkey description block (see chapter "Defining a softkey") and the "Display screen form" or "Display softkey menu" function is assigned in a method (see chapters "Methods for start softkeys" and "Functions for start softkeys").

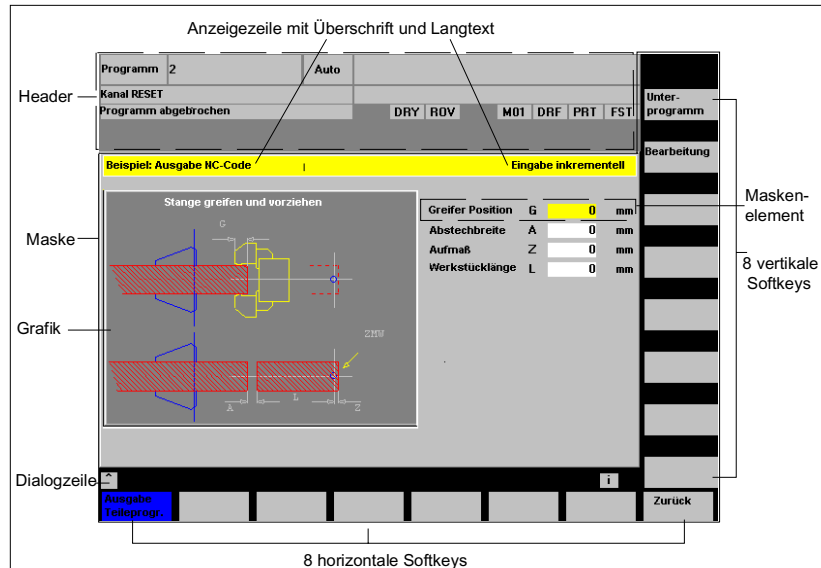
If the start softkey definition is designated as a comment (semicolon (;) at beginning of line) or the configuration file removed, the start softkey will not function.

```
//S(Start) ;Start identifier
HS6=( "1st form" ) ;Label horizontal softkey 6 "1st form"
PRESS(HS6) ;PRESS method for horiz. SK 6
    LM( "Form1" ) ;Load form1 function, where form 1 must be defined
                    ;within the same file.
END_PRESS ;End of PRESS method
HS7=( "2nd form" ) ;Label horizontal softkey 7 with "2nd form"
PRESS(HS7) ;PRESS method for horiz. SK 7
    LM( "Form2" ) ;Load form2 function, where form2 must be defined
                    ;within the same file.
END_PRESS ;End of PRESS method
//END ;End identifier of start block
```

## 3.3 Syntax of screen form descriptions

### 3.3.1 Defining a screen form

A screen form is part of a user interface consisting of a display line, form elements and/or graphic, dialog line and 8 horizontal and 8 vertical softkeys.



The definition of a screen form (form definition block) is basically structured as follows:

Start identifier of form	//M...
Variables	DEF Var1=...
Softkeys	HS1=(...)
Methods	PRESS(HS1)
Actions	LM...
End of method	END_PRESS
End identifier of form	//END

Within the screen form definition block, various variables that appear as form elements in the screen form, as well as horizontal and vertical softkeys, are defined first.

Different types of actions are then configured in methods.

The following rules must be observed when configuring screen forms:

- Texts/strings, path names and system or user variables must always be enclosed in quotation marks (" ... ").
- The *Identifier* parameter must be unique and always comprises a minimum of 2 and maximum of 32 characters. The first two characters must be two letters or an underscore and a letter (a-z, A-Z, \_).
- The system does not distinguish between upper and lower case type.
- Insertion of a space before or after the equals sign is optional.

- If a command is configured only with a file name, then the file search strategy is as follows:

Table 3-1 Search method

Entry	Search in ...
No file name, no path name	COM file of current screen form
File name	User cycles directory Manufacturer cycles directory Standard cycles directory (Flash card as well in the case of HMI Embedded sl)
File name and path name	Specified path
File name and NC path name	The system looks for the COM file on the NC, copies it to the HMI and reads it there.

### Example

A new screen form is called via start softkey "Example" from the **Parameters** operating area.

```
//S(Start)
HS7=("Example", ac7, se1)

PRESS(HS7)
  LM("Form1")
END_PRESS

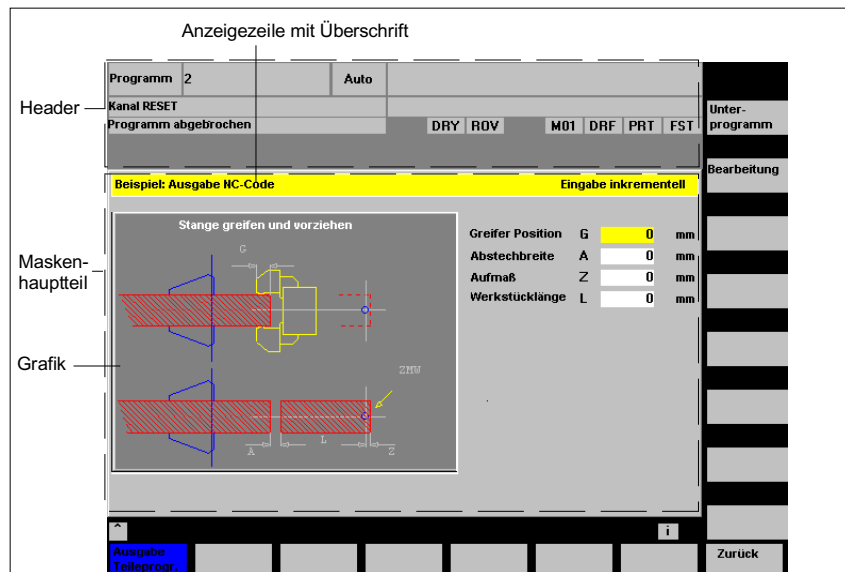
//END
//M(Form1/"Cycles")
HS1=( " ")
HS2=( " ")
HS3=( " ")
HS4=( " ")
HS5=( " ")
HS6=( " ")
HS7=( " ")
HS8=( " ")
VS1=( " ")
VS2=( " ")
VS3=( " ")
VS4=( " ")
VS5=( " ")
VS6=( " ")
VS7=( " ")
VS8=( " ")
...
;Methods
//END
```

Parameter	JOG						
Kanal RESET							T-Nr +
Programm abgebrochen			ROV			FST	T-Nr -
<b>Werkzeugkorrekturen</b>							<b>TO-Bereich 1</b>
T-Nummer	1	D-Nummer	1	Schneidenanzahl	2		D-Nr +
Werkzeugtyp	500	Schruppstahl					D-Nr -
Schneidenlage	1						Löschen...
Längenkorrektur		Geometrie		Verschleiß		Basis	Gehe zu...
Länge 1	: 1.000			0.000		0.000	Übersicht...
Länge 2	: 2.000			0.000		0.000	Neu...
Radiuskorrektur							Korrektur-
Radius	: 3.000			0.000		mm	ermitteln...
Technologie							
Freiwinkel	: 0.000			Grad			
DP25	res: 0.000						
<b>Werkzeug-</b>	R-	Setting-	Nullpunkt-	Anwender-	Aktive NV	Beispiel	
<b>korrektur</b>	Parameter	daten	verschieb.	daten	+ Korrekt.		



Parameter	JOG						
Kanal RESET							
Programm abgebrochen			ROV			FST	
<b>Zyklen</b>							

### 3.3.2 Screen form properties



The properties of the screen form are defined in the start identifier line of the form screen:

*//M(Identifier[Header]/[Graphic]/[Dimension]/[System or user variable]/[Graphic position]/[Attributes])*

with:

Identifier	Name of the screen form
Header	Screen form header as text or call for text (e.g., \$85011) from a language-specific text file (see chapter "General information about the configuring environment").
Graphic	Graphics file (with path) in double quotation marks. See also chapter "Display images".
Dimension	Position and sized of main body of screen form in pixels (distance from left, distance from top, width, height), measured from the top left corner of the screen. The measurements are separated by commas.
System or user variable	System or user variable to which current cursor position is assigned. The cursor position can be sent to the NC/PLC via the system or user variable. The first variable has the index 1. The sequence matches the configuring sequence of the variables.
Graphic position	Position of graphic (distance from left, distance from top) in pixels, measured from the top left corner of the main body of the screen form. The minimum clearance from the top is 18 pixels. The data is separated by commas.

Attributes	<p>Attributes that are entered are separated by commas. Possible attributes are:</p> <p>CMx: Column alignment (Column Mode)</p> <p>CM0: The column distribution is carried out separately for each line.</p> <p>CM1: The column distribution for the line with the most columns is valid for all lines.</p> <p>Default setting: CM0</p> <p>Response when screen form is opened (in terms of change block):</p> <p>cb0: All change blocks of the screen form are processed when screen form is opened. (default value).</p> <p>cb1: Change blocks are only processed if there is a change in the associated value.</p> <p>For the variable, Cb attributes that are specified in a variables definition take precedence over the global specification in the screen form definition.</p>						
System	<p>The <b>System</b> screen form property can be read during runtime:</p> <table border="0"> <tr> <td>Values:</td> <td>0</td> <td>HMI_Embedded is being used</td> </tr> <tr> <td></td> <td>1</td> <td>HMI_Advanced is being used</td> </tr> </table>	Values:	0	HMI_Embedded is being used		1	HMI_Advanced is being used
Values:	0	HMI_Embedded is being used					
	1	HMI_Advanced is being used					

### Accessing the screen form properties:

Read and write access to the "Header", "Graphics" and "System variable" properties of the screen form is granted within methods (e.g., PRESS block):

**Hd** = *Header*  
**Hlp** = *Graphics*  
**Var** = *System or user variable*

### Example

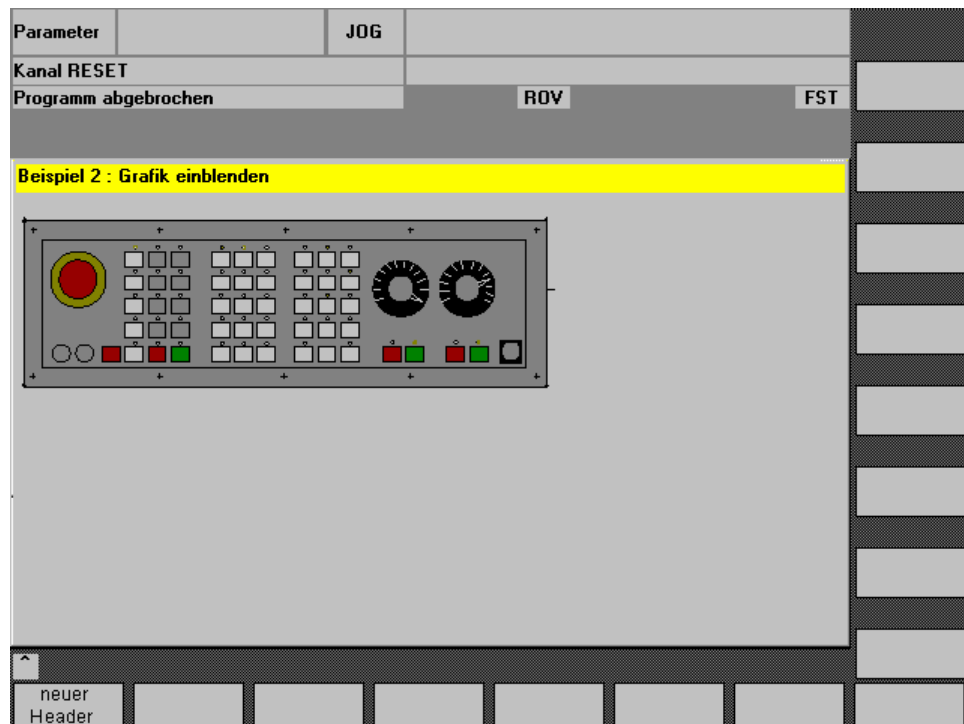
```
//S(Start)
HS7=("Example", se1, ac7)

PRESS(HS7)
  LM("Form1")
END_PRESS

//END
//M(Form1/"Example 2 : Display graphics"/"MST.BMP")
HS1=("new%nHeader")
HS2=("")
HS3=("")
HS4=("")
HS5=("")
HS6=("")
HS7=("")
HS8=("")
```

```
VS1=("")  
VS2=("")  
VS3=("")  
VS4=("")  
VS5=("")  
VS6=("")  
VS7=("")  
VS8=("")
```

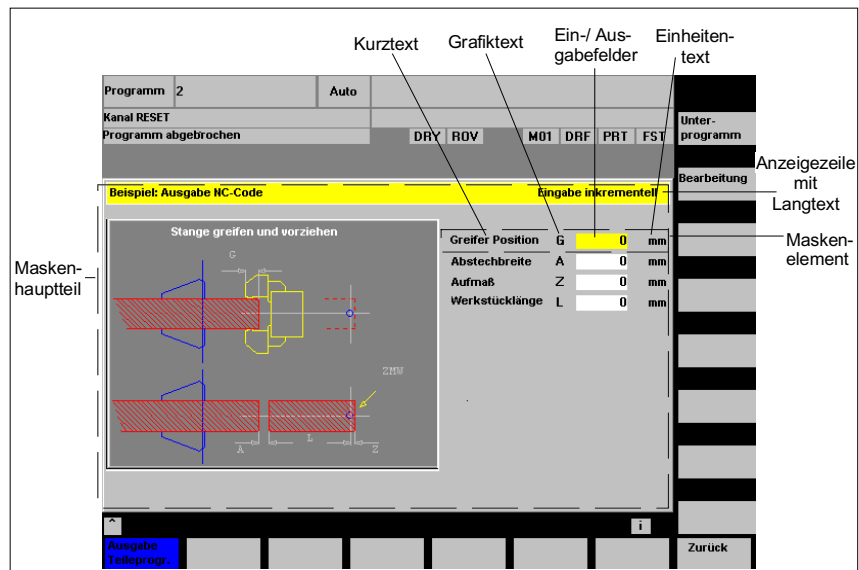
```
PRESS(HS1)  
  Hd= "New header"  
END_PRESS  
...  
//END
```



Display for example 2: Display graphics



### 3.3.3 Screen form elements



Screen form elements fill lines in the main body of the form. One or several screen form elements can be defined in each line (see chapter "Multi-column screen forms").

The term "screen form element" refers to the visible part of a variable, i.e., short text, graphics text, input/output field (I/O field) and unit text.

All variables are valid only in the active screen form. Properties are assigned to a variable when it is defined.

The values of form screen properties can be accessed within methods (e.g., a PRESS block). (See also chapter "Variable properties".)

#### Defining variables

```

DEF Identifier =
  1
  (Variable type)
  2
  /[limit values or toggle field or table identifier]
  3
  /[Default]
  4
  /[Texts (Long text, Short text|Image, Graphics text, Units text)]
  5
  /[Attributes]
  6
  /[Help display]
  7
  /[System or user variable]
  8
  /[Position of short text]
  9
  /[Position I/O field(Left, Top, Width, Height)]

```

<sup>10</sup>  
/[Colors]  
<sup>11</sup>  
/[Help]\*)

\* HMI Advanced only

---

### Note

The small figures above "(" and ")" are only an aid to assigning the parameter groups. Here the numbers are only provided in front of the parameter explanations for the purpose of facilitating assignment. They are not part of the syntax. The single parameters to be separated by commas are enclosed in round parentheses.

---

with:

Identifier                      Name of variable

- 1) Variable type      Variable type:
- |      |   |
|------|---|
| R[x] | Real (+ digit for the decimal place)                        |
| I    | Integer<br>(refer to chapter "Integer expansions").         |
| S[x] | String (+ digit for string length)                          |
| C    | Single character  |
| B    | Bool  |
| V    | Variant (see chapter "Details about Variant variable type") |

The variable type **must** be specified.

- 2) Limit values      Limit values for variables:  
Limit value MIN, limit value MAX  
The limit values are separated by a comma.  
Limit values can be entered for types I, C and R in  
decimal format or as characters in the format  
"A", "F".

The Limit Values parameter works in conjunction with the  
Limit attribute (see Attributes, Limit).  
Default setting: Empty

or

- Toggle field              List with predefined entries in the I/O field.  
The list is initiated by a \* ; the entries are  
separated by commas.  
The entries can be assigned a value  
(see "Examples").  
The entry for the limit value is interpreted as a list  
for the toggle field.  
Entering only a \* generates a variable  
Toggle field (see "Variable toggle fields").  
Default setting: No default

or

## Table identifiers

Identifiers of a table containing NCK/PLC values of the same type, which can be addressed via a channel block. The table identifier is differentiated from limit values or toggle fields by the addition of a % sign in front of it. The file containing the table description can be specified by adding a comma after the identifier and then inserting the name of the file.

## 3) Default setting

Default setting of variables

If no default has been configured, and if the variable has not been assigned any system or user variables (see system or user variable property), the first element in the toggle field is assigned.

If no toggle field is defined no default setting will be made, i.e., the variable is allocated the "not calculated" state.

Default setting: No default

See also table "Effect of defaults" in chapter "More information about variable parameters".

## 4) Texts

Texts: Long text, short text, graphics text, unit text with:

Long text: Text in the display line

Short text: Name of the screen form element

Graphics text: Text refers to the terms in the graphics.

Unit text: Unit of screen form element

(See graphic image above)

The sequence is specified.

Default setting: Empty

Instead of a short text, an image can also be displayed.

## 5) Attributes

The attributes influence the following properties:

- **Input mode**

- **Access level**

- **Alignment of short text**

- **Font size**

- **Handling the limit values**

- **Response when screen form is opened** in terms of **change block**

The attributes are separated by commas, any sequence is possible. The attributes are not valid for toggle fields.

A definition can be made for each component.

**Input mode**

(display concerns the elements of the variables):

wr0 I/O field invisible, short text visible

wr1 Read (no focus possible for input)

wr2 Read and write (line appears in white)

wr3 wr1 with focus

wr4 All elements of the variables are invisible, no focus possible

wr5 The value entered is saved immediately whenever the key is pressed (unlike wr2 – here a save is only performed on exiting the field, or when RETURN is pressed).  
Default setting: wr2

#### Access Level

Empty Can always be written  
ac0...ac7 Protection levels (see chapter "Syntax of softkey description")  
If the access level is not adequate, then the first line is displayed grayed-out, default setting: ac7

#### Text alignment, short text

al0 Left-justified  
al1 Right-justified  
al2 centered  
Default setting: al0

#### Font size

fs1 Default font size (8 pt.)  
fs2 Double font size  
Default setting: fs1  
The clearances between the lines is defined.  
For the default font size, 16 lines can fit into the screen form.  
Graphics and unit text can only be configured in the default font size.

#### Handling the limit values

Limit can be used to check whether the value of the variables is within the specified min. and max. limit values.

li0 No check  
li1 Check with respect to min.  
li2 Check with respect to max.  
li3 Check with respect to min. and max.  
Default setting: Determined by specified limit values

#### Response when screen form is opened in terms of change block

cb0: The change block, which is defined for this variable, is processed when the form screen opens (default value).  
cb1: The change block, which is defined for this variable, is only processed if there is a change in the value of the variables.

For the variables, cb attributes that are specified in a variables definition take precedence over the global specification of cb attributes in the screen form definition.

Multiple attributes are separated by a comma.

6) Help display Help display file: File name  
The name of the Help display file is in double quotation marks.  
The display appears automatically (replacing the



11) Help

HMI Advanced only:

Help: Help file, Index, Help text  
with:

Help file: Path specification of the file (PDF format)  
Index: Index in the Help text file to the Help text  
Help text: Help text for display in the Help text file

Data is separated by a comma, the sequence is fixed.  
The Help file and Help text appear in double quotation marks.  
Several Help references can be configured in a loop (Help loop) for each screen form element, i.e., the references are called up in succession and when the last reference closes, the first is displayed again.

If reference is made to the same file/index/Help text in a second or subsequent Help call, the entries do not have to be made (see "Examples").

The Help appears when the cursor is positioned on this field and the Help key is pressed.

The PDF files must be located in the  
CUS.DIR\hlp.dir or CST.DIR\hlp.dir directory.

Identical PDF and text files must have the same name.  
The name of the PDF file must be written in uppercase letters in the txt file.

**Examples**

**Example 1**

Assign properties Variable Type, Texts, Help, Colors:

```
DEF Var1 = (R///,"Actual value",,"mm"//"Var1.bmp"////8,2)
```

Variable Type:	Real
Limit values or toggle field entries:	None
Default:	None
Texts:	
Long text:	None
Short text:	Actual value
Graphics text:	None
Unit text:	mm
Attributes:	None
Help display:	Var1.bmp
System or user variable:	None
Position of short text:	No data, i.e., default position
Position I/O field:	No data, i.e., default position
Colors:	
Foreground color:	8
Background color:	2
Help:	none

**Example 2**

Assign properties Variable Type, Limit Values, Attributes, Short Text Position:

DEF Var2 = (I/0,10///wr1,a11///,,300)

Variable Type:	Integer
Limit values:	MIN: 0; MAX: 10
Default:	None
Texts:	None
Attributes:	
Input mode:	read-only
Alignment of short text:	Right-justified
Help display:	None
System or user variable:	None
Position of short text:	
Distance from left:	None
Distance from the top:	None, i.e., default distance from the top left
Width:	300
Position I/O field:	No data, i.e., default position
Colors:	No data, i.e., default setting
Help:	None

**Example 3**

Assign properties Variable Type, Default, System or User Variable, I/O Field Position:

DEF Var3 = (R//10//// "\$R[1]"//300,10,200//  
"Help.pdf",1,"Help1")

Variable Type:	String
Limit values or toggle field entries:	None
Default:	10
Texts:	None
Attributes:	None
Help display:	None
System or user variable:	\$R[1] (R-Parameter 1)
Position, short text:	Default position relative to the I/O field
Position of the I/O field:	
Distance from left:	300
Distance from the top:	10
Width:	200
Colors:	No data, i.e., default setting
Help:	In the Help.pdf file, the Help with the Help text "Help1" is called on the page with index 1 when the <I> key is pressed.

**Example 4**

Various entries in the toggle field:

DEF Var1 = (I/\* 0,1,2,3)

DEF Var2 = (S/\* "In", "Out")

DEF Var3 = (B/\* 1="In", 0="Out")

;1 and 0 are values, "In" and "Out"  
;are displayed

DEF Var4 = (R/\* ARR1)

;ARR1 is the name of an array

### Example 5

Several Help calls per screen form element (HMI Advanced only):

```
DEF Var1 = (R/////////"Help1.pdf",1,"Help1",
,2,"Help2", "Help3.pdf", 3,)
```

;1. Entry in the Help loop

Help file: HELP1.PDF

Index: 1

Help text: Help1

2. Entry in the Help loop

Help file: HELP1.PDF

Index: 2

Help text: Help2

3. Entry in the Help loop

Help file: HELP3.PDF

Index: 3

Help text: Help3

### Example 6 Displaying an image instead of a short text

```
DEF VAR1= (V///,"\\image1.bmp" ///160,40,50,50)
```

The size and position of the image is defined under *Position I/O field (left, top, width, height)*.

## 3.3.4 Multi-column screen forms

Multiple variables can also be represented in the main part of the screen form on one line.

In this case, the variables are all defined in the configuration file on a single definition line.

```
DEF VAR11 = (S///"Var11"), VAR12 = (I///"Var12")
```

To make individual variables in the configuration file more legible, the definition lines can be wrapped after every variables definition and following comma.

The key word "DEF" always indicates the beginning of a new line.

```
DEF Tnr1=(I//1/"", "T ", ""/wr1///, ,10/20, ,50),
TOP1=(I///, "Type=" /WR2/"$TC_DP1[1,1]" /80, ,30/120, ,50),
TOP2=(R3///, "L1=" /WR2/"$TC_DP3[1,1]" /170, ,30/210, ,70),
TOP3=(R3///, "L2=" /WR2/"$TC_DP4[1,1]" /280, ,30/320, ,70),
TOP4=(R3///, "L3=" /WR2/"$TC_DP5[1,1]" /390, ,30/420, ,70)
DEF Tnr2=(I//2/"", "T ", ""/wr1///, ,10/20, ,50),
TOP21=(I///, "Type=" /WR2/"$TC_DP1[2,1]" /80, ,30/120, ,50),
TOP22=(R3///, "L1=" /WR2/"$TC_DP3[2,1]" /170, ,30/210, ,70),
TOP23=(R3///, "L2=" /WR2/"$TC_DP4[2,1]" /280, ,30/320, ,70),
TOP24=(R3///, "L3=" /WR2/"$TC_DP5[2,1]" /390, ,30/420, ,70)
...
```



Parameter 1		JOG	MPPFO			
Kanal RESET						
Programm abgebrochen		ROV			FST	
TOA-Liste		Eingabe Werkzeugtyp				
T	1	Typ= 500	L1= 1.000	L2= 2.000	L3= 3.000	
T	2	Typ= 220	L1= 4.000	L2= 5.000	L3= 6.000	
T	3	Typ= 200	L1= 7.000	L2= 8.000	L3= 9.000	
T	4	Typ= 220	L1= 10.000	L2= 11.000	L3= 12.000	
T	5	Typ= 200	L1= 13.000	L2= 14.000	L3= 15.000	
T	6	Typ= 200	L1= 16.000	L2= 17.000	L3= 18.000	
T	7	Typ= 200	L1= 19.000	L2= 20.000	L3= 21.000	
T	8	Typ= 200	L1= 22.000	L2= 23.000	L3= 24.000	
T	9	Typ= 200	L1= 25.000	L2= 26.000	L3= 27.000	
T	10	Typ= 200	L1= 28.000	L2= 29.000	L3= 30.000	
T	11	Typ= 200	L1= 31.000	L2= 32.000	L3= 33.000	
T	12	Typ= 200	L1= 0.000	L2= 0.000	L3= 0.000	
T	13	Typ= 200	L1= 37.000	L2= 38.000	L3= 39.000	
T	14	Typ= 200	L1= 40.000	L2= 41.000	L3= 42.000	
T	15	Typ= 200	L1= 43.000	L2= 44.000	L3= 45.000	
T	16	Typ= 200	L1= 47.000	L2= 48.000	L3= 49.000	

When designing multi-column screen forms, consideration should be given to the capability of the hardware used, e.g.,  
HMI Embedded sl Max. 10 columns, Max. 60 DEF statements

### 3.3.5 Differences in screen form appearance

If **HMI Advanced** is installed together with **ShopMill** or **ShopTurn** on one operator panel front, then the two systems have different font types.  
HMI Advanced has **proportional fonts**, while **JobShop** products and **HMI Embedded sl** have a **fixed font**.

If **Expand user interface** is used on HMI Embedded sl and HMI Advanced, screen forms that are defined identically will appear differently by default.

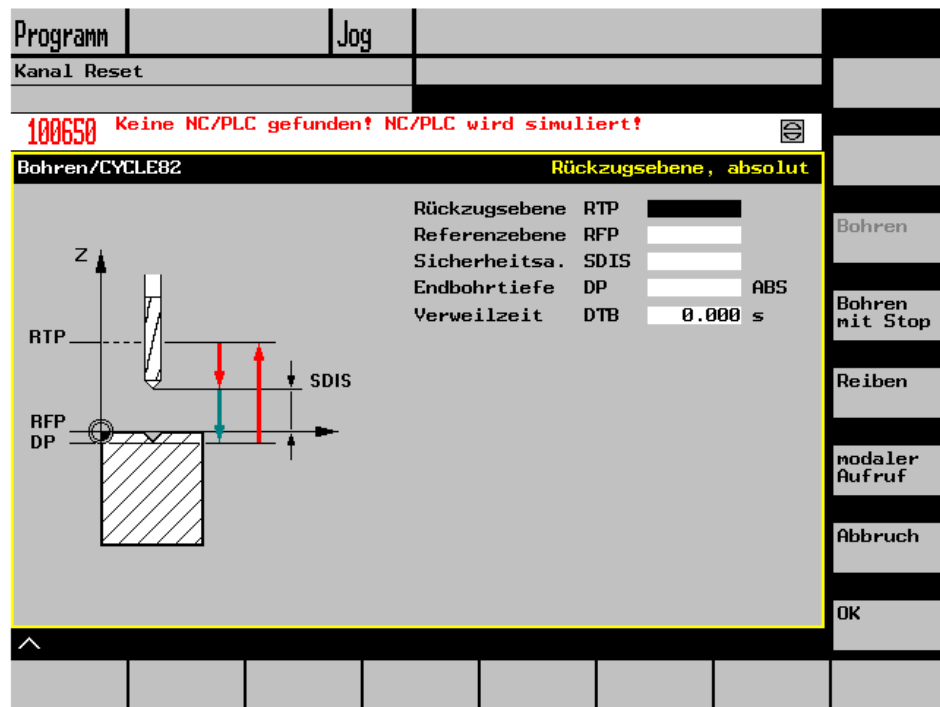


Fig. 3-1 HMI Embedded sl screen form appearance

Below you can see how, despite being configured identically, the same screen appears differently under HMI Advanced.

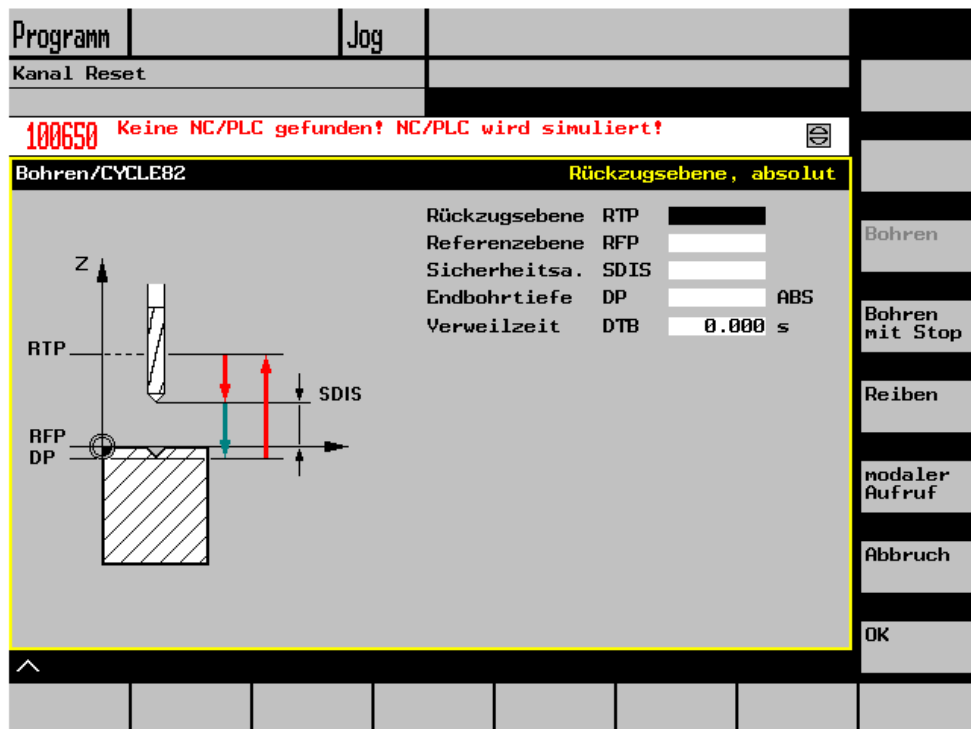


Fig. 3-2 HMI Advanced screen form appearance

## Setting for the same screen appearance

To achieve a uniform screen appearance in respect of the systems mentioned above, a fixed font should be used.

To enable a consistent screen form appearance with HMI Advanced, the fixed font is adopted via the setting in the WIZARD.INI file.

### [FONT]

#### FixedFont=1

The text will be displayed in the current language in fixed font format. The only exception to this is text for the softkey labels. Unlike the HMI Embedded sl input field, the HMI Advanced input field contains an input mode pictogram. This means it is smaller than the HMI Embedded sl field.

As you can see above, the combination of short text, graphics text, input field, and unit text is approximately the same length and contains the same information in both screens.

However, individual fields can be positioned differently in the two HMI versions depending on the length of the text to be displayed and the text field within which it will be contained.

To enable a common configuration file to be produced for both systems, the SYSTEM screen form property has been created. The runtime environment can be scanned in the SYSTEM screen form property. This enables one configuration to be used for both target systems.

### 3.3.6 Integer expansions of the variable parameter group 1)

The following extensions for determining the **display** in the I/O field and the **memory utilization** are possible for the Integer type:

Table 3-2 2nd character in the extension data type

Display format	
B	binary
D	Decimal signed
H	Hexadecimal
No data	Decimal signed

Table 3-3 3rd and/or 4th character in extension data type

Memory utilization	
B	Byte
W	Word
D	Double Word
BU	Byte, Unsigned
WU	Word, Unsigned
DU	Double word, Unsigned

## Sequence of characters in type specification

1. I Basic integer designation
2. Display format
3. Memory utilization
4. U Unsigned

**Examples** of valid Integer type definitions:

IB	Integer variable 32 bits in binary notation
IBD	Integer variable 32 bits in binary notation
IBW	Integer variable 16 bits in binary notation
IBB	Integer variable 8 bits in binary notation
I	Integer variable 32 bits in decimal notation signed
IDD	Integer variable 32 bits in decimal notation signed
IDW	Integer variable 16 bits in decimal notation signed
IDB	Integer variable 8 bits in decimal notation signed
IDDU	Integer variable 32 bits in decimal notation unsigned
IDWU	Integer variable 16 bits in decimal notation unsigned
IDBU	Integer variable 8 bits in decimal notation unsigned
IH	Integer variable 32 bits in hexadecimal notation
IHDU	Integer variable 32 bits in hexadecimal notation
IHWU	Integer variable 16 bits in hexadecimal notation
IHBU	Integer variable 8 bits in hexadecimal notation

## Information about Variant variable type

The Variant variable type is determined by the data type of the last value assignment. Currently, it can be scanned using the ISNUM or ISSTR functions. The display mode can be changed for the type property of a variable.

**ISNUM** (VAR), where:

VAR Name of the variable whose data type must be checked  
The result of the scan can be:  
FALSE = not a numerical variable (data type = String)  
TRUE = numerical variable (data type = Real) or

**ISSTR** (VAR), where:

VAR Name of the variable whose data type must be checked  
The result of the scan can be:  
FALSE = numerical variable (data type = Real)  
TRUE = not a numerical variable (data type = String)  
The Variant field type is primarily suitable either for writing variable names or numerical values to the NC code.

```
IF ISNUM(VAR1) == TRUE  
IF ISSTR(REG[4]+2) == TRUE
```

## Changing the type of a variable

The display mode can be changed for the type property of a variable.  
For **integer** variables this is display type:

B = binary
D = decimal signed
H = hexadecimal unsigned
+ U for Unsigned

For **Real**, only the number of places after the decimal point can be changed.  
Changing the basic type is illegal and generates an error message in file error.com.

Parameterization example:  
Var1.typ = "IBW"  
Var2.typ = "R3"

### 3.3.7 Binary, hexadecimal and exponential notation

Numbers can be represented in either binary, decimal, hexadecimal or exponential notation.

---

#### Note

When codes are generated with the GC function, numerical values in decimal or exponential notation are evaluated, but not those in binary or hexadecimal notation.

---

Binary:	'B01110110'
Decimal:	123.45
Hexadecimal:	'HF1A9'
Exponential:	'-1.23EX-3'

---

#### Note

Numerical values in binary, hexadecimal and exponential notation must be enclosed in apostrophes.

---

```
VAR1 = 'HF1A9'
REG[0]= 'B01110110'
DEF VAR7 = (R/'-1.23EX-3')
```

### 3.3.8 More information about variable parameters

#### Variable parameter group 3)

##### Default setting

A variable can assume various states depending on whether a default value, or a system or user variable, or both, is assigned to the variable field (I/O field or toggle field).

Table 3-4 Effect of defaults

If...			Then...
Field type	Default value	System or user variable	Reaction of field type
I/O field	Yes	Yes	Write default value to system or user variable
	No	Yes	Use system or user variable as default value
	Error	Yes	Not calculated, system or user variable is not written into/used.
	Yes	No	Default value
	No	No	Not calculated
	Error	No	Not calculated
	Yes	Error	Not calculated
	No	Error	Not calculated
	Error	Error	Not calculated
Toggle	Yes	Yes	Write default value to system or user variable
	No	Yes	Use system or user variable as default value
	Error	Yes	Not calculated, system or user variable is not written to/used.
	Yes	No	Default value
	No	No	Default = first toggle field element
	Error	No	Not calculated
	Yes	Error	Not calculated
	No	Error	Not calculated
	Error	Error	Not calculated

Not calculated: Toggling is not possible until a valid value is assigned to the variable

## Variable parameter group 8), 9)

### Positions

The short text and graphics text, as well as the I/O field and unit text, are each treated like a unit, i.e., position settings for short text apply to the graphic text and settings for the I/O field and to unit text.

The configured position entry overwrites the default value, i.e., only one value can be changed.

If no position data is configured for following screen form elements, then the position settings for the preceding screen element are applied.

If no positions have been specified for any element, then default positions are applied. The column width for the short text and I/O field is calculated as standard for each line from the number of columns and maximum line width, i.e.,  
 $\text{column width} = \text{maximum line width} / \text{number of columns}$ .

The width of the graphics and unit text is predefined and optimized to suit the requirements of programming support. If graphics or unit text has been configured, the width of the short text or I/O field is reduced accordingly.

The order of short text and I/O field can be reversed by position settings.

## Variable parameter group 6)

### Help

The Help loop for a screen form element can be extended or deleted during runtime. These functions are available **only** with **HMI Advanced**.

Extending the Help loop:

**ADDHTX** (*Identifier, Help file, Index, Help text*)

with:

Identifier	Name of the variable whose Help loop will be extended
Help file	Path specification of the file (PDF format)
Index	Index in the Help file to the Help text
Help text	Help text for display in the Help text file

The Help loop can be extended as required by calling the function repeatedly.  
`ADDHTX(VAR1, "C:\OEM\HLP\MYHLP.PDF", 15, "Machine data")`

Deleting the existing Help loop:

**CLRHTX** (*Identifier*)

with:

Identifier	Name of the variable whose Help loop is to be deleted
------------	---

`CLRHTX(VAR1)`

## Variable parameter group 2)

### Toggle field extension

The toggle field extension function can be used to display texts (entries in toggle field) as a function of NC/PLC variables. A variable, which makes use of a toggle field extension, is read-only.

```
DEF Identifier =  
  (Variable type  
   /+ $Text number |  
    * wert= "\bild"[,wert="\bild2.bmp"][, ...]  
   /[default]  
   /[Texts(Long text, Short text, Graphics text, Units text)]  
   /[Attributes]  
   /[Help display]  
   /[System or user variable]  
   /[Position of short text]  
   /[Position I/O field(Left, Top, Width, Height)]  
   /[Colors]  
   /[Help])
```

with:

Variable type	Type of variables specified in the system or user variable
Text number	Number (basis) of the language-specific texts valid as the basis number
System or user variable	System or user variable (offset) used to form the final text number (basis + offset)

For an explanation of other parameters, see 3.3.3 "Defining variables".

```
DEF VAR1=(IB/+ $85000/15//// "DB90.DBB5 ")
```

When the screen form is opened, the content of text number \$85015 are displayed in the I/O field. Default value 15 is entered in system variable DB90.DBB5. If the value saved in system variable DB90.DBB5 changes, then the displayed text number \$(85000 + <DB90.DBB5>) is recalculated in response to every change.

### Variable toggle fields

It is possible to assign a variable toggle field to a screen form element. This means that when the toggle key is pressed, the variable is assigned a value configured in one CHANGE method.

A single asterisk \* is entered for the Limit Values or Toggle Field property to identify a variable toggle field when a variable is defined.

```
DEF VAR1=(S/*)
```



## Toggle-field-dependent displays

The toggle field is overlaid with graphics, which change depending on the value of the flag byte. If, for example, the flag byte has the value 1, image1 is displayed, for value 2, image2. The position and size of the image are indicated under *Position I/O field (left, top, width, height)*.

```
DEF VAR1=(IDB/*1="\image1.bmp",
2="\image2.bmp"//,$85000/wr1//""MB[0]//160,40,50,50)
```

## 3.3.9 Variable properties

### Variable value

The default property of a variable is its value. When a variable is defined, a value is allocated to it in the form of a default setting, a system or user variable or programmed in the methods, as described below.

*Identifier.val* = Variable value

or

*Identifier* = Variable value

with:

Identifier

Name of variable

Variable value

Value of variable

```
VAR3 = VAR4 + SIN(VAR5)
VAR3.VAL = VAR4 + SIN(VAR5)
```

### Variable status

The "Variable status" property can be scanned during runtime to find out whether a variable has a valid value. This property can be read and written with the value FALSE = 0.

*Identifier.vld*

with:

Identifier

Name of variable

The result of the scan can be:

FALSE = invalid value

TRUE = valid value

```
IF VAR1.VLD == FALSE
VAR1 = 84
ENDIF
```

### Variable: Changing properties

Using the notation:

*Identifier.property* = value

to change a property assigns a new value to the variable.

Examples:

Identifier.min = MIN limit value

Identifier.**max** = MAX limit value  
Identifier.**lt** = Long text  
Identifier.**st** = Short text  
Identifier.**gt** = Graphics text  
Identifier.**ut** = Unit text  
Identifier.**wr** = Input mode  
Identifier.**ac** = Access level  
Identifier.**al** = Alignment of screen form text  
Identifier.**fs** = Font size  
Identifier.**li** = Limit  
Identifier.**hlp** = Help display  
Identifier.**var** = System or user variable  
Identifier.**fc** = Foreground color  
Identifier.**bc** = Background color  
Identifier.**htx** = Help file, index, help text  
Identifier.**val** = Variable value  
Identifier.**vld** = Variable status  
Identifier.**typ** = Variable type

The possible properties are described in chapter "Screen form properties".  
The expression on the right of the equals sign is evaluated and assigned to the variable or variable property.

### Changing the softkey label

```
HS3.st = "New Text" ;Change softkey label
```

### Addressing system variables indirectly

A system variable can also be addressed indirectly, i.e., as a function of another variable.

```
PRESS(HS1)  
  AXIS=AXIS+1  
  WEG.VAR="$AA_DTBW[ "<<AXIS<<" ]"  
                                     ;address axis address via variable  
END_PRESS
```

### Help variables

Help variables are internal arithmetic variables.  
Arithmetic variables are defined like other variables, but have no other properties apart from variable value and status, i.e., Help variables are not visible in the screen form. Help variables are of the Variant type.

**DEF Identifier**

with:

Identifier	Name of Help variables
DEF OTTO	;Definition of a Help variable

A value is assigned to a Help variable in a method.

*Identifier.val* = Help value variable  
or  
*Identifier* = Help variable value

with:	
Identifier	Name of the Help variables
Help variable value	Content of the Help variables
LOAD	
OTTO = "Test "	;Assign Otto the value "Test"
	;Help variable
END_LOAD	
LOAD	
OTTO = REG[9].VAL	;Assign value of register to
	;Help variable
END_LOAD	

The **Status** property can be used to scan the Help variable to find out whether its value is valid. Status is a read-only property.

#### *Identifier.vld*

with:	
Identifier	Name of Help variables
The result of the scan can be:	
FALSE = invalid value	
TRUE = valid value	
IF OTTO.VLD == FALSE	
OTTO = 84	
ENDIF	

### Calculation with variables

Variables are calculated every time you exit an I/O field (using ENTER or TOGGLE key). The calculation is configured in a CHANGE method that is processed every time the value changes.

The variable status (see chapter "Variable properties") can be used to scan a variable to find out whether or not its value is valid.

```
Var1 = Var5 + SIN(Var2)
Otto = PI * Var4
```

### 3.3.10 Display images

There are two display categories:

- Display images/graphics in the graphic area in accordance with the display in chapter "Defining a screen form"
- Help displays illustrating, for example, individual variables, which are **superimposed** in the graphic area.
- More Help displays can be configured instead of blocks of text or input/output fields, which you position where you like.

#### Images

Maximum size:	560 * 326 pixels	(HMI Advanced/HMI Embedded sl)
	688 * 376 pixels	(PCU50 with OP012)

## Creating screen images for HMI Embedded sl

To configure screen images, proceed as follows:

- Create the screen images using, for example, the Windows 95 Paint application.
- Use the BMP2BIN conversion program (application disk). It supports 256 colors and faster display building.  
This program converts BMP bitmaps to an optimized format of the HMI Embedded sl.

The program has the following parameters:

BMP2BIN BMPfile colTab

Explanation of parameters:

BMPfile	Name of the BMP file (wildcards can be used)
colTab	Name of the color file with color conversion table. This file can be used to assign a color of a BMP pixel to a color in HMI Embedded sl. However, this is only possible for 16-color bitmaps. For 256 color bitmaps, a color file cannot be transferred.

Structure of the color file (example):

```
[BMP]
;BMP color    HMI color
1             3
2             2
...
```

The separator is at least one blank.

HMI Embedded sl can handle converted graphics as well as .bmp files. For .bmp files, the colors are converted online. For this the files containing the color table (syscol.col, sysbw.col, etc.) have been extended to include the [BMP] section. This section allocates a color of a BMP pixel to a color of the HMI.

## Unconverted display images

Although it is possible to introduce unconverted images, we strongly recommend that you convert the BMP images offline using the **BMP2BIN** application (see above, application disk), as this accelerates output considerably.

## Prerequisites

The display images must have been saved in 16-color mode.  
256-color bitmaps are also supported. You will find the corresponding color table in chapter "Color table".

## Storage locations

The following storage locations are possible:

HMI SolutionLine searches for the display images on the CompactFlash card in the following sequence:

```
/user/sinumerik/cycles/ico/xxx
/user/sinumerik/hmi/ico/xxx
/oem/sinumerik/cycles/ico/xxx
/oem/sinumerik/hmi/ico/xxx
/addon/sinumerik/cycles/ico/xxx
/addon/sinumerik/hmi/ico/xxx
/siemens/sinumerik/cycles/ico/xxx
/siemens/sinumerik/hmi/ico/xxx
```

In this case, xxx stands for the resolution-specific directory

ico1024	Display images for resolution 1024*768
ico800	Display images for resolution 800*600
ico640	Display images for resolution 640*480

HMI Embedded sl automatically finds the resolution of the connected monitor and searches for the desired display image first in the associated resolution directory. If it is not found there, HMI Embedded sl searches for the display image in the next smaller resolution directory until – if it doesn't find the display image earlier – it reaches directory ico640.

With the JobShop products that run together with HMI Advanced, the display images and COM files are read out from the DH path, whereby the DH directories are browsed in the following sequence:

Com-Files:

```
cus.dir
cma.dir
std.dir
```

Display images:

```
Cus\help.dir
Cma\help.dir
Cst\help.dir
Help.dir
```

## Procedure

Procedure when integrating the bitmaps (customer displays):

Integration on HMI Embedded sl

- Generate the BMP file using Microsoft Paint, Version 4.0 or higher
- Archive the BMP file using the arj.exe supplied (Version 2.41) into the archive CUS.ARJ or alternatively each BMP file in a separate archive with the file name of the BMP file and extension .BM\_

### Examples:

Alternative 1: Several files per archive:

```
arj a cus.arj my_file1.bmp my_file2.bmp my_file3.bmp my_file4.bmp
```

Alternative 2: One file per archive:

```
arj a my_file1.bm_ my_file1.bmp
```

## Limit

Archiving is optional. However, it must be noted that the CompactFlash card has space for at least 10 MB.

A bitmap can be assigned to any softkey if the softkey text is preceded by 2 backslashes. The text behind the backslashes is then interpreted as the name of the file that contains the bitmap.

E.g., in ALUC.TXT 85000 0 0 "\\mybitmap.bmp"

---

### Note

The display colors might appear differently on HMI Embedded sl than in Paint. Refer to chapter 7 for more information about search strategies in general and the search sequence for display images.

---

## Display building, HMI Advanced

Graphics/display images and Help displays can have the following format:

- Bitmap (BMP)
- Metafile (WMF) or
- Icon (ICO)

Graphic programs can be freely selected provided that they are capable of generating the prescribed formats.

Displays must be saved in one of the following **directories**:

\\DH\CUS.DIR\HLP.DIR (User cycles directory)  
\\DH\COM.DIR\HLP.DIR (Comments directory)  
\\DH\CST.DIR\HLP.DIR (Standard cycles directory)  
\\DH\CMA.DIR\HLP.DIR (Manufacturer directory)

## Display in display

You can insert further displays in the background (Help display). Even as a function of the values of the variables.

Like for softkeys you can assign a graphics file to a display field. The height of the displayed graphic (of the I/O field) is specified as the fourth parameter of the positions.

You can define bitmaps in fields (short text, I/O field).

(See "Defining variables" in chapter "Screen form elements").

Example:

Short text field is overlaid with graphics

```
DEF VAR1=(S///, "\\image1.bmp" ////160,40,50,50) ; Type S irrelevant
```

Toggle field overlaid with graphics, the graphic being selected with a PLC flag byte.

```
DEF VAR1=(IDB/* 1="\\image1.bmp",  
2="\\image2.bmp"//,$85000/wr1//"MB[0]"//160,40,50,50)
```

### 3.3.11 Arrays

Arrays are tables in which data can be stored.  
Arrays can be one- or two-dimensional. A one-dimensional array is treated like a two-dimensional array with just one line or column.

#### Definition

Arrays have start identifier `//A` and end identifier `//END`. The number of lines and columns is optional. An array is structured in the following way:

`//A(identifier)`

`(a/b...)`

`(c/d...)`

...

`//END`

with:

Identifier	Name of array
a, b, c, d	Values of array

---

#### Note

Values of the String type must be enclosed in quotation marks.

---

```
//A(Thread)
; Size / Pitch / Core diameter
(0.3 / 0.075 / 0.202)
(0.4 / 0.1 / 0.270)
(0.5 / 0.125 / 0.338)
(0.6 / 0.15 / 0.406)
(0.8 / 0.2 / 0.540)
(1.0 / 0.25 / 0.676)
(1.2 / 0.25 / 0.676)
(1.4 / 0.3 / 1.010)
(1.7 / 0.35 / 1.246)
//END
```

#### Load

The LA (Load Array) function can be used to load an array from another file.

`LA(identifier [, File])`

with:

Identifier	Name of array to be loaded
File	File in which the array is defined

---

#### Note

If an array in the current configuration file must be replaced by an array from another configuration file, then both arrays must have the same name.

---

Extract from file maske.com:

```
DEF Var2 = (S/*ARR5/"Out"/,"Toggle field")
PRESS(HS5)
  LA("ARR5","arrayext.com")           ;Load array ARR5 from file
                                       ;arrayext.com
  VAR2 = ARR5[0]                       ;"Above"/"Below"/"Right"/"Left"
                                       ; appears in VAR2 toggle field
                                       ;instead of "Out/In"

END_PRESS
//A(ARR5)
("Out"/"In")
//END
```

Extract from file arrayext.com:

```
//A(ARR5)
("Top"/"Bottom"/"Right"/"Left")
//END
```

---

### Note

Please note that a valid value must be assigned to a variable after the LA function has been used to assign another array to the toggle field of the variable.

---

## Access

The value of an array access operation can be transferred with property **Value** (*identifier.val*).

Access to one-dimensional array with only one column:

*Identifier* [Z,[M],[C]]            or            *Identifier* [Z,[M],[C]].val

Access to one-dimensional array with only one line:

*Identifier* [S,[M],[C]]            or            *Identifier* [S,[M],[C]].val

Access to two-dimensional array:

*Identifier* [Z,S,[M],[C]]            or            *Identifier* [Z,S,[M],[C]].val

with:

Identifier	Name of array
Z	Line value (line index or search concept)
S	Column value (column index or search concept)
M	Access mode
	0 = Direct
	1 = searches the line, column directly
	2 = line direct, search the column
	3 = searching
	4 = searches the line index
	5 = searches the column index
C	Compare mode
	0 = Search concept must be located in value range of line or column
	1 = Search concept must be located exactly

```
VAR1 = MET_G[REG[3],1,0].VAL
       ;Assign Var1 a value from array MET_G
```



The line index (line number of the array) or column index (column number of the array) each start at 0.

If a line index or column index points to outside the array, the value 0 or a blank string is output and error variable ERR set to TRUE.

Error variable ERR is also set to TRUE if a search concept cannot be found.

### "Direct" access mode

With "Direct" access mode ( $M = 0$ ), the array is accessed with the line index in Z and the column index in S. Compare mode C is not evaluated.

### "Search" access mode

In the case of access mode  $M = 1, 2$  or  $3$ , the search always commences in line 0 or column 0.

### Compare mode

When compare mode  $C = 0$  is used, the content of the search line or search column must be sorted in ascending order. If the search concept is smaller than the first element or greater than the last, the value 0 or a blank string is output and the error variable ERR is set to TRUE.

When compare mode  $C = 1$  is used, the search concept must be present in the search line or search column. If the search concept cannot be found, the value 0 or a blank string is output and the error variable ERR is set to TRUE.

### Access mode

Table 3-5

Mode M	Line value Z	Column value S	Output value
0	Line index	Column index	Value from line Z and column S
1	Search concept, search in column 0	Column index of column from which value is read	Value from line found and column S
2	Line index of line from which return value is read	Search concept, search in line 0	Value from line Z and column found
3	Search concept, search in column 0	Search concept, search in line 0	Value from line and column found
4	Search concept, search in column S	Column index of search column	Line index
5	Line index of search line	Search concept, search in line Z	Column index

## Example

Two arrays are defined below. These are the basis for the following examples.

```
//A(Thread)
(0.3 / 0.075 / 0.202)
(0.4 / 0.1 / 0.270)
(0.5 / 0.125 / 0.338)
(0.6 / 0.15 / 0.406)
(0.8 / 0.2 / 0.540)
(1.0 / 0.25 / 0.676)
(1.2 / 0.25 / 0.676)
(1.4 / 0.3 / 1.010)
(1.7 / 0.35 / 1.246)
//END
```

```
//A(Array2)
( "DES" "PTCH" / "CDM" )
/
(0.3 / 0.075 / 0.202 )
(0.4 / 0.1 / 0.270 )
(0.5 / 0.125 / 0.338 )
(0.6 / 0.15 / 0.406 )
(0.8 / 0.2 / 0.540 )
(1.0 / 0.25 / 0.676 )
(1.2 / 0.25 / 0.676 )
(1.4 / 0.3 / 1.010 )
(1.7 / 0.35 / 1.246 )
//END
```

### Example **Access mode 1:**

The search concept is in Z. This key is always sought in column 0. The value from column S is output with the line index of the search concept found.

VAR1 = Thread[0.5,1,1] ;The value of VAR1 is 0.125

Explanation: Search for value 0.5 in column 0 of "Thread" array and output the value found in column 1 of the same line.

**Example Access mode 2:**

The search concept is in S. This concept is always searched for in line 0. The value from line Z is output with the column index of the search concept found.

```
VAR1 = ARRAY2 [3, "PTCH", 2] ;The value of VAR1 is 0.125
```

Explanation: Search for column containing "PTCH" in line 0 of array "Array2". Output the value from the column found and the line with index 3.

**Example Access mode 3:**

A search concept is in each of Z and S. The line index is searched for in column 0 with the concept in Z and the column index in line 0 with the concept in S. The value from the array is output with the line index and column index found.

```
VAR1 = ARRAY2 [0.6 "PTCH", 3] ;The value of VAR1 is 0.15
```

Explanation: Search for the line with the content 0.6 in column 0 of array "Array2", search for the column with the content "STG" in line 0. Transfer the value from the line and column found to VAR1.

**Example Access mode 4:**

The search concept is in Z. S contains the column index of the column in which concept is being searched for. The line index of the concept found is output.

```
VAR1 = Thread[0.125,1,4] ;The value of VAR1 is 2
```

Explanation: Search for value 0.125 in column 1 of array "Thread" and transfer the line index of the value found to VAR1.

**Example Access mode 5:**

Z contains the line index of line in which concept is being searched for. The search concept is in S. The column index of the search concept found is output.

```
VAR1 = Thread[4,0.2,5,1] ;The value of VAR1 is 1
```

Explanation: Search in line 4 of the "Thread" array for the value 0.2 and transfer the column index of the value found to VAR1. Comparison mode 1 was selected because the values of line 4 are not sorted in ascending order.

**Array status**

The Status property can be used to scan a configuration file to find out whether an array access operation is supplying a valid value. Status is a read-only property.

*Identifier* [Z, S, [M,C]].vld

with:

Identifier Name of array

The result of the scan can be:

FALSE = invalid value

TRUE = valid value

```
DEF MPIT = (R///"MPIT",,"MPIT",""/wr3)
```

```
DEF PIT = (R///"PIT",,"PIT",""/wr3)
```

```
PRESS(VS1)
  MPIT = 0.6
  IF MET_G[MPIT,0,4,1].VLD == TRUE
    PIT = MET_G[MPIT,1,0].VAL
    REG[4] = PIT
    REG[1] = "OK"
  ELSE
    REG[1] = "ERROR"
  ENDIF
END_PRESS
```

### 3.3.12 Table element (grid)

Contrary to the array that displays status data saved in cables, which can then be selected, the values of a table element (grid) are continually updated. This involves a tabular representation of values of system variables that can be addressed using **one** block in **one** channel.

#### Assignment

The table element definition is assigned to a variables definition via a table identifier (see chapter "Screen form elements"). The variables definition determines the values to be displayed and the table element definition determines the appearance and the assignment in the screen window. The grid uses the I/O-field-specific properties from the variables definition line.

#### Screen window

The visible area of the grid is determined by the width and height of the I/O field. Any lines or columns that cannot be seen can be displayed by scrolling horizontally and vertically.

#### Table description

The variable definition will contain a reference to a table description:

```
DEF Identifier =
  1 (Variable type)
  2 [/limit values or toggle field or table identifier]
  3 [/Default]
  4 [/Texts (Long text, Short text|Image, Graphics text, Units text)]
  5 [/Attributes]
  6 [/Help display]
  7 [/System or user variable]
  ....
```

The other parameters correspond to those described in chapter "Screen form elements".

---

#### Note

The small figures above "(" and "/" are merely an aid to assignment and do not form part of the syntax.

---

#### Table identifiers

Identifiers of a table containing NCK/PLC values of the same type, which can be addressed via a channel block. The table identifier is differentiated from limit values or toggle fields by the addition of a % sign in front of it. The file containing the table description can be specified by adding a comma after the identifier and then inserting the name of the file.

#### System or user variable

The parameter remains empty for grid, because the column definition lines contain detailed information about the variables to be displayed (see below).

The table description can be provided in a dynamic format.

### Load Grid

The table description can be made available dynamically **within the LOAD block** using the **LG** method (LG stands for **Load Grid**).

**LG** (*grid name*, *variable name* [,*file name*])

<i>Grid name:</i>	Name of the table (grid) in inverted commas
<i>Variable name:</i>	Name of the variable to which the table is to be assigned, in inverted commas
<i>File name:</i>	Name of the file in which the table (grid) is defined, in inverted commas. Only needs to be specified if the table is not defined within the file that also contains the definition of the variable.

In order to assign a table using LG, the variable must have already been defined as a grid variable and cross-referenced to an existing, valid table (see example).

#### Example:

In this example the "dummygrid" table is first assigned to variable "var1". Depending on the content of the R parameter R[0], either the "grid1" or "grid2" tables, which are defined within the same file as variable "var1", will be loaded In the LOAD block.

```
//M(FORM1/"GRID")  
Def Var1=(R/% dummygrid////////200,75,300,85)
```

```
HS1=("")  
HS2=("")  
HS3=("")  
HS4=("")  
HS5=("")  
HS6=("")  
HS7=("")  
HS8=("")  
VS1=("")  
VS2=("")  
VS3=("")  
VS4=("")  
VS5=("")  
VS6=("")  
VS7=("EXIT",ac7,se1)  
VS8=("")
```

### Load

```
reg[0] = rnp ("R[0]")  
  
if (reg[0] == 0)  
  LG ("grid1", "var1")  
else  
  LG ("grid2", "var1")  
endif
```

### End\_Load

```
PRESS(VS7)  
EXIT  
END_PRESS
```

```
//END
```

#### **//G(grid1/0/5/1,1); (name/type/lines...)**

```
(R/"longtext1","R1 to R4"/wr2/"R[1]"/80/1) ; 1st column, header "R1 to R3",  
; from R1 with offset1  
(R/"long text2","R5 to R8"/wr2/"R[5]"/80/1) ; 2nd column, from R5  
(R/"long text3","R9 to R15"/wr2/"R[9]"/80/2) ; 3rd column, from R9 with offset2,  
; R9 11 13 15  
  
//end
```

#### **//G(grid2/0/5/1,1); (name/type/lines...)**

```
(R/"longtext1","R1 to R4"/wr2/"R[1]"/60/1) ; 1st column, header "R1 to R3",  
; from R1 with offset1  
(R/"long text2","R5 to R8"/wr2/"R[5]"/60/1) ; 2nd column, from R5  
(R/"long text3","R9 to R15"/wr2/"R[9]"/60/2) ; 3rd column, from R9 with offset2,  
; R9 11 13 15  
(R/"long text 4","R9 to R15"/wr2/"R[9]"/60/2) ; 3rd column, from R9 with offset2,  
; R9 11 13 15  
  
//end
```

**//G(dummygrid/0/5/1,1); (name/type/lines...)**

(R//"*longtext1*", "R1 to R4"/wr2//"*\$R[1]*" /80/1) ; 1<sup>st</sup> column, header "R1 to R3",  
 ; from R1 with offset1  
 (R//"*long text2*", "R5 to R8"/wr2//"*\$R[5]*" /80/1) ; 2<sup>nd</sup> column, from R5  
 //end

**Content of the table block**

The table block comprises:

1. A header description
2. 1 to n column descriptions

**Header**

*//G( Table identifier/ Table type/ Number of lines / [Fixed line attribute],[Fixed column attribute])*

Table identifier:

The table identifier is used without a leading % sign. It can only be used once in each screen form.

Table type:

0 (default value)	Standard table for PLC or user data (NCK and channel-specific data)
1	(and others, reserved)

No. of lines:

Number of lines including header

Fixed line attribute:

1	Active
0	Not active

Fixed column attribute:

0	Active
1	Not active

The fixed line or fixed column is not scrolled.

The number of columns is the number of columns configured.

**Column definitions**

Column definition line:

*(type / limit values / empty / long text, column header / attributes / help image / system or OPI variable / column width / offset1, offset2, offset3)*

Type	Data type of the values in the column
Limit values	Limit values for variables: Limit value MIN, limit value MAX The limit values are separated by a comma. Limit values can be entered for types I, C and R in decimal format or as characters in the format "A", "F".

The Limit Values parameter works in conjunction with the Limit attribute (see Attributes, Limit li).  
Default setting: Empty

Long text, column header

Attributes Refer to chapter "Screen form elements".  
Help display Refer to chapter "Screen form elements".

System or OPI variables

The variables should be PLC or NC variables.

The system or user variable appears in double quotation marks.

See chapter "List of accessible system variables".

Indexed variables are usually used for table elements. The increments to be used for the respective index in order to fill the column are specified in the assigned offset parameter.

Column width Column width in pixels

Offset1 Step width for the 1<sup>st</sup> index  
Offset2 Step width for the 2<sup>nd</sup> index  
Offset3 Step width for the 3<sup>rd</sup> index

For PLC or NC variables, the index number is of significance with one or several indices. Only those variables with index are sensibly represented as grid.

### Modifying column properties

The column properties, which can be modified dynamically (written) are:

Limit values (min,max),  
Column header (st),  
Attributes (wr, ac and li),  
Help display (hlp) and  
OPI variable (var).

Column properties are modified via the variable identifier in the definition line and the column index (starting at 1).

Example:

Var1[1].st="Column 1"

Column properties cannot be read in the Load block.

The wr, ac and li attributes can be specified for column definitions.

Refer to chapter "Screen form elements".

### Column header from text file

The column header can be entered as text or text numbers (\$8xxxx) and is not scrolled.



## String variable

If the variable is a string type variable, the string length must be specified in Type.

Example for a user variable:

```
DEF CHAN string [16] TEXT[41]
```

The column definition for the CHAN variable therefore starts, e.g., (S16/...

## Processing the values

The values displayed in a grid can be modified directly by the end user within the restrictions of the rights granted by the attributes and within any limits defined.

The next three examples show the assignment between cells and PLC variables in a standard table (table type=0).

The first line shows the column headers.

### Example 1:

```
//G(MB_TAB/0/4/,1)
(I///,"MB 1 to MB 3"///"MB1"/100/1)
(I///,"MB 4 to MB 6"///"MB4"/100/1)
```

MB 1 to MB 3	MB 4 to MB 6
Value(MB1)	Value(MB4)
Value(MB2)	Value(MB5)
Value(MB3)	Value(MB6)

### Example 2 with offset > 1 in the column definition:

```
//G(MB_TAB/0/4/,1)
(I///," MB1, MB3, MB5"///"MB1"/100/2)
(I///," MB2, MB4, MB6"/// "MB2"/100/2)
Lines and columns are assigned as follows:
```

MB1, MB3, MB5	MB2, MB4, MB6
Value(MB1)	Value(MB2)
Value(MB3)	Value(MB4)
Value(MB5)	Value(MB6)

In each line, the index of the variable was increased by the offset (=2).

### Example 3 with offset and index number in the column definition:

```
//G(MB_TAB/0/4/,1)
(IB///,"M1.1, M2.1, M3.1"/// "M1.1"/100/1)
(IB///,"M1.1, M2.1, M3.1"/// "M1.1"/100/0.1)
```

M1.1, M2.1, M3.1	M1.1, M1.2, M1.3
Value(M1.1)	Value(M1.1)
Value(M2.1)	Value(M1.2)
Value(M3.1)	Value(M1.3)

In the first column, the first variable index for each line was increased by 1.

(Offset1 = 1)

In the second column, the second variable index for each line was increased by 1.

(Offset2 = 1)

### Other system variables

- Ascending numbers can be entered in the first column.

Example: (I///,"Lines"///"0"/60/1)

- Consecutive texts from the language files can be entered in the first column.

Example: (S///,"Lines"///"\$80000"/60/1)

### Focus control in the grid

The Row and Col properties can be used to set and calculate the focus within a table.

Syntax **Row** and **Col** properties:

Identifier.Row  
Identifier.Col

Line properties:

Each cell in a table has the Val and VId properties.

In order to read and write cell properties, a line and column index must be specified in addition to the variable identifiers from the definition list.

Syntax **Val** properties:

Identifier[line index, column index].Val or identifier [line index, column index]

Syntax **VId** properties:

Identifier[line index, column index]. VId

Example:

Var1[2,3].val=1.203

If the line and column indices are not specified, the indices of the focused cell apply.



```
PRESS(RECALL)           ;Start identifier of method
  LM("Form21")          ;Load Form function
END_PRESS               ;End identifier of method
//END                   ;End identifier of softkey
```

## Defining a softkey

Properties are assigned to softkeys in the softkey definition line:

*SK = (Text[, Access level][, Status])*

with:

SK softkey, e.g., HS1 to HS8, VS1 to VS8

Text

**Labeling** of softkeys as described in chapter "Syntax rules in this documentation".

Creation of text: see also chapter "General information about the configuring environment"

**Display:** Refer to chapter "Display images".

Instead of a text, a picture file name is specified in the following form:

"\\my\_pic.bmp" or via a separate text file

\$85199 with the following text in the

(language-specific) text file: E.g.,

85100 0 0 "\\c:\pic\my\_pic.bmp". An image

to be displayed on a softkey may

be 80 \* 34 pixels in size.

---

### Note

A line break in the softkey text is created with %n. A maximum of 2 lines of 10 characters each are available in HMI Advanced and 2 lines of 9 characters each in HMI Embedded sl.

---

Access level

ac0

...

ac7

For a definition, see table "Protection levels".

Default setting: AC7

Status

se1 visible

se2 disabled (gray text)

se3 displayed (last softkey used)

Default setting: se1

```
VS7=("Abort", ac1, se3) ;Define softkey VS7 and assign
                          ;the text "Abort", the access level 1
                          ;and the status "displayed".
```

```
VS8=("OK", ac6, se2) ;Define softkey VS8 and assign
                      ;the text "OK", the access level 6
                      ;and the status "cannot be used".
```

The properties Text, Access Level and Status can be changed in the methods when the application is running:

```
SK.st = Text      "Abort "      ;Softkey, e.g., the text "Abort"
SK.ac = Access level      ;Softkey, e.g., with access level 4
SK.se = Status      ;Softkey, e.g., with status 2
```

with:

Access Level

0

...

7

Status

1 visible

2 disabled (gray text)

3 displayed (last softkey used)

## Assigning access level

The meanings of the different protection levels are as follows:

Table 3-6 Protection levels

Protection level	Locked by	Area
ac0	Password	Siemens
ac1	Password	Machine manufacturer
ac2	Password	Service
ac3	Password	Users
ac4	Keylock switch position 3	Programmer, machine setter
ac5	Keylock switch position 2	Qualified operator
ac6	Keylock switch position 1	Trained operator
ac7	Keylock switch position 0	Semi-skilled operator

ac0 is the highest protection level and ac7 the lowest. The operator can only access information on the level for which he is authorized and all lower levels.

## Example

```
//S(Start)
HS7=("Example", ac7, sel)

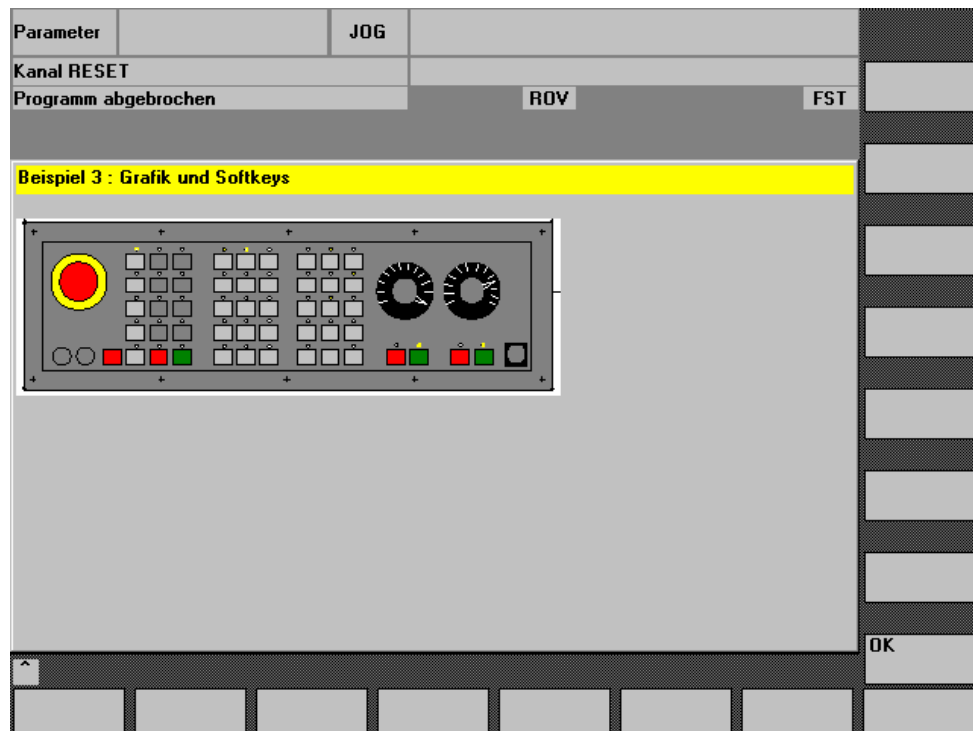
PRESS(HS7)
  LM("Form3")
END_PRESS

//END

//M(Form3/"Example 3 : Graphics and softkeys"/"MST.BMP")
HS1=("")
HS2=("")
HS3=("")
```

```

HS4=("")
HS5=("")
HS6=("")
HS7=("")
HS8=("")
VS1=("")
VS2=("")
VS3=("")
VS4=("")
VS5=("")
VS6=("")
VS7=("")
VS8=("OK", AC7, SE1)
PRESS(VS8)
EXIT
END_PRESS
//END
    
```



Display for example 3: Graphics and softkeys

### <RECALL> key



In addition to the horizontal and vertical softkeys, a <RECALL> key is also available.

In contrast to the softkeys, the RECALL key does not need to be defined. The status and access level properties can be assigned to the key during runtime.

```
PRESS( RECALL )
  RECALL.ac = 1
  LM( "Form5" )
END_PRESS
```

If you do not assign any action (function, calculation of variables, property changes) to <RECALL>, you can use it to exit newly configured user interfaces and return to the standard application.

## Screen form elements and displays

### Note

More information about defining variables for one or more screen form elements and all relevant information about creating display images can be found in chapter "Syntax for screen form definition".

## 3.5 Methods

Various types of events (exit input field, actuate softkey) can initiate particular actions in screen forms and screen-form-dependent softkey menus (softkey menus that are called from a newly configured screen form). These actions are configured in methods.

The following table shows the basic principle used to configure a method:

Method	PRESS( HS1 )
Actions: - Functions, see chapter "Functions"  - Change properties see chapters "Defining a softkey" and "Screen form elements"  - Calculate with variables see chapter "Screen form elements"	LM... LS...  Var1.st = ...  Var2 = Var3+ Var4... EXIT
End of method	END_PRESS

The following methods can be configured in a screen form:

### 3.5.1 PRESS method

The PRESS method is executed when the corresponding softkey is pressed;

**PRESS**(SK)

...

**END\_PRESS**

with:

SK           Name of softkey: HS1-HS8, VS1-VS8,  
              or RECALL plus the following keys:

#### Keys

PU	PageUP	Screen up	
PD	PageDown	Screen down	
SL	ScrollLeft	Cursor key	Left
SR	ScrollRight	"	Right
SU	ScrollUp	"	Up
SD	ScrollDown	"	Down

```
HS1 = ("other softkey menu")
```

```
HS2 = ("no function")
```

```
PRESS (HS1)
```

```
  LS("Menu1")
```

```
  Var2 = Var3 + Var1
```

```
                  ;Load new softkey menu
```

```
END_PRESS
```

```
PRESS (HS2)
```

```
END_PRESS
```

```
PRESS(PU)
```

```
  INDEX = INDEX -7
```

```
  CALL("UP1")
```

```
END_PRESS
```



### 3.5.2 LOAD method

The LOAD method is executed after the variable and softkey definitions (DEF Var1= ..., HS1= ...) have been interpreted. At this time, the screen form is not yet displayed.

#### LOAD

...

#### END\_LOAD

```
LOAD                                ;Start identifier of Load method
  Form1.Hd = $85111                 ;Assign text from language
                                     ;file as screen form header
  VAR1.Min = 0                      ;Assign Limit Value variable MIN
  VAR1.Max = 1000                   ;Assign Limit Value variable MAX
END_LOAD                             ;End identifier
```

### 3.5.3 UNLOAD method

The UNLOAD method is executed before a screen form is unloaded.

#### UNLOAD

...

#### END\_UNLOAD

```
UNLOAD
  REG[1] = VAR1                     ;Place variable in register
END_UNLOAD
```

### 3.5.4 CHANGE method

CHANGE methods are executed if a variable value changes, i.e., variable calculations that are performed as soon as a variable value changes are configured within a CHANGE method.

There are two types of CHANGE method, i.e., element-specific and global. The element-specific CHANGE method is executed if the value of a specified variable changes.

#### CHANGE(*identifier*)

...

#### END\_CHANGE

with:

Identifier                      Name of variable

If a system or user variable is assigned to a variable, cyclic updating of the variable value can be configured in a CHANGE method.

```
DEF VAR1=(S////////"DB20.DBB1")    ;A system variable is assigned
                                     ;to Var1
CHANGE(VAR1)
  IF VAR1.Val <> 1
    VAR1.st="Tool OK!"             ;If the system variable value ≠ 1,
                                     ;then the short text of the variable:
                                     ;Tool OK!
```

```
otto=1
ELSE
  VAR1.st="Caution Error!"           ;If the system variable value = 1,
                                      ;then the short text of the variable:
                                      ;Caution Error!

  otto=2
ENDIF
VAR2.Var=2
END_CHANGE
```

The global CHANGE method is executed if the value of any variable changes and no element-specific CHANGE method has been configured.

#### **CHANGE()**

```
...
END_CHANGE
CHANGE ( )
  EXIT                               ;If any variable value changes,
                                      ;the screen form is closed

END_CHANGE
```

### **3.5.5 FOCUS method**

The FOCUS method is executed if the focus (cursor) is positioned on another field in the screen form.

#### **FOCUS**

```
...
END_FOCUS
FOCUS
  DLGL("The focus has been placed on variable"
  << FOC <<)
END_FOCUS
```

The FOC function may not be initiated as a result of navigation event. The cursor may only be moved in softkey press blocks, change blocks, ... The response to cursor left, right, up, down as well as PGUP, PGDN are permanently pre-defined.

---

#### **Note**

Within the focus block, it is not possible to select a different variable, nor can a new screen form be loaded.

---

### 3.5.6 OUTPUT method

The OUTPUT method is executed if the GC function is called.

**OUTPUT**(*identifier*)

...

**END\_OUTPUT**

with:

Identifier    Name of OUTPUT method

Variables and Help variables are configured as an NC code in an OUTPUT method. The individual elements in a code line are linked by means of blanks.

```
OUTPUT(CODE1)
  "CYCLE82(" Var1.val ", " Var2.val ", " Var3.val ", "
  Var4.val ", " Var5.val ", " Var6.val ") "
END_OUTPUT
```

#### Block numbers and skip identifiers

The OUTPUT block must not contain line numbers or skip identifiers if you wish to keep the line numbers and hide markings directly set with active program support in the parts program in case of recompilations.

Editor changes in the parts program produce the following response:

Condition	Procedure
Number of blocks remains unchanged	Block numbers are retained
Number of blocks is reduced	The highest block numbers are canceled
Number of blocks is increased	New blocks are not numbered

#### Note

The NC code can be generated in an extra file by means of file functions and transferred to the NC. For more information about NC code generation, see chapter "File functions".

### 3.5.7 Version management in conjunction with OUTPUT blocks

Additional variables can be added to existing screen forms when expanding/refining the user interface. A version identifier in parentheses is appended to the additional variables in the definition following the variable name: (0 = Original, is not written), 1 = Version 1, 2 = Version 2, ...

Example:

```
DEF var100=(R//1)                   ;Original, corresponds to Version 0
DEF var101(1)=(S//"Hello")         ;Extension with effect from Version 1
```

When writing the OUTPUT block, you can specify which variables are written, with reference to a particular version identifier.

Example:

OUTPUT(NC1) ;Only the variables of the original version are  
;made available in the OUTPUT block

OUTPUT(NC1,1) ;The variables of the original version and the  
;extensions with version identifier 1 are made  
;available in the OUTPUT block

The OUTPUT block for the original version does not need a version identifier, however you can specify it with 0. OUTPUT(NC1) is equivalent to OUTPUT(NC1,0). Version identifier **n** in the OUTPUT block includes all variables of the originals 0, 1, 2, ... up to and including **n**.

The read-only screen form property CURVER allows the configuration to be adapted in order to handle different versions.

(IF CURVER==1 ...) On reset, CURVER is automatically set with the version of the recompiled code.

The code is automatically generated with the newest version, even if the code was previously recompiled with an older version.

The GC command always generates the newest version.

An additional identifier indicating the generated version is inserted in the user comment of the generated code in versions > 0.

- Variables used previously may not be changed.
- New variables are inserted in the existing (cycle) configuration in arbitrary order.
- It is not permissible to delete variables from a screen form from one version to the next.
- The screen form must contain all variables of all versions.

The newest screen form with all its variables is always displayed.

Example configuration with version identifiers:

```
;Version 0 (default)
//M(XXX)
DEF var100=(R//1)
DEF var101=(S//"Hello")
DEF TMP
```

```
VS8=("GC")
```

```
PRESS(VS8)
  GC("NC1")
END_PRESS
```

```
OUTPUT(NC1)
var100",,"var101
END_OUTPUT
```

```

; ***** Version 1, extended definition *****

//M(XXX)
DEF var100=(R//1)
DEF var101=(S//"Hello")
DEF var102(1)=(V//"HUGO")
DEF TMP

VS8=("GC")

PRESS(VS8)
  GC("NC1")
END_PRESS
...
;Original
OUTPUT(NC1)
var100,"var101"
END_OUTPUT
...
;and the new version in addition

;Version 1
OUTPUT(NC1,1)
var100,"var101"," var102"
END_OUTPUT

```

## 3.6 Functions

A variety of functions are available in screen forms and screen-form-dependent softkey menus. These can be activated by specific events (exit input field, actuate softkey) and configured in methods.

### 3.6.1 Screen form or softkey change

#### LM function

The LM function can be used to load another screen form.

**LM**("Identifier"[,"File"] [,MSx [, VARx] ] )

with:

Identifier	Name of the screen form to be loaded
File	Path name (HMI file system or NC file system) of the configuration file, default setting: Current configuration file
MSx	Screen form change mode
0:	The current screen form is discarded, the new screen form is loaded and displayed. On EXIT (see EXIT function) you will be rerouted to the standard application. You can use the MSx parameter to determine whether or not the current screen form should be terminated when changing screen forms. If the current screen form is retained, variables can be transferred to the new screen form.

A screen form, which calls another screen form but remains open when the second screen form is displayed is known as a master screen form. A screen form called from a master screen form is known as a sub-screen form.

The advantage of the MSx parameter is that when screen forms are replaced, they do not have to be reinitialized every time because the data and layout of the current screen form are retained and data transfer is facilitated.

- 1: The current screen form (master screen form) is interrupted when the LM function is activated, the new screen form (sub-screen form) is loaded and displayed. On EXIT, the sub-screen form is terminated and the program reverts to the interruption point of the master screen form. The UNLOAD block is not processed in the master screen form on interruption.  
Default setting: 0
- VARx List of variables that can be transferred from the master screen form to the sub-screen form.  
Prerequisites: MS1  
Up to 20 variables, each separated by a comma, can be transferred.

---

#### Note

Parameter VARx transfers only the value of the variable in each case, i.e., variables can be read and written in the sub-screen form, but are not visible in it. Variables can be returned from the sub-screen form to the master screen form by means of the EXIT function.

---

```
PRESS(HS1)
  LM("FORM2", "CFI.COM", 1, POSX, POSY,
    DIAMETER)
                                ;Interrupt screen form1 and open screen form2. At the same
                                ;time, transfer variables POSX, POSY and DIAMETER.
  DLGL("Form2 ended")
                                ;On return from screen form2, the following text appears in the
                                ;dialog line of screen form1: Screen form2 ended.
END_PRESS
```

#### LS function

The LS function can be used to display another softkey menu.

**LS**("identifier", "file"[, merge])

with:

- |            |   |
|------------|---|
| Identifier | Name of softkey menu  |
| File       | Path name (HMI file system or NC file system) of configuration file<br>Default setting: Actual configuration file   |
| Merge      | 0: All currently displayed softkeys are deleted - the newly configured softkeys are entered<br>1: Merge, i.e., only newly configured softkeys overwrite existing softkeys, other Softkeys are retained with their |

existing functionality and text  
Default setting: 1

---

#### Note

Even HMI, ShopMill, ShopTurn or ManualTurn standard softkeys can be retained with their existing functionality.

---

Default setting: 0

```
PRESS ( HS4 )
  LS ( "Menu2" , , 0 )      ;Menu2 overwrites existing menu,
                           ;All existing softkeys are deleted
END_PRESS
```

---

#### Note

As long as the interpreter has not displayed a screen form, i.e., no LM function has yet been processed, only one LS or one LM command, but no other action, can be configured in the PRESS methods of the definition blocks for the start softkey and softkey menus.

---



---

#### Note

The LS and LM functions may only be called within a softkey PRESS block and will not react if navigation keys are pressed (PU, PD, SL, SR, SU,SD)

---

## EXIT function

The EXIT function is used to search for master forms. If it finds a master screen form, the currently displayed form is closed and you return to the master screen form. If no master screen form is found, you exit from the newly configured user interfaces and return to the standard application.

#### EXIT

```
PRESS ( HS1 )
  EXIT
END_PRESS
```

If the current screen form has been called as a sub-screen form with transfer variables, the variables in the sub-screen form can be changed and transferred back to the master screen form.

#### EXIT[(VARx)]

with:

VARx

List of variables/variable values that are transferred from the master screen form to the subscreen form.

The variable content/values are assigned to the variables that were transferred to the subscreen form from the master screen form using the LM function.

Up to 20 variables/variable values, each separated by a comma, can be transferred.

---

**Note**

The sequence of variables/variable values must be the same as the sequence of transfer values programmed for the LM function to preclude assignment errors. Any unspecified variable values will not be changed when the transfer is made. The modified transfer variables are immediately valid in the master screen form on execution of the LM function.

---

```
//M(FORM1)
...
PRESS(HS1)
  LM("FORM2","CFI.COM",1, POSX, POSY,
    DIAMETER)
                                ;Interrupt screen form1 and open screen form2. At the same
                                ;time, transfer variables POSX, POSY and DIAMETER.
  DLGL("Form2 ended")
                                ;On return from screen form2, the following text appears in the
                                ;dialog line of screen form1: Screen form2 ended.
END_PRESS
...
//END

//M(form2)
...
PRESS(HS1)
  EXIT(5, , CALCULATED_DIAMETER)
                                ;Exit screen form2 and return to screen form1 in the line after
                                ;LM. At the same time, assign value 5 to variable
                                ;Assign the value of the CALCULATED_DIAMETER variables
                                ;to the DIAMETER variables.
                                ;The variable POSY retains its current value.
END_PRESS
...
//END
```

**EXITLS function**

You can use the EXITLS function to exit the current user interface and load a defined softkey menu.

**EXITLS**("Softkey menu"[, "path name"])

with:

Softkey menu	Name of the softkey menu to be loaded
Path name	Directory path of the softkey menu to be loaded

```
PRESS(HS1)
  EXITLS("Bar1", "AEDITOR.COM" )
END_PRESS
```



## EXE function

The EXE function can be used on HMI Advanced to call a program created as an application with the HMI Advanced OEM package or to start the Free Contour Programming application.

With HMI Embedded sl, EXE can only be used to start the Free Contour Programming application.

**EXE** (*Program name*) ; HMI Advanced or

**EXE** (GPROC) ; HMI Embedded sl

with:

Program name Name of the program that is to be executed

### Note

The EXE function GPROC is only available within the parts program editor.

### Note (HMI Advanced)

In order to start EXE, the task index of the EXE function must be entered under [CHILDS] in the application INI file as it is stored in the REGIE.INI file (see OEM documentation).

```
PRESS(VS3)
  EXE(GPROC) ;Start GPROC.EXE (free
              ;contour programming)
END_PRESS
```

## 3.6.2 PLC/NC variable, write/read MD

The **RNP** (Read NC PLC)/**WNP** (Write NC PLC) commands can be used to read and write NC or PLC variables.

Each time a WNP or RNP function is processed, NC/PLC variables are re-accessed. This means that an NC/PLC access is always executed in a CHANGE method. It is advisable to use this option in cases where a system or user variable changes value frequently. If an NC/PLC variable is to be accessed only once, then it must be configured in a LOAD or UNLOAD method.

### Read

**RNP** ("*System or user variable*")

with:

System or user variable Name of NC or PLC variable

VAR2=RNP( "\$AA\_IN[ 2 ] " ) ;Read NC variable

### Write

**WNP** ("*System or user variable*", *value*)

with:

System or

user variable

Value

Name of NC or PLC variable

Value that is to be written to the system or user variable.

If the value is a String type, it must be enclosed in quotation marks.

WNP( "DB20.DBB1" , 1 ) ;Write PLC variable

### NC variable

All machine data, setting data and R parameters are available, but only certain system variables (see "List of all accessible system variables"). In HMI Advanced, you can find the accessible system variables under operating area "Parameters"/"System variables"/"Edit view"/"Insert variable".

All global and channel-specific user variables (GUDs) can be accessed. but local and program-global user variables cannot be processed.

---

#### Note

User variables may not have the same names as system or PLC variables.

---

Table 3-7

Machine data	
Global machine data	\$MN_...
Axis-specific machine data	\$MA_...
Channel-specific machine data	\$MC_...

Table 3-8

Setting data	
Global setting data	\$SN_...
Axis-specific setting data	\$SA_...
Channel-specific setting data	\$SC_...

Table 3-9

System Variables	
R parameter 1	\$R[1]

### PLC variable

All PLC data are available.

Table 3-10

PLC data	
Byte y bit z of data block x	DBx.DBXy.z
Byte y of data block x	DBx.DBBy
Word y of data block x	DBx.DBWy
Double word y v. of data block x	DBx.DB Dy
Real y of data block x	DBx.DB Ry

PLC data	
Flag byte x bit y	Mx.y
Flag byte x	MBx
Flag word x	MWx
Flag double word x	MDx
Input byte x bit y	Ix.y or Ex.y
Input byte x	IBx or EBx
Input word x	IWx or EWx
Input double word x	IDx or EDx
Output byte x bit y	Qx.y or Ax.y
Output byte x	QBx or ABx
Output word x	QWx or AWx
Output double word x	QDx or ADx
String y with length z from data block x	DBx.DBsy.z

### Read several variables with one command

This MRNP command can be used to transfer several system or OPI variables in a single register access. This access method is significantly faster than reading via individual access attempts.

The system or OPI variables must be included on an MRNP command of the same area.

The areas of the system or OPI variables are organized as follows:

- General NC data (\$MN..., \$SN..., /nck/...)
- Channel-spec. NC data (\$MC..., \$SC..., /channel/...)
- PLC data (DB..., MB..., /plc/...)
- Axis-spec. NC data of the same axis (\$MA..., \$SA..)

Structure of the command:

#### MRNP(Variable name1\*Variable name2[\* ...], Register index)

In the variable names, "\*" is the separator. The values are transferred to register REG[Register index] and those following in the order that the variable names appear in the command.

The following therefore applies:

The value of the first variable is located in REG[Register index].

The value of the second variable is located in REG[Register index + 1]

etc. (see chapter "General screen form variables").

---

#### Notice

Please note that the list of variables must not contain more than 500 lines nor exceed the number of registers specified in chapter "General screen form variables".

---

Example:

```
MRNP("$R[0]*$R[1]*$R[2]*$R[3]",1)
```

In the previous example, REG[1] to REG[4] are described with the value of variables \$R[0] to \$R[3].

### Reading display machine data

Display machine data can be read with RNP (\$MM...) within the LOAD block. General read/write access to display machine data is not possible using the **extend user interface** function.

## 3.6.3 General screen form variables

### Registers

Registers are needed to be able to exchange data between different screen forms. Registers are assigned to a screen form. They are loaded when the first screen form is generated and assigned the value 0 or a blank string.

**REG [x]**

with:

x Number of register with x = 0...19  
Real or String type = Variant

Registers with x ≥ 20 are used by Siemens.

---

#### Note

Registers may not be used directly in OUTPUT blocks for generating NC code.

### Register value

The assignment of values to registers is configured in a method.

*Identifier.val = Register value*

or

Identifier = Register value

with:

Identifier	Name of register
Register value	Value of register

```
UNLOAD
  REG[0] = VAR1           ;Assign value of variable 1 to assign
END_UNLOAD
```

```
UNLOAD

  REG[9].VAL = 84        ;Assign value 84 to register 9
END_UNLOAD
```

These registers can then be assigned to local variables again in the next screen form.

```
LOAD
  VAR2 = REG[0]
END_LOAD
```

---

### Note

If a new form is generated from an existing form by means of the LM function, the content of registers are automatically transferred to the new screen form at the same time and so are available for further calculations in the second screen form.

---

## Register status

The Status property can be used to scan a register for valid content. Status is a read-only property.

### *Identifier.vld*

with:

Identifier    Name of register

The result of the scan can be:

FALSE = invalid value

TRUE = valid value

One possible use for the register scan function is to ensure that a value is written to a register only if the relevant screen form is a "master screen form".

```
IF REG[15].VLD == FALSE   ;Scan validity of register value
  REG[15] = 84
ENDIF
VAR1 = REG[9].VLD         ;Assign Var1 the value of the status scan
                           ;of REG[9]
```

## CFI variable FILE\_ERR

Refer to chapter "File functions".

## Error variable ERR

Variable ERR can be used to check whether the preceding lines have been executed correctly.

### ERR

The result of the scan can be:

FALSE = previous line was executed error-free

TRUE = previous line was not executed error-free

The ERR variable is read-only.

```
VAR4 = Thread[VAR1, "KDM", 3]           ;Output array value
IF ERR == TRUE                          ;Scan to check whether value has
                                        ;been found in array

    VAR5 = "Array accessing error"      ;If the value has not been found
                                        ;in the array, the value "All OK"
                                        ;accessing error" is assigned to the
                                        ;variable

ELSE                                     ;If the value has been found
    VAR5 = "All OK"                    ;in the array, the value "All OK"
                                        ;is assigned to the variable

ENDIF
```

## FOC - Focus

The input focus (current active I/O field) in a screen form is controlled by the FOC function. FOC can be read and written.

### FOC

When FOC is read, the result is the name of the focused variable.

```
IF FOC == "Var1"                        ;Read focus
    REG[1] = Var1
ELSE
    REG[1] = Var2
ENDIF
```

When FOC is written, it is possible to assign either a string or a numerical value. A string is interpreted as a variable name and a numerical value as a variable index. The FOC function may not be initiated as a result of navigation event. The cursor may only be moved in softkey press blocks, change blocks, ... The response to cursor left, right, up, down as well as PGUP, PGDN are permanently pre-defined.

---

### Note

The FOC function cannot be applied to variables with input mode wr = 0 and wr = 4 or to Help variables.

---

```
FOC = "Var1"                            ;Input focus is assigned to variable 1
FOC = 3                                  ;Input focus is assigned to the 3rd screen form
                                        ; element with WR ≥ 2.
```

## Entry

The ENTRY variable can be used to check by what method a screen form has been called.

### ENTRY

The result of the scan can be:

- 0 = No programming support
- 1 = Programming support (screen form called under programming)
- 2 = Programming support + default from previous form screen  
(Subscreen form)
- 3 = Programming support + recompilation
- 4 = Programming support + recompilation with generated  
comments and # characters
- 5 = Programming support + recompilation with generated comments,  
but no # characters

The ENTRY variable is a read only variable.

```
IF ENTRY == 0
  DLGL("The screen form was not called under Programming")
ELSE
  DLGL("The screen form was called under Programming")
ENDIF
```

## S\_CHAN

Variable n S\_CHAN can be used to determine the number of the current channel for display or any evaluation purposes.

## CURPOS

Using the CURPOS variable, it is possible to call or manipulate the position of the cursor in the active input field of the current screen form. The variable indicates how many **characters** are located **in front of the cursor**. If the cursor is located at the start of the input field, then CURPOS assumes the value of 0. If the value of CURPOS is changed, then the cursor is positioned at the appropriate location in the input field.

In order to be able to respond to changes in the variable value, it is possible to monitor for changes using a CHANGE block. If the value of CURPOS changes, then a jump is made to the CHANGE block and the instructions contained there are executed.

### 3.6.4 General functions

#### Check variables (CVAR)

Using the function CVAR (CheckVariable) it is possible to interrogate as to whether all or only certain variables or Help variables of a screen form are error-free. A variable is error-free if the state of the variable identifier.vld = 1.

Check all variables for valid content:

#### CVAR

Check specific variables for valid content:

#### CVAR(VarN)

with:

**VarN** List of variables to be checked. A total of 29 variables, separated by commas, can be checked. The maximum character length of 500 must not be exceeded.

The result of the scan can be:

- 1 = TRUE (all variables have valid content)
- 0 = FALSE (at least one variable has invalid content)

It may be useful to check if variables contain a valid value before an NC code with the GC function.

```
IF CVAR == TRUE           ;Check all variables
  VS8.SE = 1             ;If all variables are correct, softkey VS8 will be visible
ELSE
  VS8.SE = 2             ;If one variable contains an invalid value, softkey VS8
                          ;cannot be used
ENDIF
```

```
IF CVAR("VAR1", "VAR2") == TRUE
                          ;Check variables VAR1 and VAR2
  DLGL ("VAR1 and VAR2 are OK")
                          ;If the content of VAR1 and VAR2 are correct,
                          ;"VAR1 and VAR2 are OK" appears in the dialog line
                          ;(See DLGL function)
```

```
ELSE
  DLGL ("VAR1 and VAR2 are not OK")
                          ;If the content of VAR1 and VAR2 are incorrect,
                          ;"VAR1 and VAR2 are not OK" appears in the
                          ;dialog line (see DLGL function)
ENDIF
```

### Dialog line (DLGL)

It is possible to configure short texts (messages or input tips) for output in the dialog line in response to certain situations.

If default font size is selected:

With HMI Embedded sl	approximately 50 characters in the dialog line
With HMI Advanced	approximately 100 characters in the dialog line

#### **DLGL ("String")**

with:

**String** Text, which is displayed in the dialog line

```
IF Var1 > Var2
  DLGL("Value too high!") ;The text "Value too high!" appears
                          ;in the dialog line if variable1 > variable2.
ENDIF
```



## Evaluate expression (EVAL)

The EVAL function evaluates a transferred expression and then executes it. With this function, expressions can be programmed during runtime. This can be useful, for example, for indexed access operations to variables.

**EVAL**(exp)

with:

exp ;Logical expression to be evaluated

VAR1=(S)

VAR2=(S)

VAR3=(S)

VAR4=(S)

CHANGE( )

```
REG[7] = EVAL("VAR"<<REG[5])
```

;The expression in parentheses  
;is the value of VAR3 if the value of  
;REG[5] equals 3. REG[7] is  
;assigned the value of VAR3.

END\_CHANGE

The above-mentioned function is configured as follows:

VAR1=(S)

VAR2=(S)

VAR3=(S)

VAR4=(S)

CHANGE( )

```
IF REG[5] == 1
  REG[7] = VAR1
ELSE
  IF REG[5] == 2
    REG[7] = VAR2
  ELSE
    IF REG[5] == 3
      REG[7] = VAR3
    ELSE
      IF REG[5] == 4
        REG[7] = VAR4
      ENDIF
    ENDIF
  ENDIF
ENDIF
END_CHANGE
```

### 3.6.5 Subprograms

Repeatedly used configuring instructions or others, which define the process for a particular operation can be configured in subprograms. Subprograms can be loaded into the main program or other subprograms at any time and executed as often as necessary, i.e., the instructions they contain do not need to be configured repeatedly. The definition blocks of the screen forms/softkey menu constitute a main program.

#### Defining a subprogram

Subprograms are identified by the block identifier //B and terminated with //END in a configuration file. Several subprograms can be defined under each block identifier. A block is structured in the following way:

```
//B(Block name)
SUB (identifier)
...
END_SUB
[SUB(identifier)
...
END_SUB]
...
//END
with:
Block name      Name of block identifier
Identifier      Name of subprogram
```

---

#### Note

Variables programmed in subprograms must be defined in the screen form in which these subprograms are called.

---

```
//B(PROG1)           ;Beginning of block
SUB(UP1)            ;Subprogram start
...
REG[0] = 5          ;Assign the value 5 to register 0
...
END_SUB             ;End of subprogram
SUB(UP2)            ;Start of the subprogram
IF VAR1.val=="Otto"
  VAR1.val="Hans"
  RETURN
ENDIF
VAR1.val="Otto"
END_SUB             ;Subprogram end
//END               ;Block end
```

## Loading a block (LB)

The LB (LoadBlock) function can be used to load blocks containing subprograms during operation.

LB should be configured in a LOAD method so that the loaded subprograms can be called at any time.

**LB**("Block name"[,"File"])

with:

Block name	Name of block identifier
File	Path name of configuration file
	Default setting: Current configuration file

LOAD

```
LB ( " PROG1 " ) ;Block "PROG1" is located in the current configuration file
;and then loaded
```

```
LB ( " PROG2 " , "XY.COM" )
;Block "PROG2" is located in configuration file XY.COM
;and then loaded
```

END\_LOAD

---

### Note

Subprograms can also be defined directly in a screen form so that they do not have to be loaded.

---

## Indirect subprogram call (CALL)

The CALL function can be used to call a loaded subprogram from any point in a method.

Subprogram nesting is supported, i.e., you can call a subprogram from another subprogram.

**CALL**("Identifier")

with:

Identifier	Name of subprogram
------------	--------------------

```
//M(FORM1)
```

```
VAR1 = ...
```

```
VAR2 = ...
```

```
LOAD
```

```
...
```

```
LB ( " PROG1 " ) ;Load block
```

```
...
```

```
END_LOAD
```

```
CHANGE ( )
```

```
...
```

```
CALL ( " UP1 " ) ;Call and execute subprogram
```

```
...
```

```
END_CHANGE
```

```
...
```

```
//END
```

## Aborting the current subprogram (RETURN)

The RETURN function can be used to prematurely terminate execution of the current subprogram and to return to the branch point of the last CALL command. If no RETURN command is configured in the subprogram, the branch point is not reached again until the end of the current subprogram.

### RETURN

```
//B(PROG1)           ;Block start
SUB(UP2)            ;Subprogram start
  IF VAR1.val=="Otto"
    VAR1.val="Hans"
    RETURN          ;If the variable value = Otto, the value Hans is assigned
                   ;to the variable and the subprogram is terminated at
                   ;this point
  ENDIF
  VAR1.val="Otto"   ;If the variable value ≠ Otto, the value Otto is assigned
                   ;to the variable
END_SUB            ;Subprogram end
//END              ;Block end
```

## REFRESH function

The function has no parameters. It can be called in any block (e.g., PRESS ... END\_PRESS).

### Effect

1. All active variable content (input/output fields) in the display range are output again with the background and foreground.
2. All active and visible short description texts, graphic texts and unit texts are output again without clearing the text background first.

### 3.6.6 File functions

File functions can be used to generate NC code from an OUTPUT method and transfer it for execution to the NC.

#### Generating NC code

The GC (Generate Code) function generates NC code from the OUTPUT method.

```
GC("identifier"[,"target file"][,Opt],[append])
```

Identifier                      Name of OUTPUT block from which code is generated

Target file                      Path name of target file for HMI or NC file system

If the target file is not specified (only possible within programming support system), the code will be written to the location of the cursor within the file that is currently open.

Opt                                      Option for generating comments

0: Default                      Generate code with comment for the purpose of recompilability.

1:                                      Do not create comments in the generated code.

**Note:** This code cannot be recompiled.

Append                                      This parameter is only relevant if a target file is specified.

0: Default                      If the file already exists, the old content is deleted.

1:                                      If the file already exists, the new code is written at the start of the file.

2:                                      If the file already exists, the new code is written at the end of the file.

#### No entry for target file:

The GC function can only be used in the Programming Support system and writes the NC code to the file currently open in the Editor. Recompilation of the NC code is possible.

If the GC function is configured without specification of a target file under **Expand user interface**, an error message is output when it is executed.

#### Entry for target file:

The code generated from the OUTPUT block is transferred to the target file. If the target file does not already exist, it is set up in the NC file system. If the target file is stored in the HMI file system, it is stored on the hard disk (HMI Advanced only).

User comment lines (information required to recompile code) are not set up, i.e., the code cannot be recompiled.

## Special considerations for target file specification

In principle, there are three different ways of specifying a target file:

- **NC notation:**  `/_N_MPF_DIR/_N_MY_FILE_MPF`  
Only possible with HMI Embedded sl.  
The file is created in the MPF directory on the NC.
- **DH notation:**  `/MPF.DIR\MY_FILE.MPF`  
Possible with both HMI Advanced and HMI Embedded sl.  
In the case of HMI Embedded sl, the target file specification is converted into NC notation and the file is created on the NC.  
In the case of HMI Advanced/HMI Embedded WIN32, the file is created in the data management path.
- **DOS notation:**  `d:\abc\my_file.txt` or  `\\RemoteRechner\files\my_file.txt`  
Possible with both HMI Advanced and HMI Embedded sl  
The file is written to the specified directory on the hard disk/the specified PC provided that the directory is available on the hard disk/remote PC.  
In the case of HMI Embedded sl this notation can only be used to write to the RAM drive or to a networked PC provided that a network connection has been configured.

---

### Note

Invalid variables generate a blank string in generated NC code and an error message in the log book when they are read.

---

### Note

The GC function cannot be called in sub-forms because variables originating from master forms can be used in sub-forms. These variables would not, however, be available in response to a direct call.

When generated code is processed manually with the Editor, the number of characters for values created by the code generation program must not be changed. Changing these values would make it impossible to recompile the code.

Remedy:

Recompile,

Enter changes using the configured screen form (e.g., 99 → 101)

GC

---

```

//M(TestGC/"Code generation:")
DEF VAR1 = (R//1)
DEF VAR2 = (R//2)
DEF D_NAME
LOAD
  VAR1 = 123
  VAR2 = -6
END_LOAD
OUTPUT(CODE1)
  "Cycle123(" VAR1 " ," VAR2 ") "
  "M30 "
END_OUTPUT

RESS(VS1)
  D_NAME = "MPF.DIR\MESSEN.MPF"
  GC("CODE1", D_NAME)           ;NC code:
                                Cycle123(123, -6)
                                M30
                                write from the OUTPUT method to
                                file C:\MPF.DIR\MESSEN.MPF

END_PRESS

```

---

**Note**

With the following functions, DOS paths (e.g., \MPF.DIR\TEST.MPF) are used for HMI Embedded sl and HMI Advanced.

---

**Copy Program (CP)**

The CP (Copy Program) function copies files within the HMI file system or NC file system.

With HMI Embedded on NCU, copying is only possible within the NC file system.

**CP**("Source file", "Target file")

with:

Source file Complete path data of the source file

Target file Complete path data of the target file

```
CP("\MPF.DIR\CFI.MPF ", "\spf.dir\cfi.nc")
```

**CFI variable FILE\_ERR**

Variable FILE\_ERR can be used to check whether the preceding GC or CP command has been executed correctly.

Possible results are:

- 0 .        Operation okay
- 1 :        Drive/path not available
- 2 :        Path/file access error
- 3 :        Drive not ready

- 4 : Incorrect file name
- 5 : File is already open
- 6 : Access denied
- 7 : Target path not available or not permitted
- 8 : Copy source same as target
- 10: Internal error:  
FILE\_ERR = 10 means that the error cannot be classified in the other categories.

The FILE\_ERR variable is read-only.

**Example:**

```

CP("D:\source.mpf","E:\target.mpf") ;Copy from source.mpf to E:\target.mpf
IF FILE_ERR > 0 ;Query whether error has occurred
  IF FILE_ERR == 1 ;Query specific error numbers
    ;and output associated error text
    VAR5 = "Drive/path not available"
  ELSE
    IF FILE_ERR == 2
      VAR5 = "Path/file access error"
    ELSE
      IF FILE_ERR == 3
        VAR5 = "Wrong file name"
      ENDIF
    ENDIF
  ENDIF
ELSE
  VAR5 = "All OK" ;If no errors have occurred in CP (or GC)
  ;output of
  "All OK"
ENDIF

```

**Activate Program (AP)**

The AP (Activate Program) function transfers a file from the passive HMI file system to the active NC file system. The file is loaded into the NC and enabled and then deleted in the HMI file system. With HMI Embedded sl, this function has the same effect as Set enable.

**AP("File")**

with:

File Complete path name of HMI file to be transferred

```

//M(TestGC/"Code generation:")
DEF VAR1 = (R//1)
DEF VAR2 = (R//2)
DEF D_NAME
LOAD
  VAR1 = 123
  VAR2 = -6
END_LOAD
OUTPUT(CODE1)
  "Cycle123(" VAR1 "," VAR2 ")"
  "M30"
END_OUTPUT

```



```

PRESS(VS1)
  D_NAME = "\MPF.DIR\MESSEN.MPF"
  GC("CODE1",D_NAME) ;Write code from OUTPUT method to
                        ;file \MPF.DIR\MESSEN.MPF
END_PRESS
PRESS(HS8)
  D_NAME = "\MPF.DIR\MESSEN.MPF"
  AP(D_NAME)          ;Load file into NC
END_PRESS

```

### Passivate Program (PP)

The PP (Passivate Program) function transfers a file from the active file system on the NC to the passive file system on HMI Advanced. Once the PP function has been executed, the file is no longer present in the active file system of the NC. With HMI Embedded sl, this function has the same effect as Delete enable.

**PP("file")**  
with:  
File            Complete path name of NC file to be transferred

```
PP("\MPF.DIR\MESSEN.MPF")
```

### Select Program (SP)

The SP (Select Program) function selects a file in the active NC file system for execution, i.e., the file must be loaded into the NC beforehand.

**SP("file")**  
with:  
File            Complete path name of NC file

```

//M(TestGC/"Code generation:")
DEF VAR1 = (R//1)
DEF VAR2 = (R//2)
DEF D_NAME
LOAD
  VAR1 = 123
  VAR2 = -6
END_LOAD
OUTPUT(CODE1)
  "Cycle123(" VAR1 "," VAR2 ")"
  "M30"
END_OUTPUT
PRESS(VS1)
  D_NAME = "\MPF.DIR\MESSEN.MPF"
  GC("CODE1",D_NAME) ;Write code from OUTPUT method to
                        ;file \MPF.DIR\MESSEN.MPF
END_PRESS
PRESS(HS8)
  AP("\MPF.DIR\MESSEN.MPF") ;Load the file into the NC
  SP("\MPF.DIR\MESSEN.MPF") ;Select file
END_PRESS

```

### Delete Program (DP)

The DP (Delete Program) function deletes a file from the passive HMI or active NC file system.

**DP**("file")

with:

File            Complete path name of file to be deleted

```
DP ( "\MPF.DIR\CFI.MPF" )
```

### Exists Program (EP)

The EP (Exists Program) function checks whether a particular file is stored in the specified path in the NC or HMI file system.

**EP**("File", Result)

with:

File            Complete path name of file for  
                 NC file system or HMI file system

Result         Name of a variable to which the result of the scan  
                 is to be assigned

The result of the scan can be:

M = File is stored on HMI

N = file is stored on NC

Blank string = The file neither exists on the HMI nor on the NC

```
EP ( "\MPF.DIR\CFI.MPF" , VAR1 )         ;Check whether file CFI.MPF  
                                         ;HMI file system exists  
IF VAR1 == "M"  
  DLGL("File is located in HMI file system")  
ELSE  
  IF VAR1 == "N"  
    DLGL("File is located in the NC file directory")  
  ELSE  
    DLGL("File is neither in the HMI nor in the NC  
          file directory")  
  ENDIF  
ENDIF
```

### 3.6.7 NC Functions (PI services)

The PI\_SERVICE function can be used to start PI Services (Program Invocation Services) under HMI Embedded sl from the PLC in the NC area.

---

#### Note

Channel-dependent PI Services always refer to the current channel.  
PI Services, which require the TO area (PI Services of tool functions) always refer to the TO area allocated to the current channel.

---

#### PI\_SERVICE (*service, n parameter*)

with:

Service	Identifier of PI service
n parameter	List of n parameters of PI Service. The parameters are separated by commas.

For a description of available PI Services, refer to chapter "List of PI services".

```

PRESS (HS2)
  PI_SERVICE ( "_N_CREATO" , 55 )
END_PRESS
PRESS (VS4)
  PI_SERVICE ( "_N_CRCEDN" , 17 , 3 )
END_PRESS

```

### PI\_START command (6.3)

The PI\_START command executes a PI service based on OEM documentation.

Structure of the command:

#### PI\_START(*transfer string*)

Unlike the OEM documentation, the transfer string should be entered in inverted commas.

Example:

```
PI_START("/NC,001,_N_LOGOUT")
```

### 3.6.8 External functions (only HMI Advanced)

Additional, user-specific functions can be integrated by means of external functions. External functions are stored in a DLL file and identified by an entry in the definition lines of the configuration file.

**FCT** *Function name = ("File"/Type of return/Types of fixed call parameters/Types of variable call parameters).*

with:

Function name	Name of external function
File	Complete path name of DLL file

Type of return	Data type of return value for function
Type of fixed call parameter	List of data types of fixed call parameters. The data types are separated by commas.
Type of variable call parameter	List of data types of variable call parameters (variable or register). The data types are separated by commas. The data types of return values and call parameters are R, I, S, C, B.
Use of external functions in the further configuration: <i>Function name (Call parameter)</i> with:	
Function name	Name of external function
Call parameter	List of all call parameters. Individual parameters are separated by commas.

**Note**

An external function must have at least one return parameter.

```
//M(FORM1)
DEF VAR1 = (R)
DEF VAR2 = (I)
DEF RET = (I)
FCT InitConnection = ("c:\user\mydll.dll"/I/R,I,S/I,S)
                    ;The external function "InitConnection"
                    ;is declared. The data type of
                    ;the return value is Integer, the data types
                    ;of the fixed call parameters are Real,
                    ;Integer and String, the data types of the
                    ;variable call parameters are Integer
                    ;and String.

LOAD
  RET = InitConnection(VAR1+SIN(VAR3),13,"Servus",VAR2,
  REG[2])
                    ;RET is assigned the value of the external
                    ;function "InitConnection" with call parameters
                    ;VAR1+SIN(VAR3), 13, Servus, VAR2, and
                    ;REG[2].

END_LOAD
```

**Extract from DLL file:**

```
void __export WINAPI InitConnection(ExtFctStructPtr FctRet,
ExtFctStructPtr FctPar, char cNrFctPar)
FctRet->value.i      Return value of function
FctPar[0]->value.r   1. parameter (VAR1+SIN(VAR3))
FctPar[1]->value.i   2. parameter (13)
FctPar[2]->value.s   3. parameter ("Servus")
FctPar[4]->value.i   4. parameter (reference to VAR2)
FctPar[5]->value.s   5. parameter (reference to REG[2])
cNrFctPar           Number of parameters (5)
```

### 3.6.9 Operators and functions

#### Mathematical operators

Table 3-11

Mathematical operators	
<b>+</b>	Addition
<b>-</b>	Subtraction
<b>*</b>	Multiplication
<b>/</b>	Division
<b>MOD</b>	Modulo operation
<b>( )</b>	Parentheses
<b>AND</b>	AND operator
<b>OR</b>	OR operator
<b>NOT</b>	NOT operator
<b>ROUND</b>	Round off numbers with decimal places

Example:

```
VAR1.VAL = 45 * (4 + 3)
```

#### ROUND command

The ROUND command is used to round off numbers with up to 12 decimal places during execution of a screen form configuration. The variable fields cannot accept the decimal places in the display.

ROUND is controlled by the user with two parameters:

**Example:**

```
VAR1 = 5.2328543
```

```
VAR2 = ROUND( VAR1, 4 )
```

Result:

```
VAR2 = 5.2339
```

VAR1 contains the number to be rounded. The parameter "4" indicates the number of decimal places in the result, which is placed in VAR2.

#### Bit-serial operators

Bit operators	
<b>BOR</b>	Bit-serial OR
<b>BXOR</b>	Bit-serial XOR
<b>BAND</b>	Bit-serial AND
<b>BNOT</b>	Bit-serial NOT
<b>SHL</b>	Shift bits to left
<b>SHR</b>	Shift bits to right

## SHL operator

Syntax:  
variable = *value* SHL *increment*

*value*                    value to be shifted  
*increment*                number of shift increments

Bits are shifted to the left using the function **SHL**. You can specify both the value to be shifted and the number of shift increments directly or via a variable. If the limit of the data format is reached, the bits are shifted beyond the limit without displaying an error message.

### Example:

```
PRESS(VS1)
  VAR01 = 16 SHL 2                    ;Result = 64
  VAR02 = VAR02 SHL VAR04           ;Convert content of VAR02 to 32-bit unsigned
                                   ;and shift content to left by number of bits
                                   ;specified in VAR04. Then convert 32-bit value
                                   ;back to format of variable VAR02.
END_PRESS
```

## SHR operator

Syntax:  
variable = *value* SHR *increment*

*value*                    value to be shifted  
*increment*                number of shift increments

Bits are shifted to the right using the function **SHR**. You can specify both the value to be shifted and the number of shift increments directly or via a variable. If the limit of the data format is reached, the bits are shifted beyond the limit without displaying an error message.

### Example:

```
PRESS(VS1)
  VAR01 = 16 SHR 2                   ;Result = 4
  VAR02 = VAR02 SHR VAR04           ;Same effect as SHL but shift to right
```

## Mathematical functions

Table 3-12

Mathematical functions	
<b>SIN(x)</b>	Sine of x
<b>COS(x)</b>	Cosine of x
<b>TAN(x)</b>	Tangent of x
<b>ATAN(x, y)</b>	Arc tangent of x/y
<b>SQRT(x)</b>	Square root of x
<b>ABS(x)</b>	Absolute value of x
<b>SDEG(x)</b>	Conversion to degrees
<b>SRAD(x)</b>	Conversion to radian

---

### Note

The functions operate with radian measure. The functions SDEG() and SRAD() can be used for conversion.

---

```
VAR1.VAL = SQRT(2)
```

Table 3-13

Constants	
<b>PI</b>	3.14159265358979323846
<b>FALSE</b>	0
<b>TRUE</b>	1

```
VAR1.VAL = PI
```

Table 3-14

Comparison operators	
<b>==</b>	Equal to
<b>&lt;&gt;</b>	Not equal to
<b>&gt;</b>	Greater than
<b>&lt;</b>	Less than
<b>&gt;=</b>	Greater than or equal to
<b>&lt;=</b>	Less than or equal to

```
IF VAR1.VAL == 1
  VAR2.VAL = TRUE
ENDIF
```

## Conditions

Condition with a command:

```
IF  
...  
ENDIF
```

Condition with two commands:

```
IF  
...  
ELSE  
...  
ENDIF
```

The nesting depth is unlimited.

### 3.6.10 Use of strings

#### Strings

Strings can be processed as part of the configuration. These allow text to be displayed dynamically or different texts to be chained for the purpose of code generation.

The following rules must be observed with regard to string variables:

- Logic operations are processed from left to right.
- Nested expressions are solved from the inside outwards.
- No distinction is made between uppercase and lowercase type.

Strings can be deleted simply by assigning a blank string.

Strings can be attached after the equal sign using the operator "<<". Quotation marks (") in the string are represented by two successive quotation mark symbols. Strings can be checked for equality in IF instructions.

#### Examples

Default settings for the following examples:

```
VAR1.VAL = "This is an"  
VAR8.VAL = 4  
VAR14.VAL = 15  
VAR2.VAL = "error"  
$85001 = "This is an"  
$85002 = "alarm text"
```

Chaining of strings:

```
VAR12.VAL = VAR1 << " Error." ;Result: "This is an error"
```

Deleting a variable:

```
VAR10.VAL = "" ;Result: Blank string
```

Setting a variable with a text variable:

```
VAR11.VAL = VAR1.VAL ;Result: "This is an"
```



Data type matching:

```
VAR13.VAL = "This is the " << (VAR14 - VAR8) << "11th error"
;Result: "This is the 11th error"
```

Treatment of numerical values:

```
VAR13.VAL = "Error " << VAR14.VAL << ": " << $T80001 <<
$T80002 ;Result:
; "Error 15: This is an alarm text"
IF VAR15 == "Error" ;Strings in IF instruction
VAR16 = 18.1234 ;Result: VAR16 equals 18.1234,
;if VAR15 equals "Error"
ENDIF
```

Quotation marks within a string:

```
VAR2="Hello, this is a " " Test"
;Result: Hello, this is a " Test"
```

System or user-variable strings dependent on variable content:

```
VAR2.Var = "$R[" << VAR8 << "]"
;Result: $R[4]
```

## String functions

The following string functions are available:

- Determine length of string
- Find a character in a string
- Extract substring from left
- Extract substring from right
- Extract substring from mid-string
- Replace substring

## String length

Syntax:

**LEN**(*string* | *varname*)

Parameters:

*string* Every valid string expression. NULL is output if string is empty

*varname* Any valid declared variable name

**Function:**

Determine the number of characters in a string.

**Note:**

Only one of the two parameters is allowed.

**Example:**

```
DEF VAR01
DEF VAR02

LOAD
  VAR01="HELLO"
  VAR02=LEN(VAR01)                ;Result = 5
END_LOAD
```

**Find a character**

Syntax:

**INSTR**(*start*, *string1*, *string2* [,*direction*])

Parameters:

<i>start</i>	Number, position from which <i>string1</i> is searched for in <i>string2</i> . This parameter defines the start position for the search. Enter 0 to start searching at the beginning of <i>string2</i> .
<i>string1</i>	Character that is being searched for.
<i>string2</i>	Chain of characters in which the search is being made
<i>direction</i>	Optional. Defines the search direction. 0: From the left to the right, 1: From the right to the left. If not specified, then <i>direction</i> = 0

**Function:**

**INSTR** returns the position of the first occurrence of a specified character in a string.

**Note:**

0 is returned if *string1* does not occur in *string2*.

**Example:**

```
DEF VAR01
DEF VAR02

LOAD
  VAR01="HELLO/WORLD"
  VAR02=INSTR(1,"/",VAR01)        ;Result = 6
END_LOAD
```

## Substring from left

Syntax:

**LEFT**(*string*, *length*)

Parameters:

<i>string</i>	Character string or variable with the string to be processed
<i>length</i>	Number of characters that are to be read out

**Function:**

**LEFT** returns a string containing the specified number of characters starting from the left-hand side of a string.

**Example:**

```
DEF VAR01
DEF VAR02
LOAD
  VAR01="HELLO/WORLD"
  VAR02=LEFT(VAR01,5)           ;Result = "HELLO"
END_LOAD
```

## Substring from right

Syntax:

**RIGHT**(*string*, *length*)

Parameters:

<i>string</i>	Character string or variable with the string to be processed
<i>length</i>	Number of characters that are to be read out

**Function:**

**RIGHT** returns a string containing the specified number of characters starting from the right-hand side of a string.

**Example:**

```
DEF VAR01
DEF VAR02

LOAD
  VAR01="HELLO/WORLD"
  VAR02=RIGHT(VAR01,4)        ;Result = "WORLD"
END_LOAD
```

## Substring from mid-string

Syntax:

**MIDS**(string, start [, length])

Parameters:

*string* Character string or variable with the string to be processed

*start* Start from where characters are to be read in the string

*length* Number of characters that are to be read out (optional)

If *length* is not specified, then the string is transferred up to the end.

**Function:**

**MIDS** returns a string containing the specified number of characters starting at the specified position in the string.

**Example:**

```
DEF VAR01
```

```
DEF VAR02
```

```
LOAD
```

```
VAR01="HELLO/WORLD"
```

```
VAR02=MIDS(VAR01,4,4)
```

```
;Result = "LOW"
```

```
END_LOAD
```

## Replacing characters

Syntax

**REPLACE** ( String, FindString, ReplaceString [, Start [, Count ] ] )

Parameters:

*String* String is to be replaced by the ReplaceString

*FindString* String to be replaced

*ReplaceString* Replacement string (is used instead of the FindString)

*Start* Start position from which a search is made or replacement made

*Count* Number of characters that are to be searched from the Start position after the FindString

**Function:**

The **REPLACE** function replaces a character/string in a string by another character/string. It is similar to the basic function of the same name.

**Return value:**

**If...**

**Return value**

*String* = Blank string

Copy of *String*

*FindString* = Blank string

Copy of *String*

*ReplaceString* = Blank string

Copy of *String*, in which all occurrences of *FindString* are deleted

*Start* > Len(*String*)

Blank string

*Count* = 0

Copy of *String*

### 3.6.11 Functions for start softkeys

Start softkeys can be used to initiate certain functions.

The following functions are possible:

- The LM function can be used to load another screen form.  
**LM**("Identifier","File")
- The LS function can be used to display another softkey menu. **LS**("identifier"[, "file"][, merge])
- You can use the EXIT function to leave the newly configured user interfaces and return to the standard application.
- You can use the EXITLS function to exit the current user interface and load a defined softkey menu.
- The EXE function can be used on HMI Advanced to call a program created as an application with the HMI Advanced OEM package or to start the Free Contour Programming application.  
With HMI Embedded sl, EXE can only be used to start the Free Contour Programming application.

For additional information, see "Methods"

## 3.7 Error handling, log book

The log book is a file (Error.com) to which error messages generated by syntax interpretation are written.  
The operator himself must set up the file in the comment directory (HMI Advanced).

```
DEF VAR1 = (R)
DEF VAR2 = (R)
LOAD
VAR1 = VAR2 + 1 ;Error message in log book because
;VAR2 has no value
```

The system does not start to interpret syntax until the start softkey has been defined and a screen form with start and end identifiers and a definition line has been configured.

```
//S(Start)
HS6=("1st screen form")
PRESS(HS6)
  LM("Form1")
END_PRESS
//END
//M(Form1)
DEF Var1=(R)
//END
```

If **Expand user interface** detects errors when interpreting the screen form configuration files, these errors are stored in the ERROR.COM ASCII file.

The file is stored as follows:

- for HMI Embedded sl: in the /tmp/hmiemb directory on the CompactFlash card
- for HMI Advanced: in the \DH\COM.DIR\ directory for HMI Advanced

This file is deleted each time HMI Embedded sl / HMI Advanced is restarted. It details

- which action was affected by an error,
- the line and column number of the first defective character and
- the whole defective line in the configuration file.

If the screen form was created using the PC test environment, then the error file will be stored in the directory referenced by environment variable RAMDISK (HMI Embedded sl).

The ERROR.COM file is created only when errors actually occur during interpretation of the configuration files.

### Displaying ERROR.COM

HMI Advanced:

Call the editor in the "Services" or "Start-Up" area.

HMI Embedded sl

"Setup" operating area, (in "Manufacturer" access mode), / "HMI" / "Editor" / "Temp drive" (4th softkey on vertical softkey bar. This softkey is not displayed with the manufacturer password, but it responds nonetheless (it is also displayed with the system password)

- Select ERROR.COM
- Press INPUT key

With the "File functions" softkey, the file can also be copied to a PC by means of a set up Windows network drive.



# 4

## 4 Programming Support

4.1 What does the programming support do? .....	BE1/4-106
4.2 Creating new screen forms .....	BE1/4-106
4.3 Special functions .....	BE1/4-107
4.4 General cycle support .....	BE1/4-109
4.5 Activating a dialog screen from the NC program .....	BE1/4-111
4.5.1 Structure of instructions .....	BE1/4-112
4.5.2 Instruction examples .....	BE1/4-113
4.5.3 Example 1 MMC command without acknowledgment .....	BE1/4-116
4.5.4 Example 2 Hold time and optional text variables .....	BE1/4-117
4.5.5 Example 3 MMC command with synchronous acknowledgment mode .....	BE1/4-118
4.5.6 Example 4 Positioning of input/output field .....	BE1/4-119
4.5.7 Example 5 Displaying graphics in the dialog screen .....	BE1/4-121
4.5.8 Example 7 Displaying OPI variables .....	BE1/4-122
4.5.9 Example 8 Asynchronous acknowledgment mode with softkeys .....	BE1/4-123

## 4.1 What does the programming support do?

The purpose of the programming support system is to assist programmers in writing programs using the ASCII Editor.

The system provides the following standard tools for this purpose:

- Cycle support
- Free contour programming
- Blueprint programming
- Recompilation
- Simulation

**References:**        /BA/, Operator's Guide

Customized user interfaces can also be created in the Programming support system. These interfaces are configured using the ASCII Editor and the tools provided by the **Expand user interface** system.

---

### Note

For the sake of compatibility, the cycles support (//C...) function is still supported by the description language used in earlier versions and not by the syntax of the **Expand user interface** system.

---

## 4.2 Creating new screen forms

New user interfaces are created with tools provided by the **Expand user interface** system. However, the procedures applied under Programming support differ in a number of ways. These differences are described in this chapter.

### Configuration file

The definition of new screen forms for programming support is stored in configuration file AEDITOR.COM.

Newly configured screen forms can be displayed in the Editor main screen through selection of 5 start softkeys (horizontal softkeys 2, 3, 4, 5 and 6).

The default labels for softkeys 2 to 5 are "Contour", "Drilling", "Milling" and "Turning".

Horizontal softkeys 14 and 15 (softkeys 6 and 7 in the expanded menu) are assigned the "Gauge turning" and "Gauge milling" functions.



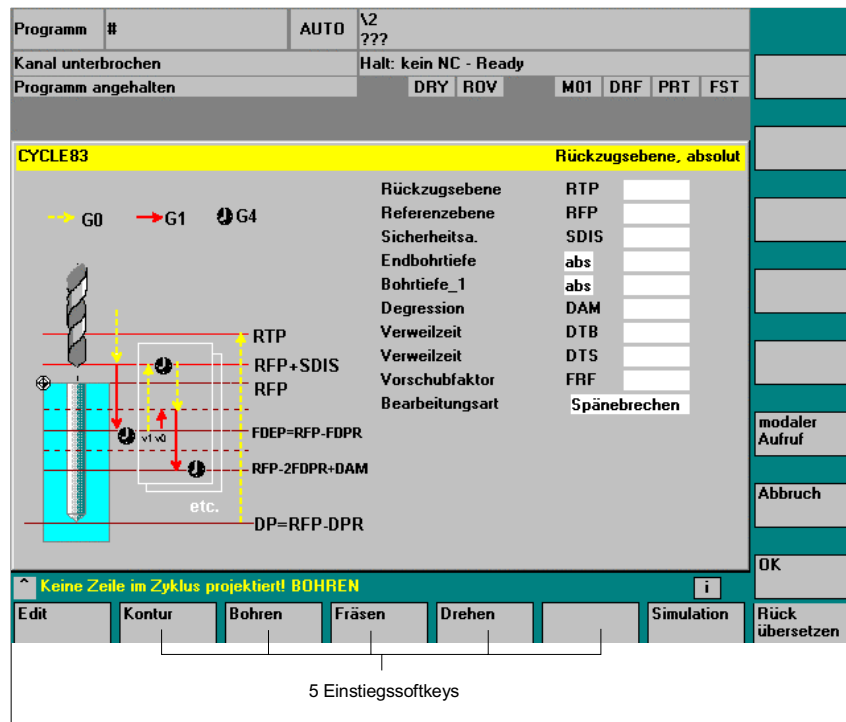


Fig. 4-1

### 4.3 Special functions

In addition to the functions provided by **Expand user interface**, the following extra functions are available under Programming support:

#### Find text

The "Find text" function is used to search for a string from the current cursor position in the NC program currently selected in the Editor and to output its value. Search backwards from the current cursor position:

**SB** ("String")

Search forwards from the current cursor position:

**SF** ("String")

with:

String      Text to be found

Rules governing text search:

- A blank must be inserted before and after the search concept unit, consisting of search string and its value, in the currently selected NC program.
- The system does not search for concepts within comment text or other strings.
- The value to be output must be a numerical expression. Expressions in the form of "X1=4+5" are not recognized.

- The system recognizes hexadecimal constants in the form of X1='HFFFF', binary constants in the form of X1='B10010' and exponential components in the form of X1='-.5EX-4'.

---

**Note**

The value of a string can be output if it contains the following between string and value:

- Nothing
  - Space character
  - Equality sign
- 

```
X100 Y200
Abc=SB("Y")           ;The variable Abc contains the value 200
or
X100 Y 200
Abc=SB("Y")           ;The variable Abc contains the value 200
or
X100 Y=200
Abc=SB("Y")           ;The variable Abc contains the value 200
```

## Recompile

In the Programming support system, it is possible to recompile NC code that has been generated with the GC function and to display the variable values in the I/O field of the associated screen form again.

Variables from the NC code are transferred to the screen form. At the same time, the variable values from the NC code are compared with the calculated variable values from the configuration file. If the values do not coincide, an error message is written to the log book because values have been changed during NC code generation.

If the NC code contains the same variable several times, it is evaluated at the point where it last occurs during recompilation. A warning is also written to the log book.

Variables not utilized in NC code during code generation are stored as user comment. The term "user comment" refers to all information required to recompile codes. User comment must not be altered.

---

**Note**

The block consisting of NC code and user comment can be recompiled only if it starts at the beginning of a line.

---

The configuration file contains the following NC code:

```
DEF VAR1=(I//101)
OUTPUT(CODE1)
  "X" VAR1 " Y200"
  "X" VAR1 " Y0"
END_OUTPUT
```

The following code is then stored in the parts program:

```
;NCG#TestGC#\cus.dir\aeditor.com#CODE1#
;#1#3#
X101 Y200
X101 Y0
;#END#
```

The Editor reads the following during recompilation:

```
X101 Y200
X222 Y0 ;The value for X has been changed in the parts program (X101 → X222)
```

The following value is displayed for VAR1 in the input screen form:

```
VAR1 = 222
```

## 4.4 General cycle support

### Introduction

When using cycle support in the parts programs, additional comment lines are generated in front of and behind the cycle call and used for recompilation. These lines start with ;# ... .

When cycles noted directly using the editor are called (also referred to as "old" cycles), then this information is missing. A description is provided in the following text as to how the information for cycles, contained in the lines in ;# can be provided by configuring INI files. This means, that after re-compilation, it will be possible to work with the following using the screen form support:

- Directly noted cycle calls
- Siemens cycle calls
- User cycle calls.

For Siemens cycles, the required configuration files are included in the scope of supply.

For screen forms generated with the **Expand user interface** function for the purpose of supporting user cycles, the information in the screen form descriptive files can be used for this process.

Cycle-specific settings can be made in the configuration to define whether cycles with ;# lines are to be processed in the parts program or recompiled via the configuration files.

This process can be used to optionally generate the addition of ;# lines required for screen form support in the parts program for directly noted "old" cycles.

**Prerequisites:**

For these cycles, a configuration generated using the **Expand user interface** tools must exist or be generated.

## Project design

The **wizard.ini** file can be stored in the following directories:

..\user  
..\oem  
..\hmi\_adv  
..\mmc2

A search is made in the specified directories in this sequence and the information of the section

**[MMC\_CycleWizard]** is then added so that it is up to date.

Entries that refer to cycle packets are located in this section. These include:

INI\_1=bohren.ini  
INI\_2=drehen.ini  
...  
INI\_n=paket.ini

Cycle-specific entries with the following form are located in the particular *paket.ini* files

<b>[cycleName]</b>	Name of the cycle' as section entry
Mname=	mandatory, screen form name
Dir=	mandatory, directory of the file with the screen form information
Dname=	mandatory, file with screen form information
Output	mandatory, name of the OUTPUT block
Anzp=	mandatory, number of parameters (defined variables)
Version=	optional, cycle version, without data 0
Code_typ=	optional, output with = 0 or without = 1 ;# lines

## Example

```
[CYCLE83]
Mname=CYC83
Dir=cst.dir
Output=bohren.ini
Anzp=17
Version=3
Code_typ=0
```

## Details

A differentiation is made between various versions of cycles with the same name using the version number.

Main block/secondary block and block numbers in front of the cycle call are kept unchanged.

Cycle call line in the parts program:

```
/1234 :44 CYCLE94( , )
```

After expansion by the string and ;# lines:

```
;NCG#CYC94#\CST.DIR\DREHEN.COM#NC1#1#*NCG;*RO*;*HD*
;#####*NCG;*RO*;*HD*
/1234 :44 CYCLE94(,,",)
;#END#*NCG;*RO*;*HD*
```

## 4.5 Activating a dialog screen from the NC program

### Introduction

The "Activate dialog screen from NC Program" function is also referred to as a "command channel".

With HMI Advanced and HMI Embedded sl, dialog screens defined by the user can be displayed:

- The appearance of the dialog screens is defined through configuration (modification of COM file in the cycles directory). The dialog screen is called and exited by function calls from the parts program.
- User-configured dialog screens do not modify the HMI system software (user interface).
- User-defined dialog screens cannot be called simultaneously in different channels.

### Application

One possible application of user-defined dialog screens is, for example, to assign defined values to particular user variables (GUD) prior to a parts program run.

- **Up to 2 channels:**  
The "command channel" is activated by default for channels 1 and 2.
- **More than 2 channels:**  
For HMI Advanced/PCU50 the "command channel" must be activated (if the SIEMENS measuring cycles have not yet been installed).  
This requires a modification to be made to the file F:\MMC2\COMIC.NSK in the Setup operating area:
  - Select file **F:\MMC2\COMIC.NSK** in the "Setup" operating area with the softkeys "MMC" → "Editor".
  - Insert the following text (after channel 1 and 2):  

```
REM CHANNEL
TOPIC(machineswitch)
COMIC_START(COMIC001MachineSwitch"...)
```

 [cf., text for 1st and 2nd channel]

When the control is restarted (OFF/ON), the command channel is activated for the corresponding channel.

## 4.5.1 Structure of instructions

### Syntax

**MMC** ("operating area, command, COM file, dialog screen name, user data definition file, graphics file (HMI Advanced only), display time or acknowledgment variable (HMI Advanced only), text variables, etc.", "Acknowledgment mode")

### Parameters:

- **Operating area**  
Name of softkey, with which the configured user dialogs can be called. Default entry: CYCLES, displayed as "Cycles" on softkey 14 and accessible via the ETC key.
- **Command**  
Picture\_on        select screen  
Picture\_off      deselect screen
- **COM file**  
Name of the dialog screen file (max. 8 characters, in the user, manufacturer, or standard cycles directory CST.DIR). The dialog display appearance is defined here. The dialog screen is used to display user variables and/or comment texts.
- **Dialog screen name**  
The individual displays are selected via the names of the dialog displays.
- **User data definition file**  
GUD file accessed on reading/writing of variables.
- **Graphics file (HMI Advanced only)**  
File name of a BMP graphic to be inserted.

\*.bmp files are stored in resolution-specific subdirectories; these include the following:

For standard cycles:

\CST.DIR\HLP.DIR\640.DIR	For 640 dpi resolution
\CST.DIR\HLP.DIR\800.DIR	For 800 dpi resolution
\CST.DIR\HLP.DIR\1024.DIR	For 1024 dpi resolution

For user cycles:

\CUS.DIR\HLP.DIR\640.DIR	For 640 dpi resolution
\CUS.DIR\HLP.DIR\800.DIR	For 800 dpi resolution
\CUS.DIR\HLP.DIR\1024.DIR	For 1024 dpi resolution

For manufacturer cycles:

\CMA.DIR\HLP.DIR\640.DIR	For 640 dpi resolution
\CMA.DIR\HLP.DIR\800.DIR	For 800 dpi resolution
\CMA.DIR\HLP.DIR\1024.DIR	For 1024 dpi resolution

### Display time or acknowledgement variable (HMI Advanced only)

Display time	in acknowledgment mode "N"
Acknowledgment variable	in acknowledgment mode "A"

### Text variables

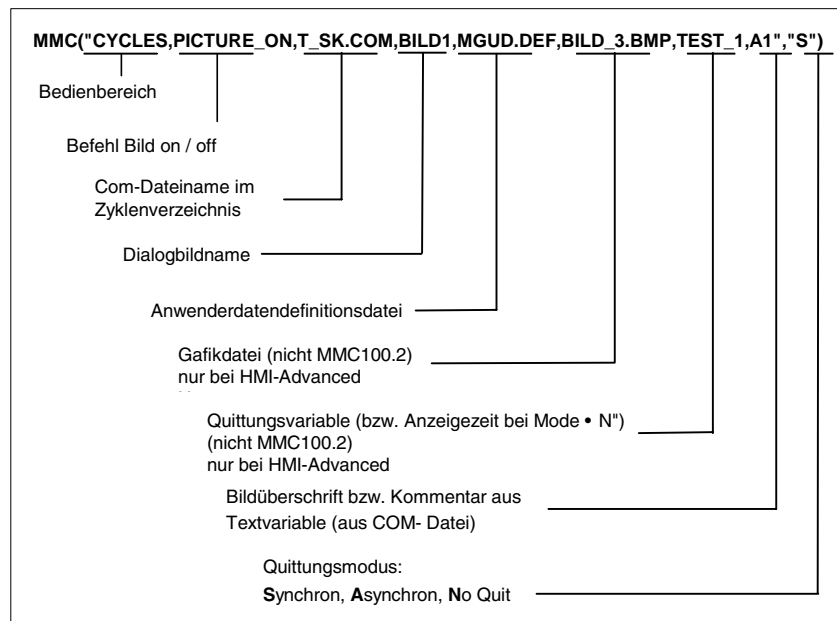
Screen header or comment text from a text variable in the COM file.

### Acknowledgment mode

"S" Synchronous, acknowledgment via "OK" softkey

## 4.5.2 Instruction examples

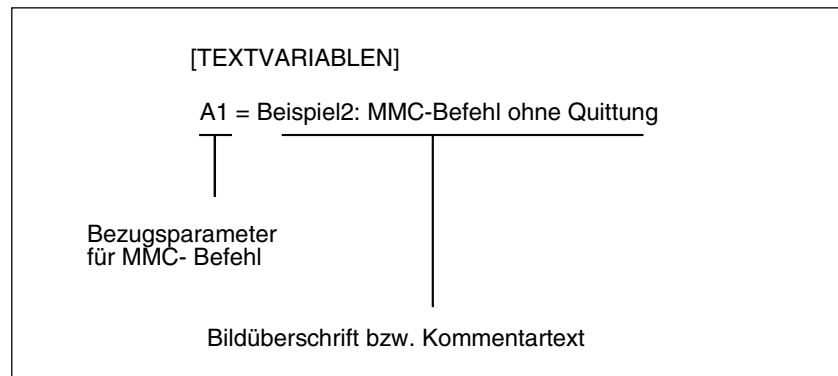
### MMC command in the parts program







## Text variables



## Configuring softkeys for screen call

Softkey assignment for MMC command with asynchronous acknowledgment mode.

### **[SCREEN3]**

SK1 = END

SK2 = Screen2

Softkeys SK0 to SK15 can be configured

### 4.5.3 Example 1 MMC command without acknowledgment

#### Parts programs

```
N10   MMC( "CYCLES, PICTURE_ON, T_SK.COM, SCREEN1, GUD4.DEF, , ,  
        A1", "N" )  
N20  TEST_1 = 1  
N25  G4 F10  
N30  MMC( "CYCLES, PICTURE_OFF", "N" )  
M30
```

#### Dialog screen file (\*.COM)

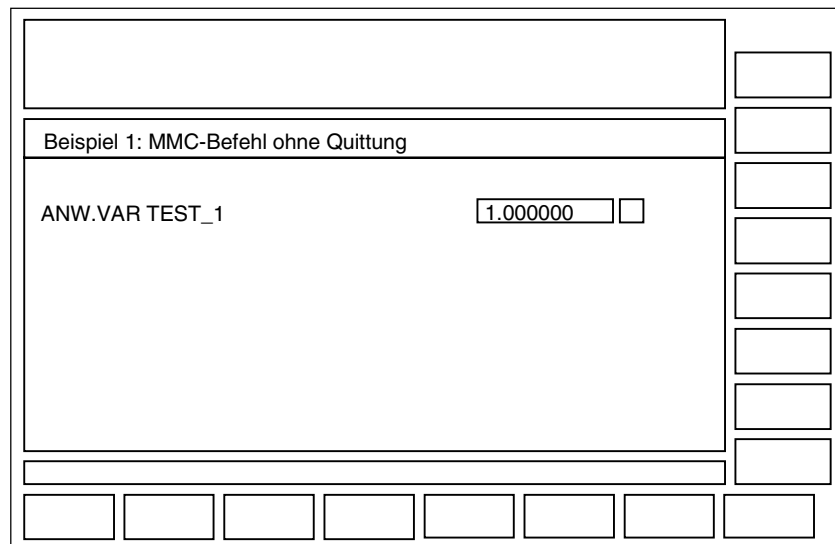
```
//C1(SCREEN1)  
(R//USER VAR TEST_1/W/TEST_1//)
```

#### Text variable

```
[TEXT VARIABLES]  
A1 =.....Example1: MMC command without acknowledgment.....
```

#### Sequence

The user variable TEST\_1 from GUD4.DEF with header A1 is displayed for a short time.  
The hold time comes from parts program block N25.



## 4.5.4 Example 2 Hold time and optional text variables

### Parts programs

```

N10 MMC( "CYCLES, PICTURE_ON, T_SK.COM, BILD6, GUD4.DEF, , 10,
        T1, G1" , "N" )
N15 G4 F15
N30 MMC( "CYCLES, PICTURE_OFF" , "N" )
M30

```

### Dialog screen file (\*.COM)

```

//C6(screen6)
(R///USER VAR TEST_1,%1/W/TEST_1///)

```

### Text variable

[TEXT VARIABLES]

T1 = Example2: Hold time and optional text variables ...  
G1 = Optional text variable

The 7th parameter is interpreted as the display time for the mode without acknowledgment (10 seconds). The contents of the table are then deleted. The screen form remains on the screen until PICTURE\_OFF is selected. The 8th parameter (T1) is the text variable for the screen header. If there is no entry, the operating area name (cycles) is displayed. The parameters 9 to 23 are optional text variables ("G1=optional text variable"). In this COM file, the optional text variables must be preassigned in the "[Text variables]" section.

Beispiel 2: Verweildauer und optionale Textvariable	
ANW.VAR TEST_1, optionale Textvariable	1.000000 <input type="checkbox"/>

### Sequence

The comment text from the COM file (USER VAR TEST\_1) is expanded in this example at the position of the first wild card (%1) to include the contents of the text variable "G1=optional text variable". By calling the text variables contained in the MMC command (9th to 23rd parameters), messages or names can be "composed" in this way.

### 4.5.5 Example 3 MMC command with synchronous acknowledgment mode

#### Parts programs

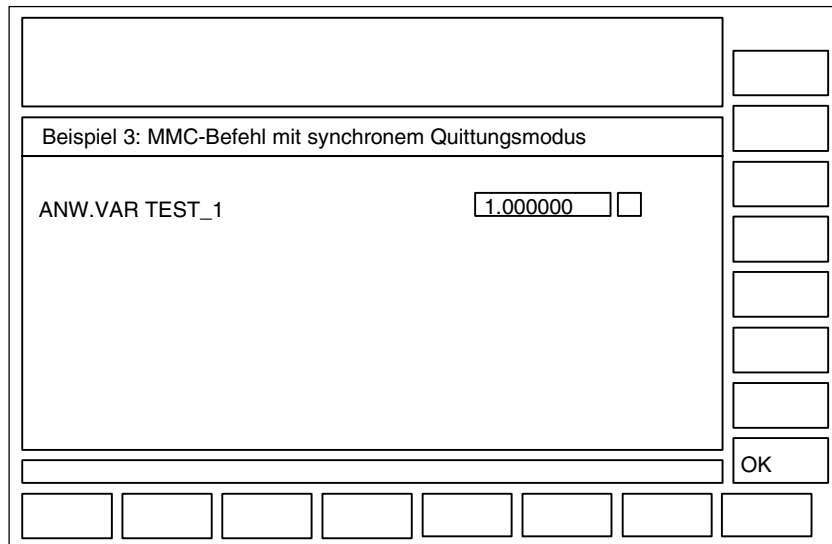
```
N15 MMC( "CYCLES, PICTURE_ON, T_SK.COM, SCREEN1, GUD4.DEF, , , F1" ,  
        "S" )  
N18 STOPRE  
N20 TEST_1 = 5  
N25 MMC(CYCLES, PICTURE_OFF", "N" )  
M30
```

#### Dialog screen file (\*.COM)

```
//C1(SCREEN1)  
(R///USER VAR TEST_1/W/TEST_1///)
```

#### Text variable

F1 = ...Example3: MMC command with synchronous acknowledgment mode...



#### Sequence

The user variable Test\_1 is displayed until the "OK" softkey is pressed.

The user variable has also been overwritten with the value 5 here.

Without STOPRE, this assignment would be carried out **before** the keyboard input (after it with STOPRE).

## 4.5.6 Example 4 Positioning of input/output field

By specifying position parameters in the COM file, you can insert the comment field, or input and output field, at any position within the display area.

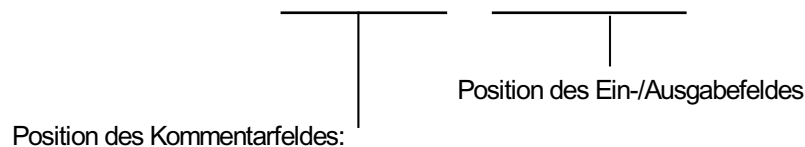
### Parts programs

```
N15 MMC("CYCLES,PICTURE_ON, T_SK.COM; Screen2,
        GUD4.DEF,,,C1","S")
N20 TEST_3 = 5
N30 MMC("CYCLES,PICTURE_OFF","N")
N40 M30
```

### Dialog screen file (\*.COM)

```
//C2(Screen12)
```

```
(R//Var.Name/R/TEST_3/6000,2800,8000/200,3000,7500)
```

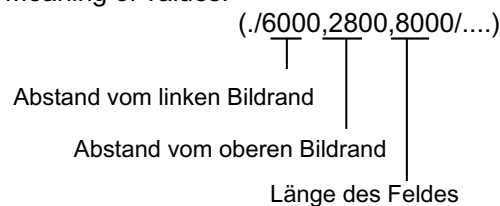


Position des Kommentarfeldes:

Erster Wert = 0 --> automatische Positionierung auf Standardwert  
Keine Angaben --> Standardpositionierung wie bei PCU20

The two parameters each consist of three numerical values; these specify the position and length of the field. The values are specified in Twips, with 15 Twips roughly corresponding to one pixel. The field height is defined as being 250 Twips.

Meaning of values:



It is possible to graphically configure 16 comment and 16 input or output fields. Where there are more than 16 fields, they can be controlled via a scroll bar.

The screenshot shows a graphical user interface with a title bar at the top. Below the title bar is a header area containing the text "Beispiel 4: Positionsparameter". The main content area features a single input field with the value "1.000000" and a label "ANW.VAR TEST\_2" positioned to its right. To the right of the input field is a small square checkbox. Below the input field is a horizontal bar, and below that is an "OK" button. At the bottom of the window, there is a row of eight small, empty square buttons.

To ensure that the cursor control works seamlessly, the configured fields must overlap one another (see below):

The screenshot shows a graphical user interface similar to the one above. It has a title bar and a header area with the text "Beispiel 4: Positionsparameter". The main content area contains four input fields, each with the value "1.000000" and a small square checkbox to its right. The input fields are arranged vertically and overlap each other. Below the input fields is a horizontal bar, and below that is an "OK" button. At the bottom of the window, there is a row of eight small, empty square buttons.

### 4.5.7 Example 5 Displaying graphics in the dialog screen

Graphics that were created with Paintbrush, for example, can be displayed in the dialog screen by means of a graphics file specification.

A comment text for the picture can also be preassigned using the COM file. You can position this comment text by specifying position parameters.

---

#### Note

You can only move the picture itself by repositioning it in the graphics program.

---

#### Parts programs

```
N10 MMC( "CYCLES, PICTURE_ON, T_SK.COM, SCREEN8, GUD4.DEF, GRA.BMP, ,
        M1", "S" )
N20 MMC( "CYCLES, PICTURE_OFF", "N" )
N30 M30
```

#### Dialog screen file (\*.COM)

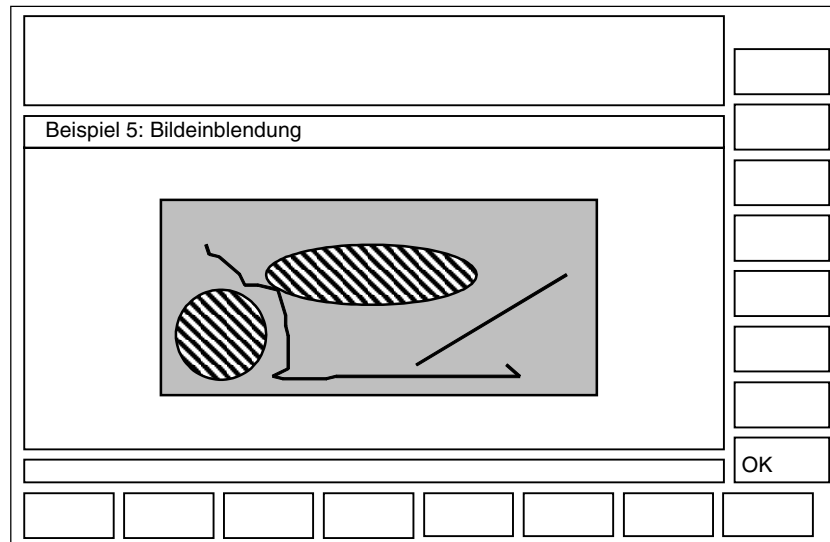
```
//C8(SCREEN8)
(I///Insert picture no. 2///4000,3000,7500)
(I///Created with Paintbrush ///4000,3250,7500)
```

#### Text variable

M1 = .....Example5: Insert picture.....

Pictures are created, e.g., with the drawing tool "Paintbrush".

Picture size: 300X500 pixels, you can only change the picture size in Paintbrush.



### 4.5.8 Example 7 Displaying OPI variables

OPI variables can be displayed using the COM file.

#### Parts programs

```
MMC( "CYCLES, PICTURE_ON, T_SK.COM, SCREEN7, GUD4.DEF, , TEST_1, J1",  
"S" )
```

#### Dialog screen file (\*.COM)

```
//C7(Screen7)  
(R//Test_1/R/Test_1)  
(I//); (Interpreted as space line)  
(R//Actual value from axis 1/R/$actual value)  
(R//1/R parameter 12/W/$R[12])
```

#### Text variable

```
J1 = ...Example7: OPI variables  
[OPIVar]  
$Actual value=/Channel/machineaxis/actToolbasePos[u1,1]  
$R[12]=/Channel/Parameter/rpa[u1,12]
```

Variable	Value	Checkbox
ANW.VAR TEST_1	1.000000	<input type="checkbox"/>
Istwert von Achse1	20.00000	
R-Parameter12	1.000000	<input type="checkbox"/>

#### Sequence

A space line is created under the variable TEST\_1.  
The actual axis value is read-only.  
R12 is preset with 1.



### 4.5.9 Example 8 Asynchronous acknowledgment mode with softkeys

Softkeys can be preprogrammed in the COM file for display in asynchronous mode, linked with the acknowledgment variable and evaluated in the parts program.

#### Parts programs

```

N10   QUIT_1 = "START"
N20   MMC("CYCLES, PICTURE_ON,T_SK.COM,SCREEN3,GUD4.DEF,,
        QUIT_1,K1","A")
N30   LABEL0:
N40   STOPRE
N50   IF MATCH (QUIT_1,"SK1") >= 0 GOTOF LABEL1
N60   IF MATCH (QUIT_1,"SK2") >= 0 GOTOF LABEL2
N70   GOTOB LABEL0
N80   LABEL2:
N90   MMC("CYCLES, PICTURE_ON,T_SK.COM,SCREEN1,GUD4.DEF,,,
        N1","N")
N100  G4F10
N110  LABEL1:
N120  MMC("CYCLES, PICTURE_OFF","N")
N130  M30

```

#### Dialog screen file (\*.COM)

```

//C3(Screen3)
(S///USER VAR QUIT_1/W/QUIT_1//)
[TEXT VARIABLES]
K1 = ..Example8: MMC command with asynchronous acknowledgment mode
N1 = ..Example8: Screen2

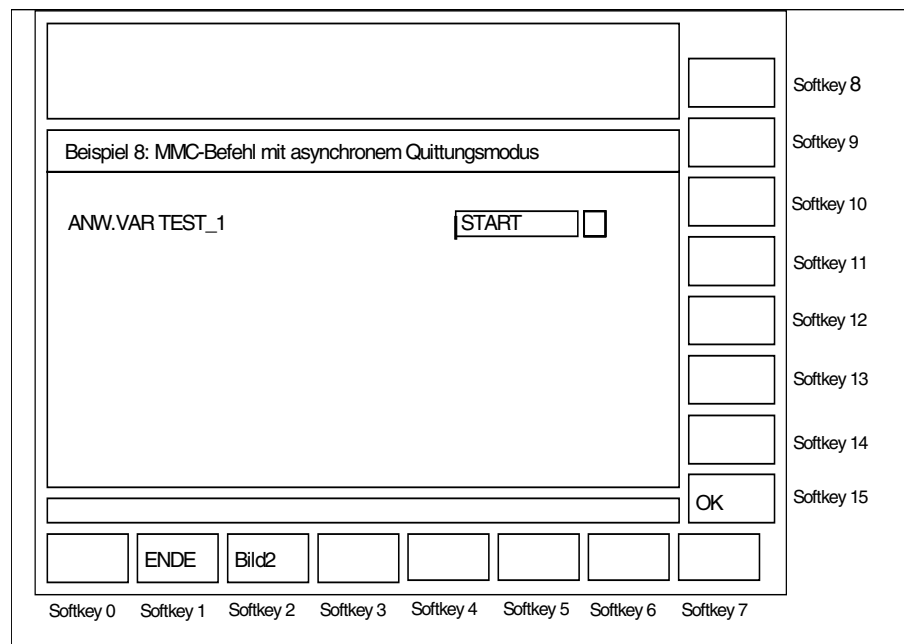
```

#### Text variable

```

[SCREEN3]
SK1 = END
SK2 = Screen2

```



## Sequence

The screen called via the asynchronous MMC command continues to be displayed until one of the two configured softkeys is actuated:

- With the "END" softkey, the user dialog is immediately exited.
- With the "Screen2" softkey, a further screen is then displayed for 10s.

## Program structure

- The acknowledgment variable is defined as a string.  
Length of the string:  $\geq 20$   
(Values  $< 20$  are evaluated internally only,  
**SK0 ... SK15** are entered at position 17...20 when a softkey is actuated).
- The string is assigned a value in the parts program and any old softkey information is deleted.
- Before the parts program can branch as a function of the acknowledgment variable, the block search must be halted by the STOPRE command.
- `IF Match (Quit_1, " SK1")  $\geq 0$  GotoF Label1`  
searches for a string within the string  
If no softkey is actuated, the loop runs again.

The image shows a graphical user interface (GUI) dialog box. At the top, there is a title bar with the text "Beispiel 8: Bild 2". Below the title bar, the main area contains the text "ANW.VAR TEST\_1" followed by a numeric input field containing "5.00000" and a small square checkbox. To the right of the main area, there is a vertical column of seven rectangular buttons. The bottom-most button in this column is labeled "OK". At the bottom of the dialog box, there is a horizontal row of eight rectangular buttons.



## Notes

# 5

## 5 Configuring OP Hotkeys and PLC Keys

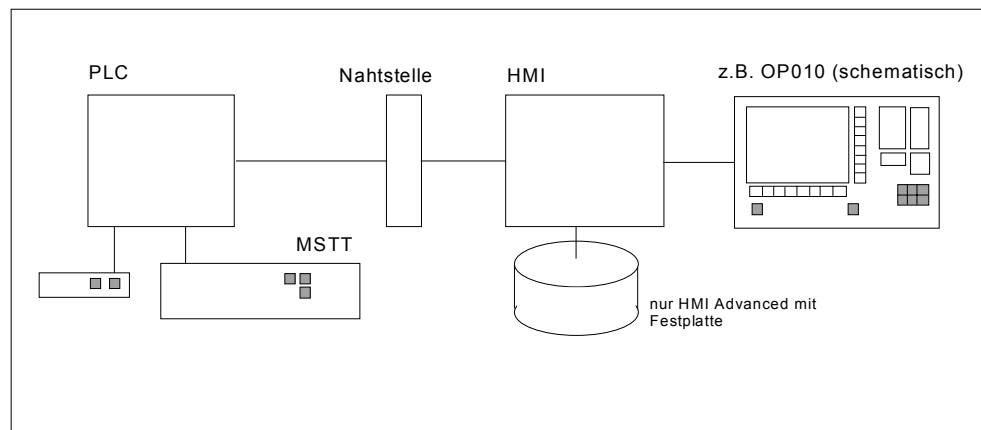
5.1 Introduction .....	BE1/5-128
5.1.1 OP hotkeys .....	BE1/5-129
5.1.2 Assigning functions to keys .....	BE1/5-130
5.1.3 Configuring the M key and Menu Select as HK7, HK8.....	BE1/5-134
5.1.4 Expansions and special cases.....	BE1/5-135
5.2 Selecting displays created with 'Expand user interface' .....	BE1/5-136
5.2.1 Assigning INI files to operating areas .....	BE1/5-136
5.2.2 Configuring 'Expand user interface' functions .....	BE1/5-137
5.3 PLC interface .....	BE1/5-138
5.3.1 Description of the interface.....	BE1/5-138
5.3.2 Requirements for the PLC program.....	BE1/5-138
5.3.3 Screen selection from PLC .....	BE1/5-139
5.4 Expanded functions for PLC keys.....	BE1/5-142
5.5 List of selectable states in HMI Advanced.....	BE1/5-143
5.6 List of selectable states in HMI Embedded sl.....	BE1/5-144
5.7 List of selectable states in ShopMill on NCU.....	BE1/5-145
5.8 List of selectable states in ShopTurn on NCU.....	BE1/5-147
5.9 Hotkey assignment on the MFII keyboard .....	BE1/5-150
5.10 Examples .....	BE1/5-151
5.11 Configuration overview .....	BE1/5-151
5.12 Terms.....	BE1/5-153

## 5.1 Introduction

This chapter deals with configuring

- A group of 6 hotkeys on the OP 010 and OP 010C and Sinumerik keyboards with hotkey blocks
- The <machine> and <Menu Select> keys. The assignment of these keys can be changed optionally.
- Keys (e.g., machine console keys) and events evaluated by the PLC, which, as PLC keys or "virtual keys", can initiate configured sequences of operations in the HMI system.

### Overview



The 6 OP hotkeys can be used to display any HMI operating area directly. This shortens the usual selection process via the main menu. This changes the default assignment of the 6 hotkeys.

The interface between the PLC and the HMI enables a PLC key to be transferred from the PLC to the HMI. The operations initiated on the HMI system in this way can be configured. Key numbers 50 to 254 are available for use by the PLC.

As an option, the <Machine> and <Menu Select> keys can be configured in the same way as the 6 OP hotkeys and are assigned to HK7 and HK8.

Configuration is carried out using the **Expand user interface** system functions described in the previous chapters.

### Applications

- Selection of operating areas (e.g., Machine, Parameters, etc.)
- Selection of specific submenus (e.g., selection of the alarm screen in the Diagnosis area)
- Initiation of actions (e.g., selection of the tool list in the Parameters area and pressing of softkey HS3)
- Specific selection of menus created using the **Expand user interface** system
- Selection of additional screens, depending on the current operating situation, in screens created using the **Expand user interface** system.

## 5.1.1 OP hotkeys

### Standard layout

The 6 OP hotkeys are laid out in 2 lines, each of which contains 3 keys:

Line 1

<b>Label (no symbol)</b>	Depends on OP		Program	Offset
	OP10	Machine		
	OP10C	Machine		
	OP10S	Position		
<b>Configured as</b>	HK1	HK2	HK3	

Line 2

<b>Label (no symbol)</b>	Prg. Manager	Alarm	Custom
<b>Configured as</b>	HK4	HK5	HK6

### Optional HK7 and HK8

The OP keys <Machine> and <Menu Select> can be configured in the same way as HK1 to HK6. This makes it possible to disable the default functions of these keys and to activate new, user-defined functions.

<b>Label (no symbol)</b>	Machine	Menu Select	
<b>Configured as</b>	HK7	HK8	

You can find more information about HK7 and HK8 in chapter “Configuring <M> key and <Menu select> as HK7, HK8”.

---

#### Note

Hot keys 1 and 7 (<M> key) cannot be distinguished by the hardware on the OP10S (position of labeling). Hot key 7 is always triggered when one of the keys is pressed. If HK1 is configured, this event can only be triggered by an external (MFII) keyboard.

---

## 5.1.2 Assigning functions to keys

When the system is supplied, the assignment of hotkeys to the functions is entered in file **keys.ini**.

### Directories

With HMI Advanced, the file can exist in various directories. These are as follows:

- user
- oem
- add\_on \*)
- mmc0w32 \*)
- mmc2 \*)
- hmi\_adv \*)

\*) These directories are reserved for Siemens.

The files are executed in the above sequence. Entries in the directory, which appears at the beginning of the sequence will disable existing entries in directories, which appear later in the sequence.

The assignments for HK1 to HK6 are supplied and stored as follows in keys.ini in directory **mmc2** for the standard system:

Key	Function
HK1 Position	Machine operating area, last screen
HK2 Program	Program operating area, last screen
HK3 Offset	Parameters operating area, last screen
HK4 Prg. manager	Main screen programming
HK5 Alarm	Diagnosis operating area, alarm screen
HK6 Custom	Default Custom operating area, last screen (user-configured operator interface)

The assignment for **ShopMill/ShopTurn** applications is as follows:

Key	Function
HK1 Position	Machine operating area, last screen
HK2 Program	Program editor, last status
HK3 Offset	Offset area, last status
HK4 Prg. manager	Program directory, last status
HK5 Alarm	Diagnosis operating area, alarm screen
HK6 Custom	Default Custom operating area, last screen (user-configured operator interface)

With HMI Embedded sl, the data in keys.ini can be adapted using the Software Configuration Kit (SCK):



## Entry format

Details about the input format can be found in the "Hotkey event" subsection below. To better understand the repeated actuation of the same hotkey used in that subsection, multiple keypress will be described first.

## Multiple keypress

A function extension determines the sequence when the hotkey is pressed repeatedly:

1. The number "n" in the key data in file Keys.ini represents the number of keypress repeats, thus allowing a separate task/state/action combination to be assigned each time the hotkey is pressed. This means that each time the key is pressed, a change of state can take place, and a screen and a softkey can be selected.
2. When the user changes to another operating area, the repeat status of the hotkey is cleared. In the new area, the hotkey is considered not to have been pressed.  
It does not matter how the user changed to the other operating area (operating area menu, etc.). In addition, each time a function key is pressed (e.g., softkey, area switchover key, channel switchover key, etc.), the status is reset.
3. Inputting data into fields in the current screen form does not interrupt the repetition sequence. Pressing a softkey in the current screen form interrupts the repetition sequence with the softkey that was used to call the current screen form.

---

### Note

Multiple keypress is possible for HK1–HK8 and for keys HK9–HK49, which are reserved for Siemens.

---

## Configuration file keys.ini

Every version of the configuration file **keys.ini** in the above mentioned directories drives the the hotkey responses in a section **[KeyConfiguration]**. Every entry (line) defines the response of the system to a specific keypress (in terms of multiple keypresses) of a specific hotkey. Hereinafter, this is called a hotkey event.

The **keys.ini** configuration file has a dedicated section for assigning the ini files for PLC keys. Without this entry, the ini files will not be detected. The default settings are as follows:

### [HMI\_INI\_FILES]

Task0 = machine.ini

Task1 = paramet.ini

Task2 = program.ini

Task3 = services.ini

Task4 = diagnosis.ini

Task5 = startup.ini

Task6 = shopmill.ini **alternative** Task6 = shopturn.ini depending on the system,  
mandatory!

Task11=custom.ini

## Hotkey event

Structure:

**KEY***x.n* = **Task**:=*task*, **State**:=*state*, **Action**:=*action*

The maximum number of attributes, which can be assigned to a hotkey event are the 4 attributes in bold print above.

The KEY key attribute identifies the event and must always come first. All other attributes are optional, but at least one other attribute must be specified. These attributes can appear in any sequence in the line.

- Attribute **Key** (hotkey)  
The value *x.n* contains the hotkey number *x* and a discrete event *n* corresponding to the *n*th activation of the key (when the key is pressed several times in succession).  
For example, Key 3.5 addresses the hotkey event for HK3, 6th activation.  
The range of values of the hotkey numbers is 1 – 254.  
The range of values for event *n* is 0 – 9 (10 keypresses).  
Hotkeys 1 – 8 are assigned to the keys of the OPs (see chapter “OP hotkeys”).  
Hotkeys 9 – 49 are reserved for Siemens. PLC-specific keys from 50 – 254 are also available. These keys are not actual hotkeys, but are used for selecting screen forms via the PLC (virtual keys). Keys 50 – 254 can also be configured without a task. In this case, the event is always assigned to the current task.
- Attribute **Task** (operating area)  
A hotkey is assigned to an operating area (task) via a task number. The task number also defines the **horizontal** softkey to which the operating area is assigned in the area switchover menu. This creates a permanent connection between the task number and the corresponding softkey. In the HMI system, there are three softkey menus (two in HMI Embedded sl) reserved for the area switchover function. The default assignments are as follows:

Pressing the area switchover key selects softkey menu 1:

SK1 → Task0

...

SK8 → Task7

Pressing the <ETC> key again accesses softkey menu 2

SK1 → Task8

...

SK8 → Task15

Pressing the <ETC> key again accesses softkey menu 3 (only possible in HMI Advanced):

SK1 → Task16

...

SK8 → Task23

Pressing the <ETC> key again returns to softkey menu 1.

In HMI Advanced, the range of values for the task numbers is between 0 and 23.

In HMI Embedded it is between 0 and 15.

If no task is specified, the event is assigned to the current task (current operating area).

The task numbers can be accessed:

- From the **Regie.ini** (Control.ini) file for HMI Advanced.
- With the **SCK** (Software Configuration Kit) tool in HMI Embedded sl.

An application-specific assignment between SKs and operating areas (tasks) can be preset during configuration:

HMI Advanced	Section [TaskConfiguration] in file control.ini
HMI Embedded sl	SCK "Modify configuration"

- **Attribute State:**  
The State attribute can be used to select a specific screen within a task. The range of values depends on the individual application and is limited to between 0 and 65534 (65535 is used by the system).

For lists of selectable states, refer to:

“List of selectable states in HMI Advanced”

“List of selectable states in HMI Embedded sl”

“List of selectable states in ShopMill on NCU”

“List of selectable states in ShopTurn on NCU”

- **Attribute Action:**  
The Action attribute can be used to select a specific softkey within a task. In order to be effective, the attribute must be set from a defined state, e.g., the main menu.  
The value range depends on the corresponding application, with a maximum of between 0 and 17. A special role is played by the Action attribute with a value  $\geq 100$ , which initiates a screen selection via **Expand user interface**.  
Actions in the value range 0 - 99 are not executed for ShopMill/ShopTurn.

---

#### Note

A power ON is required in order for the changes to keys.ini to take effect.

---

### Example of multiple hotkey keypress

The first time the hotkey is pressed, the associated area is activated and a state/action may be initiated in this area (entry 1). Pressing the hotkey again will execute the subsequent entries for this hotkey and an area switchover will not take place. Once all the entries configured for a hotkey have been executed, the cycle is repeated. The entries are always executed in ascending order (from 0-9). Execution in descending order is not possible. There must be no "event gaps" in the configuration (missing entries). A gap is treated like the end of the chain and processing will restart at Key x.0 the next time a key is pressed.

### Configuring example

```
[KeyConfiguration]
KEY1.0 = Task:=0, State:=10, Action:=2      ;Hotkey 1, 1st activation
KEY1.1 = State:=10, Action:=3              ;Hotkey 1, 2nd activation
KEY1.2 = Action:=4                          ;Hotkey 1, 3rd activation
```

### Assignment of the softkeys to the action values

Horizontal softkeys SKHi	Action
SKH1	0
SKH2	1
SKH3	2
SKH4	3
SKH5	4
SKH6	5
SKH7	6
SKH8	7

Vertical softkeys SKVi	Action
SKV1	8
SKV2	9
SKV3	10
SKV4	11
SKV5	12
SKV6	13
SKV7	14
SKV8	15

<RECALL> key	16
<ETC> key (HMI-Embedded sl only)	17

### 5.1.3 Configuring the M key and Menu Select as HK7, HK8

The <M> key (Machine) and <Menu Select> key can be optionally configured as hotkeys. Doing so causes these keys to lose their original definition and function. The new functionality is defined by the **Keys.ini** configuration file. If there is a key definition in the configuration file for the key, but no **response** was saved, then the key does not have a function.

Hotkey assignment:

	MFII keyboard	Hotkey
<M> key (Machine)	SHIFT+F10	HK7
<Menu Select> (area switchover)	F10	HK8

If there is no **entry** in the keys.ini configuration file for hotkey 7 or hotkey 8, the key will not be mapped as a hotkey but will retain its original function (compatible mode). A key definition without a response will be assigned the <empty> attribute.

## Examples

```
[KeyConfiguration]
; Activate mapping of the M key to hotkey 7 and set new
; key reaction
Key7.0 = Task:= 3, State := 10

; Activate mapping of the Menu Select key (F10) to
; hotkey 8,
; key has no function
Key8.0 = <empty> ; No reaction assigned
```

### 5.1.4 Expansions and special cases

#### HMI Advanced

The machine manufacturer can overload the entries in **keys.ini** in directory mmc2 with his own settings. These settings can be entered in the /user or /oem directories searched earlier during execution. Only deviations in the settings must be stored, not the complete block.

**Special cases Action in HMI Advanced:**

The ETC key, Action 17, has no effect.

#### HMI Embedded sl

**Special cases Task, State, and Action**

- The range of values for the **Task** is limited to between 0 and 15.
- HMI Embedded sl always selects the **last** screen combination that was active before the task change (state not configured).
- The range of values for **State** is limited to 0. **Only** the **main menu** of the operating area can be selected.
- The range of values for **Action** is between 0 and 17 :
  - SK horizontal 1 to 8,
  - SK vertical 1 to 8,
  - Recall and
  - ETC limited.

## 5.2 Selecting displays created with 'Expand user interface'

An Action value  $\geq 100$  automatically switches the HMI software to **Expand user interface** operational sequences.

If Action  $\geq 100$ , the configuration of the **INI files** determines which softkey menu or screen form should be displayed and in which state.

### 5.2.1 Assigning INI files to operating areas

#### HMI Advanced/HMI Embedded sl

The following INI files and sections can be configured for the jump entry to **Expand user interface**:

Operating area	File	Section
Machine	machine.ini	[KeyConfigurationAuto] [KeyConfigurationMDA] [KeyConfigurationJOG]
Parameters	paramet.ini	[KeyConfiguration]
Program	program.ini	[KeyConfiguration]
Services	services.ini	[KeyConfiguration]
Diagnostics	diagnosis.ini	[KeyConfiguration]
Startup	startup.ini	[KeyConfiguration]
Custom	custom.ini	[KeyConfiguration]

#### ShopMill on NCU/ ShopTurn on NCU

For ShopMill and ShopTurn on NCU, the INI file is called **shopmill.ini** and **shopturn.ini**, respectively. Both files contain the following sections in which **Expand user interface** functions can be configured.

```
[MachineManual]
...
[MachineAutomatic]
...
[ProgramManager]
...
[Program]
...
[MessagesAlarms]
...
[ToolsZeroOffset]
...
[MachineMDI]
```

## 5.2.2 Configuring 'Expand user interface' functions

One or more states (according to the State list) can be configured for each action ( $\geq 100$ ) and you can also specify which **Expand user interface** function is to be initiated. If nothing is configured in these INI files, no screen forms or softkey menus will appear.

### General notation:

**Action.State = Screen form/softkey menu to be loaded, configuration file**

Action = Action is Action  $\geq 100$  configured in KEYS.INI.  
State = Current state of the application.  
Configuration file = File in which the configuration is stored.  
LS/LM= **Expand user interface** - commands to load from softkeys/screen forms.

### Examples

#### [KeyConfiguration]

```
100.10=LS("Softkey1","param.com")
100.30=LM("Form1","param.com")
101.10=LS("Softkey1","param.com")
101.30=LM("Form2","param.com")
102=LM("Form2","param1.com")
```

For the first line, for example, this means:

If an Action 100 is initiated when in state (status/screen) 10, the softkey menu configured in file param.com will be displayed with the name softkey1.

For the last line, for example, this means:

If Action 102 is initiated, the screen form with the name screen form2 will be called in the **actual** screen. This screen form is configured in file param1.com.

## 5.3 PLC interface

### 5.3.1 Description of the interface

In the PLC interface, area DB19.Byte10 is provided for key selection. Here, the PLC can select **one** key between 50 and 254 directly.  
(Keys 1 to 49 are reserved for Siemens, keys 50 to 254 are specific PLC keys.)

---

#### Note

In M:N mode, the area for the 2nd HMI interface is DB19.Byte60.

---

### Acknowledgment

An HMI system acknowledgment is divided into two steps:

1. In the first step, the HMI software transfers the control information 255 to DB19.Byte10.
2. In the second step, the actual acknowledgment takes place on the PLC in which DB19.Byte10 is cleared.

This is necessary in order that, despite the lack of synchronism between the HMI and the PLC, the same key code transferred twice in immediate succession can be detected by the HMI. This defined virtual key activation enables the HMI to identify every PLC key sequence uniquely. The control information is of no significance to the PLC program (transparent) and must not be changed.

### Next key input

If the transfer byte is set to 0, the PLC program can specify a new key. The current key request is processed in parallel in the HMI system. The request leads to a switchover to the corresponding task or initiates a state/action command in the current task (configured in **keys.ini**).

If the task switchover cannot be executed, the operator interface will send a message to indicate this.

### 5.3.2 Requirements for the PLC programm

A new request can only be set if the HMI system has acknowledged the previous request (0 in the interface). If the PLC program is deriving the key from an MCP key or from another source, it must provide sufficient intermediate storage for the key to ensure that no data is lost if the key is pressed very quickly.



### 5.3.3 Screen selection from PLC

There is an interface between the PLC and HMI Embedded sl for selecting the displays. These PLC displays offer the same means of display and functions as for the displays selected via the softkeys.

#### Rough description of the interface

The interface contains:

- The display number
- Control bits from PLC to HMI
- Control bits from HMI to PLC

Because each NCU can be addressed by up to two HMI Embedded sl's at the same time in systems with "several operator panel fronts and NCUs", this interface can exist in duplicate.

**References:** /FB/, B3 "Several Operator Panels and NCUs"

The interface is located in the operator panel front interface (DB19). The interface requires a total of 8 bytes in DB19, 4 bytes for each HMI Embedded sl.

#### Location and structure of the interface

The interface between HMI Embedded sl and the PLC uses the following data:

HMI1:

DB19.DBW28: Display number  
DB19.DBB30: PLC control bits → HMI, PLC byte  
DB19.DBB31: Control bits HMI → PLC, HMI byte

HMI2:

DB19.DBW28: Display number  
DB19.DBB80: PLC control bits → HMI, PLC byte  
DB19.DBB81: Control bits HMI → PLC, HMI byte

PLC byte:

Bit0 Display selection  
Bit1 deselect screen

HMI byte:

Bit0 Display selection/deselection accepted  
Bit1 Display is selected/deselected  
Bit2 Display is selected  
Bit3 Display is deselected  
Bit4 Error, display selection not possible  
Bit7 Inactive bit

Two bytes are used for the **display number** to be transferred by the PLC, one byte respectively (PLC byte and HMI byte) for **coordinating** the display selection.

## Using the interface

The interface is operated by the **PLC OEM program** with the following functions:

Table 5-1: DB 19 extract, first HMI interface

	PLC → HMI	Selection	Deselect ion	HMI → PLC	Selec-tion	Deselec-tion
DBW 28	Screen number	(1)				
Bit	DBB 30			DBB 31		
0	Display selection	1 (2) 0 (4)		Select/deselect accepted	1 (3) 0 (6)	1 (2) 0 (3)
1	deselect screen		1 (1) 0 (4)	Display is being selected Display is deselected	0 (3) 1 (5)	0 (3)
2				Display is selected	0 (3) 1 (7)	0 (3)
3				Display is deselected	0 (3)	0 (2) 1 (3)
4				Error, display selection not possible	0	0 (2)
5				–		
6				–		
7				Inactive	0	

## Display selection

The relations with the individual steps are shown in the above table by the numbers in brackets (step numbers).

1. The PLC enters the display number in the number word.
2. The PLC sets bit0 in the PLC byte for display selection if bit0 and bit7 are set to zero in the HMI byte.
3. HMI Embedded sl acknowledges to the PLC that it has accepted the display selection by setting bit0 in the HMI byte. At the same time, the value zero is written to bit3 and bit4 in the HMI byte.
4. The PLC resets bit0 in the PLC byte.
5. HMI Embedded sl acknowledges to the PLC that the interpreter has started by setting bit1 in the HMI byte.
6. The HMI resets bit0 in the HMI Embedded sl byte.
7. Bit2 is set in the HMI byte if the corresponding display appears.

## Display deselection

The relations with the individual steps are shown in the above table by the numbers in brackets (step numbers).

1. The PLC sets bit1 for display deselection in the PLC byte if bit1 and bit2 are set in the HMI byte and bit7 is reset.
2. HMI Embedded sl acknowledges to the PLC that it has accepted the display deselection by setting bit0 in the HMI byte. At the same time, the value zero is written to bit3 and bit4 in the HMI byte.
3. HMI Embedded sl acknowledges the display deselection to the PLC by setting bit3 and resetting bit0, bit1 and bit2 in the HMI byte if the interpreter is terminated.
4. The PLC resets bit1 in the PLC byte.

After display deselection by the PLC or the EXIT command, HMI Embedded sl switches back to the last selected HMI display.

## Error when selecting a display

Bit4 is set in the HMI byte if the interpreter has not responded after 20 seconds or if the required display cannot be opened.

The states of the HMI that do not support PLC display selection are indicated to the PLC via bit7 in the HMI byte; e.g., in HMI Embedded sl during switchover between standard and ShopMill/ShopTurn interface.

## Operating areas

With **HMI Advanced**, the displays are shown in a separate operating area similar to the measuring cycle displays. This operating area can be selected manually by means of the sixth softkey (PLC displays) on the second operating area menu. If no PLC display is active when the area is selected, the text "Currently no PLC display active" is displayed in the header. The PLC is informed when the area is selected/deselected manually via bit2 of the HMI byte.

It is also possible to change to other operating areas via the MMC 103's operating area after a display has been selected by the PLC.

With **HMI Embedded sl**, it is not possible to manually select/deselect displays selected by the PLC via an operating area. The PLC display remains in the foreground even after a change of operating area has occurred.

## Configuring PLC displays

The interface is activated by means of the section [PLC\_SELECT] in the COMMON.COM file during HMI boot. HMI Embedded sl searches the directories for the COMMON.COM file in the following sequence:

- User cycles directory (CUS)
- Standard cycles directory (CST)
- Comment directory (COM)

The first file found is evaluated. The display number is assigned to the configured display in the section [PLC\_SELECT] of the COMMON.COM file.

In general, the syntax of an assignment line is:

```
PCi = display name, file ; Comment  
i , Number of the display in the interface  
Screen name , Display (screen form) identifier  
File , File in which the display is configured  
Example:  
[PLC_SELECT] ;Fixed section introduction  
PC1= CYC82, drilling.com ;Assignment line  
PC2= CYCLE90, millthre.com  
PC3=..., etc.
```

## 5.4 Expanded functions for PLC keys

Changes of task and state can also be configured for PLC keys.

The number range for the action is as follows:

For standard applications	between 0 – 17
For <b>Expand user interface</b> applications	≥ 100

The PLC-specific keys have numbers between **50 – 254**, with numbers 50 – 81 reserved for OEM applications.

### Examples

Same example as in chapter “Assigning functions to keys”

```
[KeyConfiguration]  
HK50.0 = Action:=100
```

The configuration shown here does not execute a change of task or state and therefore remains in the current task and screen. An Action 100 is forwarded.

A change of task and state could be configured here in exactly the same way.  
HK50.0 = Task:=1, State:=10, Action:=100

## 5.5 List of selectable states in HMI Advanced

In general, the following conditions apply to all tasks in HMI Advanced:

No configuration: Maintains current state.  
Configuration with 0: The basic state of an operating area is activated.

### Machine operating area

In the Machine operating area, the status is always dependent on the status of the machine (AUTO, MDA, JOG, REF). The following can be accessed via the PLC key:

1. The main screen for the relevant mode

State	Mode group	Machine function	Display
0	JOG	REF	JOG/REF main screen
0	JOG	none	JOG main screen
0	JOG	REPOS	REPOS main screen
0	MDA	none	The "MDA" start screen
0	MDA	TEACH	MDA/Teach main screen
0	MDA	REF	MDA/REF main screen
0	AUTO	none	Auto main screen

2. The main actual value screens for each mode (softkey vertical 6)

State	Mode group	Machine function	Display
10	JOG	none	Zoom actual values JOG
20	MDA	none	Zoom actual values MDA
30	AUTO	none	Zoom actual values Auto

3. Possibly still in JOG / MDA

State	Mode group	Machine function	Display
60	JOG	none	Handwheel selection JOG
70	JOG	none	Increment selection JOG
80	MDA	none	Handwheel selection MDA

Switching between the WCS and MCS is possible via the PLC.

### Parameter operating area

It is not possible to move from the current screen in this area.

### Program operating area

State	Function
10	Data selection
20	Program management
70	Protocol

### Services operating area

State	Function	
10	Data in	
20	Data out	
40	Manage data	
60	Data selection	
80	External drives	
90	Series start-up	
100	upgrading	

### Diagnostics operating area

The diagnostics main screen is the alarm overview.

The following horizontal softkeys can be used to access other states from this state:

State	Function	
10	Alarms	
20	Messages	
30	Alarm log	
40	Service displays	
50	PLC status	

Messages, alarm log and service displays can always be accessed.

### Startup operating area

The startup main screen is an overview of the available NC axes and drives.

State	Function	
0	NC axes and drives	
10	Machine data	
40	PLC status	
50	Optimization/test	(V7.1 and later)
60	HMI	

### Custom operating area

Refer to Chapter 6.

## 5.6 List of selectable states in HMI Embedded sl

There are only two configuration options for the state in HMI Embedded sl:

No configuration: Maintains current state.  
Configuration with 0: The basic state of an operating area is activated.

## 5.7 List of selectable states in ShopMill on NCU

Legend:

\* If option available (display MD is set)

Custom screen

These designated functions can be configured with the **Expand user interface** system.  
If this type of configuration is available, it is activated. Otherwise, the standard ShopMill screen will appear.

### Machine Manual

State	Function
19	Basic screen
2	T, S, M, etc.
30	Workpiece zero
5	Workpiece zero - set up edge
7	Workpiece zero – /user screen form
31	Workpiece zero – align edge / user screen form
32	Workpiece zero – spacing 2 edges / user screen form
33	Workpiece zero – rectangular edge
8	Workpiece zero – arbitrary edge /user screen form
34	Workpiece zero – /rectangular pocket
9	Workpiece zero – 1 hole /user screen form
35	Workpiece zero – 2 holes
36	Workpiece zero – 3 holes
37	Workpiece zero – 4 holes
38	Workpiece zero – /rectangular spigot
10	Workpiece zero – 1 circular spigot /user screen form
39	Workpiece zero – 2 circular spigots
40	Workpiece zero – 3 circular spigots
41	Workpiece zero – 4 circular spigots
42	Workpiece zero - set up plane
11	Workpiece zero – calibration probe, length* /user screen form
12	Workpiece zero – calibration probe, radius
50	Tool gauging
16	Tool gauging – length manual / user screen form
17	Tool gauging – diameter / user screen form
13	Tool gauging – length auto* / user screen form
14	Tool gauging – diameter auto* / user screen form
51	Tool gauging – / user screen form
15	Tool gauging – calibration probe* /user screen form
52	Tool gauging – calibration fixed point* /user screen form
60	Swiveling*
4	Positioning
18	Face milling
1	ShopMill settings
90	– /user screen form

## MDA

State	Function
20	MDA

## Machine Auto

State	Function
200	Basic screen
210	Program control
220	Block search
230	– /user screen form
242	Simultaneous recording – top view*
243	Simultaneous recording – 3-plane view*
244	Simultaneous recording – volume model*
250	Setting

## Program Manager

State	Function
300	NC directory
310	Parts programs*
320	Subprograms*
330	User directory 1 *
340	User directory 2 *
350	User directory 3 *
360	User directory 4 *
380	Standard cycles*
381	Manufacturer cycles*
382	User cycles*
383	User directory 5 *
384	User directory 6 *
385	User directory 7 *
386	User directory 8 *

## Program

State	Function
400	Machining plan / G code editor
412	Simulation – top view*
413	Simulation – 3-plane view*
414	Simulation – volume model*

## Messages/alarms

State	Function
500	Messages
510	– /user screen form
520	– /user screen form



**Tools zero offsets**

State	Function
600	Tool list
610	Tool wear
620	User tools list*
630	Magazine
640	Zero offset
650	R parameters
660	– /user screen form
680	User data
690	Machine data

**5.8 List of selectable states in ShopTurn on NCU**

Legend:

\* If option available (display MD is set)

**Custom screen**

These designated functions can be configured with the **Expand user interface** system.  
If this type of configuration is available, it is activated. Otherwise, the standard ShopTurn screen will appear.

**Maschine manual (without "Machine manual" option)**

State	Function
19	Basic screen
2	T, S, M, etc.
30	Workpiece zero
31	Workpiece zero – /user screen form
34	Workpiece zero – /user screen form
35	Workpiece zero – /user screen form
36	Workpiece zero – /user screen form
37	Workpiece zero – /user screen form
38	Workpiece zero – /user screen form
40	Workpiece zero – /user screen form
5	Workpiece zero – measure edge Z
50	Measure tool
51	Measure tool – length manual X /user screen form
52	Measure tool – length manual Z /user screen form
53	Measure tool – zoom* / user screen form
54	Measure tool – / user screen form
55	Measure tool – / user screen form
56	Measure tool – calibration probe* /user screen form
57	Measure tool – / user screen form
58	Measure tool – automatic Z *
59	Measure tool – automatic X *
4	Position
18	Face milling*
80	Cutting*
90	– /user screen form (tailstock)
1	Shopturn settings

**Machine hand (with "Machine manual" option)**

State	Function
19	Basic screen
50	Measure tool
51	Measure tool – length manual X /user screen form
52	Measure tool – length manual Z /user screen form
53	Measure tool – zoom* / user screen form
54	Measure tool – / user screen form
55	Measure tool – / user screen form
56	Measure tool – calibration probe* /user screen form
57	Measure tool – / user screen form
58	Measure tool – automatic Z *
59	Measure tool – automatic X *
1300	Straight
1400	Drilling
1410	Drilling – centered
1420	Drilling – thread centered
1433	Drilling – centering*
1434	Drilling – drilling*
1435	Drilling – reaming*
1440	Drilling – deep-hole drilling*
1453	Drilling – tapping*
1454	Drilling – thread milling*
1500	Turning
1513	Turning – cutting 1
1514	Turning – cutting 2
1515	Turning – cutting 3
1523	Turning – groove 1
1524	Turning – groove 2
1525	Turning – groove 3
1533	Turning – undercut form E
1534	Turning – undercut form F
1535	Turning – undercut thread DIN
1536	Turning – undercut thread DIN
1543	Turning – thread, longitudinal
1544	Turning – thread, taper
1545	Turning – thread, facing
1550	Turning – cut-off
1600	Milling*
1613	Milling – rectangular pocket*
1614	Milling – circular pocket*
1623	Milling – rectangular spigot*
1624	Milling – circular spigot*
1633	Milling – longitudinal groove*
1634	Milling – circumferential groove*
1640	Milling – polyhedron*
1670	Milling – engraving*
1730	Simulation – 3-window view*
1740	Simulation – side view*
1750	Simulation – front view*
90	– /user screen form (tailstock)
1	ShopTurn settings

**MDA**

State	Function
20	MDA

**Machine Auto**

State	Function
200	Current block display
210	Program control
220	Block search
230	– /user screen form
242	Simultaneous recording – 3-window view*
243	Simultaneous recording – side view*
244	Simultaneous recording – front view*
250	Setting

**Program Manager**

State	Function
300	NC directory
310	Parts programs*
320	Subprograms*
330	User directory 1 *
340	User directory 2 *
350	User directory 3 *
360	User directory 4 *
380	Standard cycles*
381	Manufacturer cycles*
382	User cycles*
383	User directory 5 *
384	User directory 6 *
385	User directory 7 *
386	User directory 8 *

**Program**

State	Function
400	Machining plan / G code editor
412	Simulation – 3-window view*
413	Simulation – side view*
414	Simulation – front view*

### Messages/alarms

State	Function
500	Messages
510	– /user screen form
520	– /user screen form

### Tools/zero offsets

State	Function
600	Tool list
610	Tool wear
620	OEM tool list*
630	Magazine
640	Zero offset
650	R parameters
660	– /user screen form
670	Spindles
680	User data
690	Machine data

## 5.9 Hotkey assignment on the MFII keyboard

Hotkey	OP label	Key on MFII
HK1	Position	<SHIFT+F11>
HK2	Program	<END> (NB)*
HK3	Offset	<Page Down> (NB)*
HK4	Prg.Manager	<Home> (NB)*
HK5	Alarm	<Page Up> (NB)*
HK6	Custom	<SHIFT+F12> or Cursor Down (NB)*
HK7	M Machine	<SHIFT+F10>
HK8	Menu Select	<F10>

\*) Key is located on numeric keypad: <NumLock> must be off.

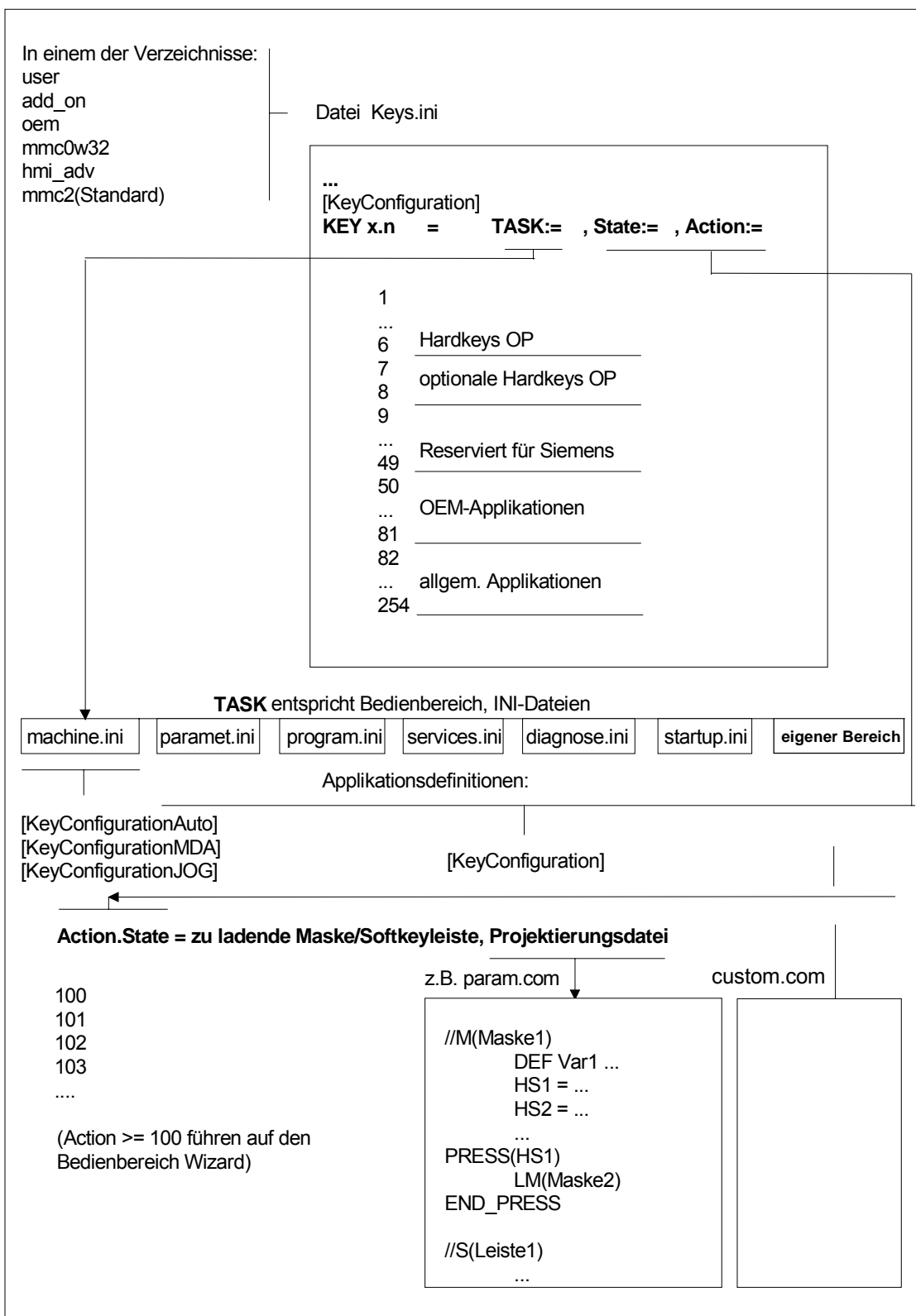
## 5.10 Examples

### Possible entries for Keys.ini

KEY1.0 = Task:=0	Selection of machine area in last display with first keystroke
KEY1.1 = Task:=0, State:=0	Selection of machine area in start display with second keystroke
KEY1.2 = State:=0, Action:=3	Selection of machine area in start display with third keystroke and selection of third softkey
KEY1.3 = State:=0, Action:=100	Selection of machine area in start display with fourth keystroke and selection of <b>Expand user interface</b> function with the configured action<= 100 in machine.ini

## 5.11 Configuration overview

Below is a schematic diagram illustrating the associations between configured OP hotkeys and PLC keys and the **Expand user interface** configuration.



## 5.12 Terms

### Hotkeys

6 keys on OP010, OP010C and SINUMERIK keyboards with hotkey blocks.  
Pressing the keys selects an operating area directly.  
As an option, 2 additional keys can be configured as hotkeys.

**References:** /BH/, Operator Components Manual

### PLC keys

PLC keys are provided as hotkeys via the PLC interface of the HMI software. Their reactions can be configured in the HMI.

They are set up as:

- Buttons on the machine control panel
- Evaluations of PLC signal logic

They are also referred to as “virtual keys”.

### SK

Softkey



## Notes



# 6

## 6 Custom Operating Area

6.1 Introduction .....	BE1/6-156
6.1.1 Delivery .....	BE1/6-156
6.1.2 Activation during startup .....	BE1/6-157
6.1.3 Activation by keys .....	BE1/6-157
6.1.4 Characteristics of the Custom area .....	BE1/6-157
6.1.5 Integration of the Custom operating area .....	BE1/6-158
6.2 Layout and functions.....	BE1/6-159
6.2.1 Starting screen form layout.....	BE1/6-159
6.3 Behavior during operating area change .....	BE1/6-160

## 6.1 Introduction

The operating areas described thus far can be extended and modified using **Expand user interface** tools. Extensions can only be applied to softkeys, which have not been used previously. With the resources described in the following, it is possible both in

- HMI Embedded sl, as well as in
- HMI Advanced

to configure an autonomous operating area in which all of the 8 horizontal and 8 vertical softkeys are available for an application-specific user interface. The default name of this operating area is **Custom**.

The Custom operating area can be accessed directly (on OPs with hotkey blocks: OP 010, OP 010C) with:

- The "Custom" hotkey or
- Horizontal softkey 4 on the first expanded menu (default setting) (softkey menu, which appears when the <ETC> key is pressed once in the main screen).

### 6.1.1 Delivery

#### HMI Advanced

The CUSTOM operating area is available as standard and can be reset via an entry in file **regie.ini** (control.ini) or assigned to any other horizontal softkey.

#### HMI Embedded sl

The CUSTOM operating area is available as standard. It is supplied with the application diskette and can be assigned to any horizontal softkey by the customer using the Software Configuration Kit (SCK.exe).

Details for integrating the Custom operating area into the system can be found in chapter "Integration of Custom operating area".

## 6.1.2 Activation during startup

### HMI Advanced

See chapter "Integration of the Custom operating area".

### HMI Embedded sl

Using the display machine data MD 9016: MM\_SWITCH\_TO\_AREA, you can define in which operator area HMI Embedded sl should start up, whereby the CUSTOM operating area can also be specified.

The value in the MD provides the number of the softkey for the required operating area.

Horizontal softkeys 1 – 8:	1 – 8
Softkeys of the 1st continue menu after pressing ETC:	9 – 16

The standard default setting is 12.

## 6.1.3 Activation by keys

### With hotkey

The CUSTOM hotkey on the OP 010 and OP 010C always activates this operating area (default). Other additional configurations are possible.

### With horizontal softkey

The standard activation is via horizontal softkey 4 in the 1st expanded menu.

The softkey assignment can be changed for

HMI-Advanced

as described in chapter "Integration of the Custom operating area" using the Software Configuration Kit (SCK).

## 6.1.4 Characteristics of the Custom area

The CUSTOM operating area displays a standard empty window with a configurable header across the whole area of the local menu.

The CUSTOM operating area also permits a configurable text in the operating area display field in the global menu.

All softkeys are empty and can be assigned as required by the customer via **Expand user interface** and given **Expand user interface** functionality.

When switching from and back to the CUSTOM area, the configured screen, which was active before exiting the CUSTOM operating area, is active.

## 6.1.5 Integration of the Custom operating area

### HMI Advanced

The operating area is enabled in the **REGIE.INI** file and is assigned to a softkey.

Example:

Activation via horizontal softkey 4 in the 1<sup>st</sup> expanded softkey menu (standard assignment).

a) The entry must be made in section **[TaskConfiguration]**:

```
Task11 = name := custom, Timeout := 12000
```

b) In file control.ini this operating area can also be selected as the standard start-up area. The entry must be made in section **[Miscellaneous]**:

```
PoweronTaskIndex = 11
```

The corresponding lines can be deactivated by adding a ";" comment line at the beginning of the line if the Custom area is not to be selected or should not be activated on start-up.

The area can be assigned to a different softkey by changing the task number.

### HMI Embedded sl

The HMI software for the CUSTOM application evaluates file **custom.ini** and decides whether the area should be displayed. The file contains a section for this purpose (HMI Embedded sl only):

#### **[Activate]**

Activate=True

The operating area is assigned to a horizontal softkey by the customer using the Software Configuration Kit (SCK.EXE), menu option "Modify configuration".

In order for the Custom operating area to be activated by default on start-up, the softkey numbers must be entered in display machine data MD 9016:

MM\_SWITCH\_TO\_AREA.

If this display MD has the value -1, HMI Embedded sl will start up in the operating area specified as the start-up area with SCK. (When delivered, softkey 1 in the main menu →area MACHINE).

## 6.2 Layout and functions

### 6.2.1 Starting screen form layout

The inputs in file **custom.ini** are used for the starting screen form and those in file **re\_xx.ini** for the operating area name in HMI Advanced.

HMI Advanced	HMI Embedded sl
<p><b>Screen form header</b> A text with the screen form header can be entered into this section <b>[Header]</b>. This can be entered as text: Text="XY special functions" or using an alarm text number: Text=\$80XXX in order that the header can be laid-out depending on the language. Text="Custom" is set as default.</p>	
<p><b>Picture in start screen form</b> A path for a screen that is displayed when the application starts up can be entered in the section <b>[Picture]</b>: Picture=\directory\bild.bmp</p>	
<p><b>Operating area name</b> The name specified appears in the top left of the screen form displayed.  In file re_xx.ini, xx indicates the language. (File re_xx.ini only exists in HMI Advanced.)</p> <p><b>[HSoftkeyTexts]</b> HSK11 = "Custom"</p>	<p><b>Operating area name</b> The name specified appears in the top left of the screen form displayed.  Section <b>[Task name]</b>  Notation: Text = \$80xxx</p>
<p><b>Softkey labeling</b>  The name entered as the operating area name in file re_xx.ini is also displayed on the configured softkey.</p>	<p><b>Softkey labeling</b> Section <b>[Softkey]</b> Notation: Text = \$80xxx  The text entered is displayed on the softkey assigned. If nothing has been entered, CUSTOM will appear as standard.</p>

All Custom area elements, which are used to access other elements, e.g.,

- Softkey menus
- I/O fields

and their associated functions must be configured using **Expand user interface** tools.

In the CUSTOM application, **all** softkeys are available as **Expand user interface** softkeys.

These softkeys are configured in file **custom.com** (like the standard version of aeditor.com supplied) as described below.

In HMI-Embedded sl, the **common.com** file for customer start softkeys must contain references to **custom.com**. The common.com file is supplied so that all CUSTOM operating area softkeys are linked to custom.com.

### 6.3 Behavior during operating area change

When you change from the **CUSTOM** operating area to another operating area and back again, the window, which was active when you closed the CUSTOM operating area, reappears the next time you open it.



# 7

## 7 Configuring and Working Environment

7.1	General information about the configuring environment.....	BE1/7-162
7.2	COMMON.COM control file .....	BE1/7-163
7.2.1	Control file function .....	BE1/7-163
7.2.2	Storage location of COMMON.COM.....	BE1/7-164
7.2.3	Configuration files .....	BE1/7-164
7.2.4	Structure of the COMMON.COM control file .....	BE1/7-165
7.3	Storing the configuration files .....	BE1/7-172
7.3.1	HMI Embedded sl .....	BE1/7-172
7.3.2	HMI Advanced .....	BE1/7-173
7.4	Common HW platform for both HMI systems .....	BE1/7-174
7.4.1	General search strategy .....	BE1/7-175
7.4.2	Search strategy for COMMON.COM .....	BE1/7-177
7.4.3	Search sequence for display images.....	BE1/7-177

## 7.1 General information about the configuring environment

### Scope of delivery

The software that interprets the configuration files for user interfaces and provides/activates the functions described in previous chapters is included in the scope of delivery of the HMI software, along with an ASCII editor for creating the content of the configuration files (program editor).

### Products

ShopMill on NCU and ShopTurn on NCU are based on HMI Embedded sl. HMI Embedded WIN32 and HMI Advanced can be combined on a single hardware platform. Thus, the operating systems of these products are available as alternatives to HMI Advanced on a single hardware platform. Particularities in the configuration and search strategies for this case are described in chapter "Shared HW platform for both HMI systems".

### Creating displays

If the configured screen forms are to include images, then a suitable graphics program (e.g., Windows Paint) will be required (see Graphics section in chapter "Syntax").

### Differences between systems

Hardware differences:

- HMI Advanced on PCU 50 has a hard disk
- HMI Embedded sl only has RAM and FlashCard memory.
- HMI Embedded WIN32 on PCU 50 or PC with Windows NT or Windows XP evaluates hard disk configuration files but operates in the same way as HMI Embedded sl in all other respects.

Because of these differences, the configuration files are created differently.

### Language-dependent texts

Language-dependent texts used in screen forms are stored in ASCII text files whose syntax corresponds to that of the alarm text files.

Language-dependent texts can be used for:

- Softkey labels
- Headings
- Help texts
- Any other texts

The file names are defined as follows:

Alsc.txt contains the language-dependent texts for the Siemens standard cycles

Almc.txt contains the language-dependent texts for the manufacturer cycles

Aluc.txt contains the language-dependent user texts



## Format of a text entry

The text number is assigned to the text within the file by the following method:

```
8xxxx  0    0    "text"
```

with:

xxxx      5000 to 9899 (text identification number range reserved for users).  
You must assign unique numbers.

Text      Text that appears in the screen form, enclosed in inverted commas

Examples:

```
85000  0    0    "Retraction level"
85001  0    0    "Drilling depth"
85002  0    0    "Thread pitch"
85003  0    0    "Pocket radius"
```

Parameters 2 and 3 are separated by blanks and act as control characters for alarm text output. To ensure that the text format is identical to that of the alarm texts, these two parameters must always be set to 0.

The following control characters are possible in the texts:

```
%n      Line break
%@x      Axis name of axis x (x is the axis number),
          (HMI-Embedded sl only)
          Display axis names, general solution
          HMI Embedded sl and HMI Advanced:
          NC access to corresponding machine data element containing the
          relevant axis name; text assembly by means of the string functions it
          contains.
```

## 7.2 COMMON.COM control file

### 7.2.1 Control file function

In HMI Embedded sl, this central control file contains the following information:

- Assignment between start softkeys and configuration files
- Assignment of display numbers to configuration files in PLC interface DB19
- Control entries (size of LOG file, available memory on the temporary directory for configuration files).

When using HMI Advanced, no entries are required in the COMMON.COM file.

Changes to COMMON.COM only take effect following a restart.

## 7.2.2 Storage location of COMMON.COM

### HMI Embedded sl

The central control file COMMON.COM must be stored in one of the following NC directories:

- CUS ;User cycles directory
- CMA ;Manufacturer cycles directory
- CST ;Standard cycles directory

The system searches through the directories for COMMON.COM in the following order. The first file with this name is evaluated.

### HMI Advanced

The central control file COMMON.COM must be stored in one of the following NC directories or in the DH path:

- CUS or  
CUS.DIR on the DH path ;User cycles directory
- CMA or  
CMA.DIR on the DH path ;Manufacturer cycles directory
- CST or  
CST.DIR on the DH path ;Standard cycles directory
- COM or  
COM.DIR on the DH path ;Comment directory

## 7.2.3 Configuration files

### HMI Embedded sl

All file names must follow DOS conventions (xxxxxxx.eee).

The total permissible number of configuration files is 10.

Color format for the Help displays: 256-color bitmaps in bmp format.

The memory requirement for the displays and configuration files is determined solely by the size of the CF card used. As with a hard disk, the possible number of files cannot be specified.

### HMI Advanced

Files with the names listed in chapter "Table of start softkeys" are searched for in directories (see chapter "Storing the configuration files") in the specified order. If files of the same name are stored in different directories, then the first file detected according to the search strategy is interpreted.

The sizes of configuration files are not subject to any particular restrictions. It should be noted, however, that large files are processed more slowly.

## 7.2.4 Structure of the COMMON.COM control file

COMMON.COM contains various sections for specific hardware settings. For HMI Embedded sl, section [MMC\_DOS] is relevant for the **Expand user interface** system.

---

### Note

The file is supplied together with the cycles. It must always be created on the NC.

---

### Syntax of COMMON.COM

Comments: Text contained within a line after a semicolon ";" is treated as a comment and is not evaluated.

[MMC\_DOS] ;Start of the DOS section

Parameters: All parameters are optional.

Parameter=*Value* ;Any number of blanks may be inserted between  
;Parameter '=' and *value*.

Possible parameters:

*SCxxx=file* or *HCyyy=file*

*SCxxx* Softkey Connection (connection between softkey and configuration file). *xxx* represents an internal softkey identity of the start softkey. See Chapter 2 "Defining a start softkey". The softkey identity must be entered directly after SC. Softkeys will not be displayed unless a softkey connection is defined for them.

*File* The configuration file, which contains the desired softkey and screen form definitions. HMI file names may only be 8 characters in length. The file extension is added with a dot.

Examples:

SC101= **my\_file.com** ; (my\_file.com on HMI, NC)

*HCyyy* Hotkey Connection (connection between actual key and configuration file). *yyy* represents an internal hotkey identity of the start hotkeys. The hotkey identity must be entered directly after HC. Only hotkeys for which a hotkey connection has been defined have an effect.

---

### Note

The file names listed in chapter "Table of start softkeys" are already stored as defaults. However, the matching files must be created on the NC or HMI by the user.

---

## Start softkeys

The start softkeys named here can be used to activate the associated configuration files.

The possible start softkeys for screen forms are predefined. Additional start softkeys are not possible. Access points are specific to operating areas.

**SCxxx=File**

**SCxxx** Softkey Connection (link between softkey and configuration file). xxx represents one of the internal softkey identities for start softkeys listed below

**File** Name of configuration file

Access points in operating area:

Operating area	SCxxx	Screen form:	
MACHINE	SC101	MACHINE JOG	Horizontal SK 1
	SC111	MACHINE MDA	Horizontal SK 1
	SC122	MACHINE AUTO	Horizontal SK 2
PARAMETERS PROGRAM	SC207	Main screen of PARAMETERS	Horizontal SK 7
	SC308	Main screen of PROGRAM	Horizontal SK 8
	SC312	1. softkey line of the editor	Horizontal SK 2
	SC313	1. softkey line of the editor	Horizontal SK 3
	SC314	1. softkey line of the editor	Horizontal SK 4
	SC315	1. softkey line of the editor	Horizontal SK 5
	SC316	1. softkey line of the editor	Horizontal SK 6
	SC327	2. Softkey menu of the editor	Horizontal SK 7
SERVICES	SC407	Main screen of SERVICES	Horizontal SK 7
DIAGNOSIS	SC507	Main screen of DIAGNOSIS	Horizontal SK 7
IBN	SC607	Main screen of START-UP	Horizontal SK 7
	SC616	2. Softkey menu of the editor	Horizontal SK 6
	SC617	2. Softkey menu of the editor	Horizontal SK 7
ShopMill	SC818	Machine Manual operating area (large screen form)	Horizontal SK 8
	SC8181	Machine Manual operating area (medium-sized screen form)	Horizontal SK 8
	SC8182	Machine Manual operating area (small screen form)	Horizontal SK 8
	SC8131	Machine Manual-Workpiece Zero operating area	Vertical SK 1
	SC8132	Machine Manual-Workpiece Zero operating area	Vertical SK 2
	SC8133	Machine Manual-Workpiece Zero operating area	Vertical SK 3
	SC8134	Machine Manual-Workpiece Zero operating area	Vertical SK 4
	SC8135	Machine Manual-Workpiece Zero operating area	Vertical SK 5
	SC8136	Machine Manual-Workpiece Zero operating area	Vertical SK 6
	SC8137	Machine Manual-Workpiece Zero operating area	Vertical SK 7

Operating area	SCxxx	Screen form:	
	SC8141	Machine Manual-Measure Tool operating area	Vertical SK 1
	SC8142	Machine Manual-Measure Tool operating area	Vertical SK 2
	SC8143	Machine Manual-Measure Tool operating area	Vertical SK 3
	SC8144	Machine Manual-Measure Tool operating area	Vertical SK 4
	SC8145	Machine Manual-Measure Tool operating area	Vertical SK 5
	SC8146	Machine Manual-Measure Tool operating area	Vertical SK 6
	SC8147	Machine Manual-Measure Tool operating area	Vertical SK 7
	SC826	Machine Auto operating area (large screen form)	Horizontal SK 6
	SC8261	Machine Auto operating area (medium-sized screen form)	Horizontal SK 6
	SC8262	Machine Auto operating area (small screen form)	Horizontal SK 6
	SC8426	Program-Drilling operating area	Vertical SK 6
	SC8436	Program-Milling operating area	Vertical SK 6
	SC8454	Program-Miscellaneous operating area	Vertical SK 4
	SC8951	Program-Miscellaneous-Workpiece Zero operating area	Vertical SK 1
	SC8952	Program-Miscellaneous-Workpiece Zero operating area	Vertical SK 2
	SC8953	Program-Miscellaneous-Workpiece Zero operating area	Vertical SK 3
	SC8954	Program-Miscellaneous-Workpiece Zero operating area	Vertical SK 4
	SC8955	Program-Miscellaneous-Workpiece Zero operating area	Vertical SK 5
	SC8956	Program-Miscellaneous-Workpiece Zero operating area	Vertical SK 6
	SC8957	Program-Miscellaneous-Workpiece Zero operating area	Vertical SK 7
	SC8961	Program-Miscellaneous-Measure Tool operating area	Vertical SK 1
	SC8962	Program-Miscellaneous-Measure Tool operating area	Vertical SK 2

Operating area	SCxxx	Screen form:	
	SC8963	Program-Miscellaneous-Measure Tool operating area	Vertical SK 3
	SC8964	Program-Miscellaneous-Measure Tool operating area	Vertical SK 4
	SC8965	Program-Miscellaneous-Measure Tool operating area	Vertical SK 5
	SC8966	Program-Miscellaneous-Measure Tool operating area	Vertical SK 6
	SC8967	Program-Miscellaneous-Measure Tool operating area	Vertical SK 7
	SC857	Messages/Alarms operating area	Horizontal SK 7
	SC858	Messages/Alarms operating area	Horizontal SK 8
	SC867	Tools/Zero offsets operating area	Horizontal SK 7
	SC8492	Program-G-Code-Editor operating area	Horizontal SK 2 (contour cycles support) *
	SC8493	Program-G-Code-Editor operating area	Horizontal SK 3 (drilling cycles support) *
	SC8494	Program-G-Code-Editor operating area	Horizontal SK 4 (milling cycles support) *
	SC8495	Program-G-Code-Editor operating area	Horizontal SK 5 (turning cycles support) *
	SC8496	Program-G-Code-Editor operating area	Horizontal SK 6*
	SC8406	Program-G-Code-Editor operating area (expanded area)	Horizontal SK 6 (measuring cycles support)
	SC8407	Program-G-Code-Editor operating area (expanded area)	Horizontal SK 7 (measuring cycles support)
ShopTurn	SC818	Machine Manual operating area (large screen form)	Horizontal SK 8
	SC8181	Machine Manual operating area (medium-sized screen form)	Horizontal SK 8
	SC8182	Machine Manual operating area (small screen form)	Horizontal SK 8
	SC8131	Machine Manual-Workpiece Zero operating area	Vertical SK 1
	SC8132	Machine Manual-Workpiece Zero operating area	Vertical SK 2
	SC8133	Machine Manual-Workpiece Zero operating area	Vertical SK 3

SC8134	Machine Manual-Workpiece Zero operating area	Vertical SK 4
SC8135	Machine Manual-Workpiece Zero operating area	Vertical SK 5
SC8136	Machine Manual-Workpiece Zero operating area	Vertical SK 6
SC8137	Machine Manual-Workpiece Zero operating area	Vertical SK 7
SC8141	Machine Manual-Measure Tool operating area	Vertical SK 1
SC8142	Machine Manual-Measure Tool operating area	Vertical SK 2
SC8143	Machine Manual-Measure Tool operating area	Vertical SK 3
SC8144	Machine Manual-Measure Tool operating area	Vertical SK 4
SC8145	Machine Manual-Measure Tool operating area	Vertical SK 5
SC8146	Machine Manual-Measure Tool operating area	Vertical SK 6
SC8147	Machine Manual-Measure Tool operating area	Vertical SK 7
SC826	Machine Auto operating area (large screen form)	Horizontal SK 6
SC8261	Machine Auto operating area (medium-sized screen form)	Horizontal SK 6
SC8262	Machine Auto operating area (small screen form)	Horizontal SK 6
SC8246	Program-Drilling operating area	Vertical SK 6
SC9436	Program-Turning operating area	Vertical SK 6
SC9456	Program-Milling operating area	Vertical SK 6
SC8454	Program-Miscellaneous operating area	Vertical SK 4
SC857	Messages/Alarms operating area	Horizontal SK 7
SC858	Messages/Alarms operating area	Horizontal SK 8
SC867	Tools/Zero offsets operating area	Horizontal SK 7
SC8492	Program-G-Code-Editor operating area	Horizontal SK 2 (contour cycles support) *
SC8493	Program-G-Code-Editor operating area	Horizontal SK 3 (drilling cycles support) *
SC8494	Program-G-Code-Editor operating area	Horizontal SK 4 (milling cycles support) *
SC8495	Program-G-Code-Editor operating area	Horizontal SK 5 (turning cycles support) *

SC8496	Program-G-Code-Editor operating area	Horizontal SK 6*
SC8406	Program-G-Code-Editor operating area (expanded area)	Horizontal SK 6 (turning measuring cycles support)*
SC8407	Program-G-Code-Editor operating area (expanded area)	Horizontal SK 7 (milling measuring cycles support)*

\* This involves Siemens screen forms.

## Displays, which can be called from the PLC

If displays are to be called from the PLC on the basis of **Expand user interface**, then it is necessary to create a section [PLC\_SELECT].

This section is for assigning the display numbers to the configured displays. In general, the syntax of an assignment line is:

<b>PC</b> <i>i</i>	= <i>display name, file</i>	<b>;</b> <i>Comment</i>
<i>i</i>		Number of the display in the interface
<i>Screen name</i>		Display (screen form) identifier
<i>File</i>		File in which the display is configured

Example:

```
[PLC_SELECT]           ;Fixed section introduction
PC1= CYC82, drilling.com ;Assignment line
PC2= CYCLE90, millthre.com
PC3=..., etc.
```

## Control entries

CHK\_FILE\_EXIST=*ram*

*ram* This control flag: specifies whether configuration files must be copied each time by the NC or whether a check must be made as to whether the files are already stored on the RAM drive of the HMI.

Possible values:

**0:**

A check whether the file already exists on the temporary drive is not carried out.

This mode is only set while the configuration files are being set up online on the NC. Changes to configuration files in the NC then become operative immediately on the HMI Embedded sl, slowing down the display building process.

**1:**

Default setting (if CHK\_FILE\_EXIST has not been set):

The configuration files are read once into the temporary directory and executed thereafter from there. This means that: Runtime is improved, but there is no reaction to changes in the configuration files in the NC.



**LOGSIZE=KB**

**KB** Size of the LOG file in kilobytes (max. size = 64 KB). A LOG file named ERROR.COM is created on the temporary directory of the HMI; its size is determined by this parameter setting.

**RAMDISK\_SIZE = kBrd**

**kBrd** Size of available RAMDISK in KB.  
RAMDISK\_SIZE can be located at any position - within or outside the sections. If the element occurs several times, then the **first** occurrence is relevant.

Example:

RAMDISK\_SIZE=200 ; → 200 KB

The standard value (if the element RAMDISK\_SIZE was not specified) is 300 KB.

If, **after** copying, the selected size has been exceeded, all of the COM files in the temporary directory are deleted **before** the next copying operation. (Files for screen forms in the background (for operating area change) are still kept.)

**Example of COMMON.COM**

```
[MMC_DOS]
sc101=tooth.com           ; MACHINE
                           ; Files are located in the
                           ; Flash memory of the HMI

sc111=mda.com
sc122=auto.com
sc207=param.com          ; PARAMETER
sc314=aeditor.com        ; PROGRAM, Editor
sc315=aeditor.com
sc316=aeditor.com
sc407=dienste.com        ; SERVICES
sc507=diagnose.com       ; DIAGNOSTICS
sc607=inbetrn.com        ; Start-up
sc826=cmm.com            ; ShopMill, Machine, AUTO
sc857=cmm.com            ; Messages/
sc858=cmm.com            ; Alarms
sc867=cmm.com            ; Tools, ZO
CHK_COMMON.COM=1         ; faster processing
                           ; by HMI
LOGSIZE=30                ; 30 kB for LOG of errors in
                           ; the screen forms
[PLC_SELECT]              ; declaration of displays that can
                           ; be called from the PLC,
                           ; see below.
PC1= CYC82, drilling.com  ; display 1
PC2= CYCLE90, millthre.com ; display 2
```

## 7.3 Storing the configuration files

### 7.3.1 HMI Embedded sl

#### Storage

User wizard files (com files for “normal” user dialogs, i.e., everything except cycles support):	/card/user/sinumerik/hmi/proj
User wizard files for user cycles support	/card/user/sinumerik/cycles/proj
Similarly, manufacturer wizard files are stored in:	/card/oem/sinumerik/cycles/proj /card/oem/sinumerik/hmi/proj
Bitmaps:	/card/user/sinumerik/cycles/ico/icoxxx /card/oem/sinumerik/cycles/ico/icoxxx /card/user/sinumerik/hmi/ico/icoxxx /card/oem/sinumerik/hmi/ico/icoxxx
Ini files:	/card/user/sinumerik/hmi/cfg /card/oem/sinumerik/hmi/cfg
Header icons:	/card/user/sinumerik/hmi/ico/icoxxx /card/oem/sinumerik/hmi/ico/icoxxx
Associated texts (aluc.txt, bzw. aluctx.s0x)	/card/user/sinumerik/cycles/lng/xxx /card/oem/sinumerik/cycles/lng/xxx /card/user/sinumerik/hmi/lng/xxx

Whereby xxx stands for the language

#### Procedure

In the Linux environment, the user wizard files are copied to the **/user/sinumerik/hmi/proj** directory (for “normal” user dialogs, i.e., everything except cycles support) of the CompactFlash card. User wizard files for user cycles support are copied to the **/user/sinumerik/cycles/proj** directory of the CompactFlash card. All files are **unpacked** and copied to the relevant directory.

Similarly, the manufacturer wizard files are copied to the **/oem/sinumerik/...** directories.

The copy operation can be performed either using Linux tools (PuTTY or WinSCP) or using the copy function in the PROGRAM operating area. In delivery condition, three USB drives and access to the CompactFlash card are set up in the PROGRAM operating area.

The following directory structure is set up on the USB medium:

```

\cycles
\cycles\proj          (Com files)
\cycles\prog          (customer cycles (.spf))
\cycles\lng           (Language directory – contains only subdirectories)
\cycles\lng\xxx      (Language directory, e.g., ger, eng, etc. – one directory for
                    each language) The aluc.txt file for the relevant language
                    is stored here.
\cycles\lico          (Display image directory – contains only subdirectories for
                    each resolution)
\cycles\lico\lico640 Directory for the display images of resolution 640*480
                    as .bmp or .bin
\cycles\lico\lico800
\cycles\lico\lico1024
or for wizard dialogs assigned to other areas in HMI:
\hmi\proj
\hmi\lng\...
\hmi\lico\...

```

In the PROGRAM operating area, the entire \cycles or \hmi directory is copied from the USB medium to the /user/sinumerik directory of the CompactFlash card.

### 7.3.2 HMI Advanced

When using HMI Advanced, no control file entries are required.

The central control file COMMON.COM must be stored in one of the following NC directories or in the DH path:

- CUS or CUS.DIR on the DH path ;User cycles directory
- CMA or CMA.DIR on the DH path ;Manufacturer cycles directory
- CST or CST.DIR on the DH path ;Standard cycles directory
- COM or COM.DIR on the DH path ;Comment directory

The directories are searched for the configuration files in the specified order. If files of the same name are stored in different directories, then the first file detected according to the search strategy is interpreted.

#### Storing the alarm text files

Alarm text files are stored in the following directory:

```
\DHMB.DIR\
```

File name:

```
ALUC_xx.COM
```

Texts can be stored in text file ALUC\_xx.COM as described in chapter "General information about the configuring environment". The text language is assigned on the basis of the text file name.

"XX" is replaced by one of the following codes in the text file name:  
e.g., ALUC\_uk.COM for English texts

Abbreviation xx	Language
gr	German
uk	English
fr	French
it	Italian
sp	Spanish
nl	Dutch
fi	Finnish
sw	Swedish
pl	Polish
tr	Turkish
ch	Chinese
tw	Chinese (Taiwan)
ko	Korean
hu	Hungarian
po	Portuguese
ru	Cyrillic
cz	Czech

### Testing configured screen forms in the PC/PG environment

You have installed the PC version of the HMI Advanced software on your PC/PG.

The directory structure is the same as on HMI Advanced.

The error log is generated in: \DH\COM.DIR\ERROR.COM

## 7.4 Common HW platform for both HMI systems

HMI Embedded WIN32 searches the configuration files for **Expand user interface** on the same paths as HMI Advanced. The starting point is the data management path.

The search sequences described below are relevant for ShopMill/ShopTurn on NCU if HMI Advanced and HMI Embedded sl are both operated in parallel on a single hardware platform. In this case, *Expand user interface* accesses the same configuration files under ShopMill/ShopTurn on NCU running on HMI Embedded sl as on HMI Advanced.

## Supplementary conditions

HMI Embedded WIN32 evaluates the **mmchome** property of section **[DHSTART]** in configuration file **DH.INI**, which contains the root directory for the data management path.

DH.INI must be stored in the BIN directory on which MMC0.EXE is started. The maximum length for the data management root directory entry is 100 characters.

### 7.4.1 General search strategy

#### Start-up

On start-up, HMI Embedded WIN32 uses the entries in the Windows Registry to ascertain where **HMI Advanced** is installed.

Within the directory indicated in the registry, a search is made for file **DH.INI** in the subdirectories **..user**, **..loem**, **..add\_on**, **..mmc2** in the sequence specified. If DH.INI cannot be found there, a search is carried out for the file in the currently selected directory of HMI Embedded WIN32.

Example:

If HMI Advanced was found on **f:\HMI\HMI Advanced**, then a search for **dh.ini** is made in the following sequence:

```
f:\HMI\HMI Advanced\user
f:\HMI\HMI Advanced\oem
f:\HMI\HMI Advanced\add_on
f:\HMI\HMI Advanced\mmc2
```

#### Control variables for search sequences

Search sequence with HMI Advanced:

CUS directory in the data management directory specified in dh.ini  
 CMA directory in the data management directory specified in dh.ini  
 CST directory in the data management directory specified in dh.ini  
 COM directory in the data management directory specified in dh.ini  
 Wizard directory on RAM disk

**Filename.bin**  
**Filename.bmp**

#### Bitmap name without path

In HMI Advance, files with the file ending **.bin** are not used

Search sequence if no archive is specified and the file name does not contain a path:

1. **Filename.bin** in the **CUS** directory of the data management path
2. **Filename.bmp** in the **CUS** directory of the data management path
3. **Filename.bin** in archive **Filename.bi\_** in the **CUS** directory of the data management path
4. **Filename.bmp** in archive **Filename.bm\_** in the **CUS** directory of the data management path

5. **Filename.bin** in the **CUS\resolution** directory of the data management path
6. **Filename.bmp** in the **CUS\resolution** directory of the data management path
7. **Filename.bin** in archive **Filename.bi\_** in the **CUS\resolution** directory of the data management path
8. **Filename.bmp** in archive **Filename.bm\_** in the **CUS\resolution** directory of the data management path
9. **Filename.bin** in the **CMA** directory of the data management path
10. **Filename.bmp** in the **CMA** directory of the data management path
11. **Filename.bin** in archive **Filename.bi\_** in the **CMA** directory of the data management path
12. **Filename.bmp** in archive **Filename.bm\_** in the **CMA** directory of the data management path
13. **Filename.bin** in the **CMA\resolution** directory of the data management path
14. **Filename.bmp** in the **CMA\resolution** directory of the data management path
15. **Filename.bin** in archive **Filename.bi\_** in the **CMA\resolution** directory of the data management path
16. **Filename.bmp** in archive **Filename.bm\_** in the **CMA\resolution** directory of the data management path
17. **Filename.bin** in the **CST** directory of the data management path
18. **Filename.bmp** in the **CST** directory of the data management path
19. **Filename.bin** in archive **Filename.bi\_** in the **CST** directory of the data management path
20. **Filename.bmp** in archive **Filename.bm\_** in the **CST** directory of the data management path
21. **Filename.bin** in the **CST\resolution** directory of the data management path
22. **Filename.bmp** in the **CST\resolution** directory of the data management path
23. **Filename.bin** in archive **Filename.bi\_** in the **CST\resolution** directory of the data management path
24. **Filename.bmp** in archive **Filename.bm\_** in the **CST\resolution** directory of the data management path
25. **Filename.bin** in the current directory (Bin directory)
26. **Filename.bmp** in the current directory (Bin directory)
27. **Filename.bin** in archive **Filename.bi\_** in the current directory (Bin directory)
28. **Filename.bmp** in archive **Filename.bm\_** in the current directory (Bin directory)
29. **Filename.bin** in archive **CUS.ARJ** in the **CUS** directory of the data management path
30. **Filename.bmp** in archive **CUS.ARJ** in the **CUS** directory of the data management path
31. **Filename.bin** in archive **CUS.ARJ** in the **CUS\resolution** directory of the data management path
32. **Filename.bmp** in archive **CUS.ARJ** in the **CUS\resolution** directory of the data management path
33. **Filename.bin** in archive **CUS.ARJ** in the **CMA** directory of the data management path
34. **Filename.bmp** in archive **CUS.ARJ** in the **CMA** directory of the data management path
35. **Filename.bin** in archive **CUS.ARJ** in the **CMA\resolution** directory of the data management path
36. **Filename.bmp** in archive **CUS.ARJ** in the **CMA\resolution** directory of the data management path
37. **Filename.bin** in archive **CUS.ARJ** in the **CST** directory of the data management path
38. **Filename.bmp** in archive **CUS.ARJ** in the **CST** directory of the data management path

39. **Filename.bin** in archive **CUS.ARJ** in the **CST\resolution** directory of the data management path
40. **Filename.bmp** in archive **CUS.ARJ** in the **CST\resolution** directory of the data management path

Steps 29 through 40 are then repeated for CMA.ARJ

41. **Filename.bin** in archive CMA.ARJ in the CUS directory of the data management path

....

52. **Filename.bmp** in archive CMA.ARJ in the CST\resolution directory of the data management path

Steps 41 through 52 are then repeated for CST.ARJ

53. **Filename.bin** in archive CST ARJ in the CUS directory of the data management path

....

64. **Filename.bmp** in archive CST.ARJ in the CST\resolution directory of the data management path

65. **Filename.bin** in archive CUS.ARJ in the current directory (Bin directory)
66. **Filename.bmp** in archive CUS.ARJ in the current directory (Bin directory)
67. **Filename.bin** in archive CMA.ARJ in the current directory (Bin directory)
68. **Filename.bmp** in archive CMA.ARJ in the current directory (Bin directory)
69. **Filename.bin** in archive CST.ARJ in the current directory (Bin directory)
70. **Filename.bmp** in archive CST.ARJ in the current directory (Bin directory)
71. **Filename.bxx**

Partial binary files, which have to be processed for reasons of compatibility with Real mode and when starting Protected mode. These files are only searched for in the currently selected directory.

## 7.4.2 Search strategy for COMMON.COM

For information about the search sequence for COMMON.COM, see HMI Embedded sl section in chapter "Storage location of COMMON.COM".

Configuration file COMMON.COM is handled separately. In order to activate the changes in this file, HMI Embedded WIN32 must be **restarted** in connection with HMI Advanced (ShopMill / ShopTurn).

A connection to the NC is established.

## 7.4.3 Search sequence for display images

The expanded search function for images (bitmaps) in HMI Embedded sl can also be used for images in the **Expand user interface** system.

If HMI Embedded WIN32 (ShopMill/ShopTurn) is installed together with HMI Advanced on a single hardware platform, a relevant data management path can be preset.

If a data management path is defined, it is prioritized and processed before the currently selected directory in order that HMI Embedded WIN32 and HMI Advanced access the **same images**.

The new search mechanism also includes the directories of the data management path that are of relevance for **Expand user interface** and the possible archives for images (cus.arj, cma.arj, cst.arj).

The search mechanism always searches for the individual files first, before searching possible archives. Consequently, the search sequence for images is as follows:

- Individual image before archive (.bin files are searched for before .bmp files). Once the search of the unpacked files has been completed, the archives containing a single file (.bi\_, then .bm\_) are searched.
- Bitmap name with followed by bitmap name without path

For more information about the search sequence for bitmaps with the various formats, see "General information about searches".





# 8

## 8 Terms, Tables, Lists, Examples

8.1 Terms.....	BE1/8-180
8.2 Table of start softkeys.....	BE1/8-183
8.3 Color table.....	BE1/8-184
8.4 List of accessible system variables.....	BE1/8-186
8.5 List of PI services .....	BE1/8-195
8.6 Examples .....	BE1/8-199

## 8.1 Terms

Table 8-1:

Term	Description
Action	Anything that is configured within a → method: → Functions, → Calculating variables, → Changing properties, ...
User variable	The user can declare user-defined variables for optional use in the → parts program or data block.
Array	Table or list of data
Attribute	Characteristic that assigns an object (→ screen form or → variable) specific, detailed → properties.
Operator interface	The term "user interface" refers in this document to the information displayed on the screen, including allocated softkeys and some special operator panel front keys.
Group	Reload unit for → configuration file
Definition lines	Program section in which → variables and → softkeys are defined.
I/O field	Input/output field for displaying and modifying values of variables.
Editor	ASCII Editor with which characters can be entered in a file and edited.
Characteristics	Characteristics of an object (e.g., of a → variable)
Start softkey	→ Softkey with which the first newly created → screen form is started.
Start softkey definition block	Program section from //S(Start).....//END in which a → Start softkey is defined
Event	Any action, which initiates execution of a → Method: Input of characters, actuation of → Softkeys, ...
Focus	Highlighting on screen, which identifies the → Screen form element that is currently selected for editing/monitoring. Screen form element marked by the cursor.
Function	Sequence of operations programmed in → Methods as a function of → Parameters.
Graphic	Graphic, which is displayed when a screen form is opened.
Master screen form	→ Screen form, which calls another screen form but remains open when a second screen is displayed.
Help display	Display, which is opened context-sensitively in relation to screen form elements.
Help loop	Several help references can be configured in a loop for each → Screen form element, i.e., the references are called up in succession and when the last reference closes, the first is displayed again.
Help variable	Internal arithmetic variable to which no → Properties can be assigned, i.e., it is not visible in the → Screen form.
HSx	Horizontal → Softkey x
Interpreter	The interpreter automatically converts the defined code from the → Configuration File into a → Screen form and controls its use.

Term	Description
Machine data	Settings of properties of the SINUMERIK system made by Siemens/machine manufacturer/end user and stored in the system. The following types of auxiliary function are available: \$MN_... General NC machine data \$MC_... Channel-specific machine data \$MA_... Axis-specific machine data \$MM_... Machine data of operator panel front In addition to this there is → setting data and drive machine data.
Screen form	Screen display of → User interface
Screen form-dependent softkey menu	Softkey menu, which is called from a newly configured screen form.
Screen form tree	A group of interlinked → Screen forms
Screen form definition block	Program section from //M.....//END for defining a → Screen form
Screen form element	Visible component of a → Variable: Short text, graphics text, → I/O field and unit text
Main body of screen form	Header, long text, graphics and → Screen form elements
Form-independent softkeys	Softkeys, which are not called from a screen form, i.e., start softkey and softkey menus, which the user configures before the first, new screen form.
Method	Programmed sequence of operations, which is executed when a corresponding → Event occurs.
HMI Embedded sl	Computer unit of operator panel front without hard disk
HMI Advanced	Computer unit of operator panel front with hard disk
NC	Numerical Control: System component of a SINUMERIK system, which controls the motional sequences of axes on the basis of → Parts programs.
NC code	Permissible language elements used to write → Parts programs of the SINUMERIK system
NC function	→PI service
User comment	Comment, which is automatically created during → NC code generation.
Parameters	Parameters are variable elements of the programming syntax and replaced by other words/symbols in the → Configuration file.
PI service	→ Function, which, on an → NC, executes a clearly defined operation. PI services can be called from the → PLC and from → HMI Advanced/HMI Embedded sl.
PLC	Programmable Logic Control: Programmable controller, which performs mainly logic operations in the SINUMERIK system.
Programming support	Provision of → Screen forms to assist programmers in writing → Parts programs with "higher-level" components Free contour programming Cycle support Blueprint programming →Recompiling →Simulation directly from the → Editor.

Term	Description
Configuration file	File or files, which contain definitions and instructions that determine the appearance of → Screen forms and their → Functions.
RECALL key	Standard key on front of operator panel
Tab card	Memory for data exchange between → Screen forms.
Recompiling	Sections of → NC code can be generated in a → Parts program from input fields in → Screen forms of the → Programming support system. Recompilation is the reverse operation. The input fields, which have created a selected section of NC code, are retrieved from the NC code and displayed in the original screen form.
Setting data	Data, which communicates the properties of the machine tool to the NC control, as defined by the system software. In contrast to → Machine data, setting data can be altered immediately by the operator.
ShopMill	Optimized SINUMERIK application and user interface for 2½D milling operations.
ShopTurn	Optimized SINUMERIK application and user interface for turning operations.
Simulation	Simulation of a → Parts program run (e.g., graphically) without movement of the actual machine axes.
Softkey	Key on front of operator panel, which activates a function displayed on the assigned area of the screen.
Softkey labels	Text/image on the screen area, which is assigned to a → Softkey.
Softkey menu	All horizontal or all vertical → softkeys
Softkey menu definition block	Program section from //S.....//END for defining a → Softkey menu
Column index	Column number of an array
Standard application	→ User interfaces that are integrated in standard software.
System variable	NC variable providing the NC states for parts program accesses and accesses from → HMI.
Parts programs	A program formulated in the NC language, which specifies motion sequences for axes and various special actions.
Toggle field	A list of values in the → I/O Field; check with toggle field: The value input in a field must be the same as one of the listed values.
Sub-screen form	→ Screen form that is called from another screen form (master screen form) without the → Master screen form being closed.
Variable	Designation of a memory location, which can be displayed in a → Screen form by assigning → Properties and in which input data and the results of arithmetic operations can be entered.
VSx	Vertical → Softkey x
Line index	Row number of an array
Access Level	Graduated system of authorization, which makes the accessibility and utilization of functions on the → user interface dependent on the authorization rights of the user.

## 8.2 Table of start softkeys

Correlation between start softkey/configuration file (default setting)

Table 8-2: HMI Advanced/HMI Embedded sl

Basic screen	Horizontal softkey	Configuration file
Machine JOG	1	MA_JOG.COM
Machine MDA	1	MA_MDA.COM
Machine Automatic	2	MA_AUTO.COM
Parameters	7	PARAM.COM
Program	8	PROG.COM
Services	7	SERVICE.COM
Diagnostics	7	DIAG.COM
Startup expanded softkey menu	7 6, 7	STARTUP.COM
Editor (reserved)	2, 3, 4, 5	AEDITOR.COM
Editor expanded softkey menu	6 6, 7	AEDITOR.COM

## 8.3 Color table

### Configuring screen forms

The following standard color table is available for HMI Advanced and HMI Embedded sl for the purpose of configuring screen forms (subset of the respective standard color table):

Table 8-3: Color table

Number	Color
1	Black
2	Red/brown
3	Dark green
4	Light gray
5	Dark gray
6	blue
7	Red
8	Brown
9	Yellow
10	white

The appearance of the colors may vary slightly with:

- HMI Advanced
- HMI Embedded sl
- ShopMill on NCU
- ShopTurn on NCU

On the EIN6, gray scales are available.

### HMI Advanced

For bitmaps in HMI Advanced, the current color table of the shipped software must be used in the character program.

### HMI Embedded sl

For bitmaps in HMI Embedded sl, the current color table of the shipped software must be used in the character program.

Color tables depend on the "New Fashion" option.  
You will find the color tables in the tool box under  
8x0d\examples\_tools\wizard.bsp\hmi\_emb\  
The file names indicate the application of each table.

- HMI\_EMB\_OLD\_FASHION.PAL :

This color palette should be used for HMI Embedded sl with Old Fashion  
Color indices 160 to 231 are available

- HMI\_EMB\_NEW\_FASHION.PAL :

This color palette should be used for HMI Embedded sl with New Fashion  
Color indices 160 to 231 are available

- HMI\_EMB\_OLD\_AND\_NEW\_FASHION.PAL :

This color palette should be used for HMI Embedded sl with both Old and New Fashion; the bitmaps generated with this color palette are **identical in appearance for Old and New Fashion.**

Color indices 160, 163, 184, 187, 196, 199, 204, 205, 207, 217, 219, 220, 221, 223, 226 and 228 are available

The previous color table HMI\_EMB.PAL is superseded by the tables specified above. Only colors 160 through 231 can be used. This is the only way to ensure that images look the same on HMI Embedded sl and HMI Advanced.

### Activating a color table from PaintShopPro

File → Open → ...\\*.bmp  
Colors → Open picture palette → ...\\*.PAL  
Apply palette with ☉ Color Indices "Open"

### System colors

There is a choice of 10 colors available for the color of a form screen element (text, input field, screen from background, etc.). The system colors have been expanded - and there is a difference between the old and new designs (refer to, e.g., the header color in the table below).

In order to achieve a demarcation between unique and **design-dependent** colors, it has been defined that the unique colors are found between 0 and 128. The design-dependent colors that have been newly introduced as part of this expansion, are defined from index 128. This means, that also when the unique colors are expanded (up to 128), it is prevented that both color types are mixed.

### Newly defined colors

Index	Color description	Color	
		Old design	New design
128	System color active field	Yellow	Orange
129	Background color	Gray	Light gray
130	Header color (active)	Yellow	blue
131	Header font color (active)	Black	white

## 8.4 List of accessible system variables

Table 8-4:

Name	Index	Description
\$A_DBB[x]	x=ByteNo	Data byte from/to PLC
\$A_DBD[x]	x=Offset	Double data word (32bit) from/to PLC
\$A_DBR[x]	x=Offset	Real data (32 bit) from/to PLC
\$A_DBW[x]	x=Offset	Data word (16 bit) from/to PLC
\$A_DLB[index]	index=Offset	Data byte in link area
\$A_DLD[index]	index=Offset	Data access to double word in link area
\$A_DLR[index]	index=Offset	Real data in link area
\$A_DLW[index]	index=Offset	Data word in link area
\$A_IN[x]	x=DigitalinputNo	Value of HW digital input
\$A_INA[x]	x=AnaloginputNo	Value of HW analog input
\$A_INCO[x]	x=InputNo	Comparator input NC
\$A_INSE		Safe Programmable Logic: External input of the NCK I/O
\$A_INSED		Safe Programmable Logic: Image of external NCK input
\$A_INSEP		Safe Programmable Logic: External input of the PLC I/O
\$A_INSEPD		Safe Programmable Logic: Image of external PLC inputs
\$A_INSI		Safe Programmable Logic: Internal NCK input safety
\$A_INSID		Safe Programmable Logic: Image of internal NCK input safety
\$A_INSIP		Safe Programmable Logic: Internal PLC input safety
\$A_LINK_TRANS_RATE		Link transfer rate
\$A_MARKERSI		Safe Programmable Logic: NCK markers
\$A_MARKERSIP		Safe Programmable Logic: Image of PLC markers
\$A_OUT[x]	x=DigitaloutputNo	Value of HW digital output
\$A_OUTA[x]	x=AnalogoutputNo	Value of HW analog output
\$A_OUTSE		Safe Programmable Logic: External output of the NCK I/O
\$A_OUTSED		Safe Programmable Logic: Image of external NCK output
\$A_OUTSEP		Safe Programmable Logic: External output of the PLC I/O
\$A_OUTSEPD		Safe Programmable Logic: Image of the external PLC output
\$A_OUTSI		Safe Programmable Logic: Internal NCK output safety
\$A_OUTSID		Safe Programmable Logic: Image of internal NCK output safety
\$A_OUTSIP		Safe Programmable Logic: Internal PLC output 611D safety
\$A_OUTSIPD		Safe Programmable Logic: Image of internal PLC output 611D safety
\$A_TIMERSI		Safe Programmable Logic: NCK timers



Table 8-5:

Name	Index	Description
\$A_PBB_IN[index]	index=Offset	IN data byte
\$A_PBB_OUT[index]	index=Offset	OUT data byte
\$A_PBD_IN[index]	index=Offset	IN double data word
\$A_PBD_OUT[index]	index=Offset	OUT double data word
\$A_PBR_IN[index]	index=Offset	IN real data
\$A_PBR_OUT[index]	index=Offset	OUT real data
\$A_PBW_IN[index]	index=Offset	IN data word
\$A_PBW_OUT[index]	index=Offset	OUT data word
\$A_TC_FCT		Command number
\$A_TC_LFN		Source location number
\$A_TC_LFO		Source location number
\$A_TC_LTN		Target location number
\$A_TC_LTO		Target location number
\$A_TC_MFN		Source magazine
\$A_TC_MFO		Source magazine number
\$A_TC_MTN		Target magazine number
\$A_TC_MTO		Target magazine number
\$A_TC_STATUS		Command status
\$A_TC_THNO		Number of toolholder
\$A_TC_TNO		T number
\$A_TOOLMLN[x]	x=ToolNo T	Current location
\$A_TOOLMN[x]	x=ToolNo T	Current magazine
\$AA_COUP_ACT[x]	x=Spindle following	Current coupling status following spindle
\$AA_COUP_OFFS[x]	x=Axis	Offset to leading axis/leading spindle, setpoint
\$AA_COUP_OFFS[x]	x=Spindle	Position offset for synchronous spindle (setpoint)
\$AA_CURR[x]	x=Axis	Actual current value of axis or spindle
\$AA_DELT[x]	x=Axis	Drive-specific distance to go in WCS
\$AA_DTBB[x]	x=Axis	Drive-specific path from start of block in basic coordinate system
\$AA_DTBW[x]	x=Axis	Drive-specific path from start of block in WCS
\$AA_DTEB[x]	x=Axis	Drive-specific path, end of block in basic coordinate system
\$AA_DTEPB[x]	x=Axis	Drive-specific dist.-to-go infeed oscillation in basic coordinate system
\$AA_DTEPW[x]	x=Axis	Drive-specific dist.-to-go infeed oscillation in WCS
\$AA_DTEW[x]	x=Axis	Drive-specific path, end of block in WCS
\$AA_EG_ACTIVE [a,b]	a=Following axis b=Leading axis	EG coupling active
\$AA_EG_AX[n,a]	n=Index leading axis a=Following axis	EG leading axis number
\$AA_EG_DENOM [a,b]	a=Following axis b=Leading axis	EG coupling factor denominator
\$AA_EG_NUM_LA[a]	a=Following axis	EG no. of leading axes
\$AA_EG_NUMERA [a,b]	a=Following axis b=Leading axis	EG coupling factor numerator
\$AA_EG_SYN[a,b]	a=Following axis b=Leading axis	EG synchronous position leading axis
\$AA_EG_SYNCDIFF[a]	a=Axis identifier	EG synchronism deviation
\$AA_EG_SYNFA[a]	a=Following axis	EG synchronous position following axis
\$AA_EG_TYPE[a,b]	a=Following axis b=Leading axis	EG coupling type

Name	Index	Description
\$AA_ESR_ENABLE[a]	a=Axis	ESR axis enable
\$AA_ESR_ENABLE[a]	a=Axis	ESR enable
\$AA_ESR_STAT[a]	a=Axis	ESR status
\$AA_ETRANS[x]	x=FrameNo	Offset of external frames
\$AA_FXS[x]	x=Axis	Status after travel to fixed stop
\$AA_IBN[x]	x=Axis	Toolholder actual value
\$AA_IEN[x]	x=Axis	Holder for active tool relative to workpiece zero point
\$AA_IM[x]	x=Axis	Toolholder
\$AA_IW[x]	x=Axis	Toolholder position, setpoint
\$AA_LEAD_P[x]	x=Axis	Real master value - position
\$AA_LEAD_SP[x]	x=Axis	Simulated master value - position
\$AA_LEAD_SV[x]	x=Axis	Simulated master value - velocity
\$AA_LEAD_TYP[x]	x=Axis	Source of master value
\$AA_LEAD_V[x]	x=Axis	Real master value - velocity
\$AA_LOAD[x]	x=Axis	Drive load in % (for 611D only)
\$AA_MM[x]	x=Axis	Measured value in MCS
\$AA_MM1[x]	x=Axis	Access to measured value in MCS
\$AA_MM2[x]	x=Axis	Access to measured value in MCS
\$AA_MM3[x]	x=Axis	Access to measured value in MCS
\$AA_MM4[x]	x=Axis	Access to measured value in MCS
\$AA_MW[x]	x=Axis	Measured value in WCS
\$AA_OFF[x]	x=Axis	Overlaid motion for programmed axis
\$AA_OFF_LIMIT[x]	x=Axis	Drive-specific limit value reached, correction for \$AA_OFF
\$AA_OSCILL_REVERSE_POS1[x]	x=Axis	Current reversal position 1 oscill. in synchronous actions
\$AA_OSCILL_REVERSE_POS2[x]	x=Axis	Current reversal position 2 oscill. in synchronous actions
\$AA_OVR[x]	x=Axis	Drive-specific override for motion-synchronous actions
\$AA_POWER[x]	x=Axis	Drive active power in [Watts]
\$AA_S[x]	x=SpindleNo	Spindle speed, setpoint
\$AA_SOFTENDN[x]	x=Axis	Software limit position, negative direction
\$AA_SOFTENDP[x]	x=Axis	Software limit position, positive direction
\$AA_STAT[x]	x=Axis	Axis status
\$AA_SYNA_MEM		Free memory space, motion synchronous actions
\$AA_SYNC[x]	x=Axis	Coupling of the slave axis for master value coupling
\$AA_TORQUE[x]	x=Axis	Drive torque setpoint in [Nm]
\$AA_TYP[x]	x=Axis	Axis type
\$AA_VACTB[x]	x=Axis	Drive-specific feed, actual value
\$AA_VACTM[x]	x=Axis	Drive-specific feed, setpoint
\$AA_VACTW[x]	x=Axis	Drive-specific feed, actual value
\$AA_VC[x]	x=Axis	Drive-specific feed, additive path feed correction
\$AC_ALARM_STAT		ESR alarm status
\$AC_AXCTSWA[CTn]	CTn=Axis container no.	Axis container status
\$AC_DELT		Distance to go, path WCS
\$AC_DRF[x]	x=Axis	DRF value
\$AC_DTBB		Distance from start of block in basic coordinate system
\$AC_DTBW		Distance from block start in WCS
\$AC_DTEB		Distance from end of block in basic

Name	Index	Description
		coordinate system
\$AC_DTEPB		Distance to go for oscillating infeed of basic coordinate system
\$AC_DTEPW		Distance to go for oscillating infeed in WCS
\$AC_DTEW		Distance from block end in WCS
\$AC_FCT0[x]	x=PolynomNo	a0 coefficient nth polynomial for synchr. action
\$AC_FCT1[x]	x=PolynomNo	a1 coefficient nth polynomial for synchr. action
\$AC_FCT2[x]	x=PolynomNo	a2 coefficient nth polynomial for synchr. action
\$AC_FCT3[x]	x=PolynomNo	a3 coefficient nth polynomial for synchr. action
\$AC_FCTLL[x]	x=PolynomNo	Lower limit value nth polynomial for synchronized action
\$AC_FCTUL[x]	x=PolynomNo	Upper limit value nth polynomial for synchronized action
\$AC_FIFOx[y]	x=FIFONo (1-10) y=ParameterNo	FIFO variables for synchronous actions
\$AC_MARKER[x]	x=MarkerNo	Counter for motion synchronous actions
\$AC_MEA		Probe has responded
\$AC_OVR		Path override for synchronous actions
\$AC_PARAM[x]	x=ParameterNo	Dyn. parameter for motion synchronization
\$AC_PATHN		Normalized path parameter
\$AC_PLTBB		Distance from start of block in basic coordinate system
\$AC_PLTEB		Distance from block end in basic coordinate system
\$AC_PRESET[x]	x=Axis	PRESET value
\$AC_PROG		Program status
\$AC_RETPOINT[x]	x=Axis	Return position on contour for repositioning
\$AC_SDIR[x]	x=SpindleNo	Rotat. status
\$AC_SMODE[x]	x=SpindleNo	Spindle mode
\$AC_STAT		Channel status
\$AC_TIME		Time from the start of the block in seconds (including the times for the internally generated intermediate blocks)
\$AC_TIMES		Time from the start of the block in seconds (without times for the internally generated intermediate blocks)
\$AC_TIMEC		Time from the start of the block in interpolation clock cycles (including clock cycles for the internally generated intermediate blocks)
\$AC_TIMESC		Time from the start of the block in interpolation clock cycles (without the clock cycles for internally generated intermediate blocks)
\$AC_TIMER[x]	x=TimerNo	Timer location in seconds
\$AC_VACTB		Interpolation feedrate, setpoint
\$AC_VACTW		Path velocity in WCS
\$AC_VC		Additive path feed correction for synchronous actions

Name	Index	Description
\$AN_AXCTAS[n]		Actual axis container address (rotary position)
\$AN_AXCTSWA[CTn]	CTn=Axis container no.	Axis container rotation active
\$AN_ESR_TRIGGER		ESR start signal
\$AN_MARKER[x]	x=MarkerNo	Marker variable for motion synchronous actions
\$MC_DISPLAY_AXIS	Bits 16-31	Machine axis identifier
\$MC_DISPLAY_AXIS	Bits 0-15	Identifier for geometry/auxiliary axis
\$MC_MM_NUM_BASE_FRAMES		Number of base frames in channel
\$MN_EXTERN_LANGUAGE	and	CNC system language
\$MN_EXTERN_CNC_SYSTEM		
\$MN_MAX_CUTTING_EDGE_NO		Max. value D number
\$MN_MAX_CUTTING_EDGE_PER_TOOL		Max. number of edges per tool
\$MN_MAX_SUMCORR_PERCUTTING_EDGE		Max. number of total offsets per cutting edge
\$MN_MM_KIND_OF_SUMCORR		Properties of total offsets in NCK
\$MN_MM_NUM_CC_MAGAZINE_PARAM		Number of parameters for one tool magazine
\$MN_MM_NUM_CC_MAGLOC_PARAM		Number of parameters for one tool magazine location
\$MN_MM_NUM_CC_MON_PARAM		Number of parameters of monitoring user data of one tool edge
\$MN_MM_NUM_GLOBAL_BASE_FRAMES		Number of channel-dependent basic frames
\$MN_MM_NUM_SUMCORR		Total number of total offsets in NCK
\$MN_MM_NUM_TOOL_ADAPTER		Max. number of traversed tool adapter data blocks
\$MN_MM_TOOL_MANAGEMENT_MASK		Settings NCK tool management
\$P_UBFR[ x ,MI]	x=FrameNo	Mirroring of settable frame
\$P_UBFR[x,RT]	x=FrameNo	Rotation of settable frame
\$P_UBFR[x,SC]	x=FrameNo	Scaling factor of settable frame
\$P_UBFR[x,SI]	x=FrameNo	Fine offset for frames
\$P_UBFR[x,TR]	x=FrameNo	Translation of settable frame
\$P_CHBFRMASK		Active channel-specific basic frames
\$P_EG_BC[a]		EG block change criterion
\$P_NCBFRMASK		Active channel-independent basic frames
\$P_OFFN		Offset normal
\$P_PFRAME[x,TR] \$P_ACTFRAME \$P_IFRAME	or or	Compilation of active frame
\$P_TOOL		Number of active tool edge
\$P_TOOLL[1]		Active tool length 1
\$P_TOOLL[2]		Active tool length 2
\$P_TOOLL[3]		Active tool length 3
\$P_TOOLND[x]	x=ToolNo	Number of cutting edges
\$P_TOOLNO		Number of active tool
\$P_TOOLR		Active tool radius
\$P_UIFR[x,y,MI]	x=FrameNo, y=Axis	Mirroring of settable frame
\$P_UIFR[x,y,RT]	x=FrameNo, y=Axis	Rotation of settable frame
\$P_UIFR[x,y,SC]	x=FrameNo, y=Axis	Scaling factor of settable frame
\$P_UIFR[x,y,SI]	x=FrameNo, y=Axis	Fine offset for frames
\$P_UIFR[x,y,TR]	x=FrameNo, y=Axis	Translation of settable frame
\$P_UIFRNUM		Index of active set frame
\$R[x]	x=ParameterNo	R parameters

Table 8-6:

Name	Index	Description
\$\$SC_PA_ACTIV_IMMED[x]	x=Number protection zone	Immediately active after referencing
\$\$SC_PA_CENT_ABS[x,0]	x=Number protection zone	Abscissa of arc center of 1st contour element
\$\$SC_PA_CENT_ABS[x,1]	x=Number protection zone	Abscissa of arc center of 2nd contour element
\$\$SC_PA_CENT_ABS[x,2]	x=Number protection zone	Abscissa of arc center of 3rd contour element
\$\$SC_PA_CENT_ABS[x,3]	x=Number protection zone	Abscissa of arc center of 4th contour element
\$\$SC_PA_CENT_ABS[x,4]	x=Number protection zone	Abscissa of arc center of 5th contour element
\$\$SC_PA_CENT_ABS[x,5]	x=Number protection zone	Abscissa of arc center of 6th contour element
\$\$SC_PA_CENT_ABS[x,6]	x = Number protection zone	Abscissa of arc center of 7th contour element
\$\$SC_PA_CENT_ABS[x,7]	x=Number protection zone	Abscissa of arc center of 8th contour element
\$\$SC_PA_CENT_ABS[x,8]	x=Number protection zone	Abscissa of arc center of 9th contour element
\$\$SC_PA_CENT_ABS[x,9]	x=Number protection zone	Abscissa of arc center of 10th contour element
\$\$SC_PA_CENT_ORD[x,0]	x=Number protection zone	Ordinate of arc center of 1st contour element
\$\$SC_PA_CENT_ORD[x,1]	x=Number protection zone	Ordinate of arc center of 2nd contour element
\$\$SC_PA_CENT_ORD[x,2]	x=Number protection zone	Ordinate of arc center of 3rd contour element
\$\$SC_PA_CENT_ORD [x,3]	x=Number protection zone	Ordinate of arc center of 4th contour element
\$\$SC_PA_CENT_ORD[x,4]	x=Number protection zone	Ordinate of arc center of 5th contour element
\$\$SC_PA_CENT_ORD[x,5]	x=Number protection zone	Ordinate of arc center of 6th contour element
\$\$SC_PA_CENT_ORD[x,6]	x=Number protection zone	Ordinate of arc center of 7th contour element
\$\$SC_PA_CENT_ORD[x,7]	x=Number protection zone	Ordinate of arc center of 8th contour element
\$\$SC_PA_CENT_ORD[x,8]	x=Number protection zone	Ordinate of arc center of 9th contour element
\$\$SC_PA_CENT_ORD[x,9]	x=Number protection zone	Ordinate of arc center of 10th contour element
\$\$SC_PA_CONT_ABS[x,0]	x=Number protection zone	Abscissa of end point of 1st contour element
\$\$SC_PA_CONT_ABS[x,1]	x=Number protection zone	Abscissa of end point of 2nd contour element
\$\$SC_PA_CONT_ABS[x,2]	x=Number protection zone	Abscissa of end point of 3rd contour element
\$\$SC_PA_CONT_ABS[x,3]	x=Number protection zone	Abscissa of end point of 4th contour element
\$\$SC_PA_CONT_ABS[x,4]	x=Number protection zone	Abscissa of end point of 5th contour element
\$\$SC_PA_CONT_ABS[x,5]	x=Number protection zone	Abscissa of end point of 6th contour element
\$\$SC_PA_CONT_ABS[x,6]	x=Number protection zone	Abscissa of end point of 7th contour element
\$\$SC_PA_CONT_ABS[x,7]	x=Number protection zone	Abscissa of end point of 8th contour element
\$\$SC_PA_CENT_ABS[x,8]	x=Number protection zone	Abscissa of end point of 9th contour element
\$\$SC_PA_CONT_ABS[x,9]	x=Number protection zone	Abscissa of end point of 10th contour element
\$\$SC_PA_CONT_NUM[x]	x=Number protection zone	Number of valid contour elements
\$\$SC_PA_CONT_ORD[x,0]	x=Number protection zone	Ordinate of end point of 1st contour element
\$\$SC_PA_CONT_ORD[x,1]	x=Number protection zone	Ordinate of end point of 2nd contour element
\$\$SC_PA_CONT_ORD[x,2]	x=Number protection zone	Ordinate of end point of 3rd contour element
\$\$SC_PA_CONT_ORD[x,3]	x=Number protection zone	Ordinate of end point of 4th contour element
\$\$SC_PA_CONT_ORD[x,4]	x=Number protection zone	Ordinate of end point of 5th contour element
\$\$SC_PA_CONT_ORD[x,5]	x=Number protection zone	Ordinate of end point of 6th contour element
\$\$SC_PA_CONT_ORD[x,6]	x=Number protection zone	Ordinate of end point of 7th contour element
\$\$SC_PA_CONT_ORD[x,7]	x=Number protection zone	Ordinate of end point of 8th contour element
\$\$SC_PA_CONT_ORD[x,8]	x=Number protection zone	Ordinate of end point of 9th contour element
\$\$SC_PA_CONT_ORD[x,9]	x=Number protection zone	Ordinate of end point of 10th contour element
\$\$SC_PA_CONT_TYP[x,0]	x=Number protection zone	Contour type of 1st contour element
\$\$SC_PA_CONT_TYP[x,1]	x=Number protection zone	Contour type of 2nd contour element
\$\$SC_PA_CONT_TYP[x,2]	x=Number protection zone	Contour type of 3rd contour element
\$\$SC_PA_CONT_TYP[x,3]	x=Number protection zone	Contour type of 4th contour element
\$\$SC_PA_CONT_TYP[x,4]	x=Number protection zone	Contour type of 5th contour element
\$\$SC_PA_CONT_TYP[x,5]	x=Number protection zone	Contour type of 6th contour element
\$\$SC_PA_CONT_TYP[x,6]	x=Number protection zone	Contour type of 7th contour element
\$\$SC_PA_CONT_TYP[x,7]	x=Number protection zone	Contour type of 8th contour element
\$\$SC_PA_CONT_TYP[x,8]	x=Number protection zone	Contour type of 9th contour element
\$\$SC_PA_CONT_TYP[x,9]	x=Number protection zone	Contour type of 10th contour element
\$\$SC_PA_LIM_3DIM[x]	x=Number protection zone	Limitation of protection zone applicate
\$\$SC_PA_MINUS_LIM[x]	x=Number protection zone	Lower limit of protection zone, applicate

Name	Index	Description
\$SC_PA_ORI[x]	x=Number protection zone	Plane assignment of protection zone
\$SC_PA_PLUS_LIM[x]	x=Number protection zone	Upper limit of protection zone, applicate
\$SC_PA_T_W[x]	x=Number protection zone	Workpiece or tool-related protection zone
\$SN_PA_ACTIV_IMMED[x]	x=Number protection zone	Immediately active after referencing
\$SN_PA_CENT_ABS[x,0]	x=Number protection zone	Abscissa of arc center of 1st contour element
\$SN_PA_CENT_ABS[x,1]	x=Number protection zone	Abscissa of arc center of 2nd contour element
\$SN_PA_CENT_ABS[x,2]	x=Number protection zone	Abscissa of arc center of 3rd contour element
\$SN_PA_CENT_ABS[x,3]	x=Number protection zone	Abscissa of arc center of 4th contour element
\$SN_PA_CENT_ABS[x,4]	x=Number protection zone	Abscissa of arc center of 5th contour element
\$SN_PA_CENT_ABS[x,5]	x=Number protection zone	Abscissa of arc center of 6th contour element
\$SN_PA_CENT_ABS[x,6]	x=Number protection zone	Abscissa of arc center of 7th contour element
\$SN_PA_CENT_ABS[x,7]	x=Number protection zone	Abscissa of arc center of 8th contour element
\$SN_PA_CENT_ABS[x,8]	x=Number protection zone	Abscissa of arc center of 9th contour element
\$SN_PA_CENT_ABS[x,9]	x=Number protection zone	Abscissa of arc center of 10th contour element
\$SN_PA_CENT_ORD[x,0]	x=Number protection zone	Ordinate of arc center of 1st contour element
\$SN_PA_CENT_ORD[x,1]	x=Number protection zone	Ordinate of arc center of 2nd contour element
\$SN_PA_CENT_ORD[x,2]	x=Number protection zone	Ordinate of arc center of 3rd contour element
\$SN_PA_CENT_ORD[x,3]	x=Number protection zone	Ordinate of arc center of 4th contour element
\$SN_PA_CENT_ORD[x,4]	x=Number protection zone	Ordinate of arc center of 5th contour element
\$SN_PA_CENT_ORD[x,5]	x=Number protection zone	Ordinate of arc center of 6th contour element
\$SN_PA_CENT_ORD[x,6]	x=Number protection zone	Ordinate of arc center of 7th contour element
\$SN_PA_CENT_ORD[x,7]	x=Number protection zone	Ordinate of arc center of 8th contour element
\$SN_PA_CENT_ORD[x,8]	x=Number protection zone	Ordinate of arc center of 9th contour element
\$SN_PA_CENT_ORD[x,9]	x=Number protection zone	Ordinate of arc center of 10th contour element
\$SN_PA_CONT_ABS[x,0]	x=Number protection zone	Abscissa of end point of 1st contour element
\$SN_PA_CONT_ABS[x,1]	x=Number protection zone	Abscissa of end point of 2nd contour element
\$SN_PA_CONT_ABS[x,2]	x=Number protection zone	Abscissa of end point of 3rd contour element
\$SN_PA_CONT_ABS[x,3]	x=Number protection zone	Abscissa of end point of 4th contour element
\$SN_PA_CONT_ABS[x,4]	x=Number protection zone	Abscissa of end point of 5th contour element
\$SN_PA_CONT_ABS[x,5]	x=Number protection zone	Abscissa of end point of 6th contour element
\$SN_PA_CONT_ABS[x,6]	x=Number protection zone	Abscissa of end point of 7th contour element
\$SN_PA_CONT_ABS[x,7]	x=Number protection zone	Abscissa of end point of 8th contour element
\$SN_PA_CONT_ABS[x,8]	x=Number protection zone	Abscissa of end point of 9th contour element
\$SN_PA_CONT_ABS[x,9]	x=Number protection zone	Abscissa of end point of 10th contour element
\$SN_PA_CONT_NUM[x]	x=Number protection zone	Number of valid contour elements
\$SN_PA_CONT_ORD[x,0]	x=Number protection zone	Ordinate of end point of 1st contour element
\$SN_PA_CONT_ORD[x,1]	x=Number protection zone	Ordinate of end point of 2nd contour element
\$SN_PA_CONT_ORD[x,2]	x=Number protection zone	Ordinate of end point of 3rd contour element
\$SN_PA_CONT_ORD[x,3]	x=Number protection zone	Ordinate of end point of 4th contour element
\$SN_PA_CONT_ORD[x,4]	x=Number protection zone	Ordinate of end point of 5th contour element
\$SN_PA_CONT_ORD[x,5]	x=Number protection zone	Ordinate of end point of 6th contour element
\$SN_PA_CONT_ORD[x,6]	x=Number protection zone	Ordinate of end point of 7th contour element
\$SN_PA_CONT_ORD[x,7]	x=Number protection zone	Ordinate of end point of 8th contour element
\$SN_PA_CONT_ORD[x,8]	x=Number protection zone	Ordinate of end point of 9th contour element
\$SN_PA_CONT_ORD[x,9]	x=Number protection zone	Ordinate of end point of 10th contour element
\$SN_PA_CONT_TYP[x,0]	x=Number protection zone	Contour type of 1st contour element
\$SN_PA_CONT_TYP[x,1]	x=Number protection zone	Contour type of 2nd contour element
\$SN_PA_CONT_TYP[x,2]	x=Number protection zone	Contour type of 3rd contour element
\$SN_PA_CONT_TYP[x,3]	x=Number protection zone	Contour type of 4th contour element
\$SN_PA_CONT_TYP[x,4]	x=Number protection zone	Contour type of 5th contour element
\$SN_PA_CONT_TYP[x,5]	x=Number protection zone	Contour type of 6th contour element
\$SN_PA_CONT_TYP[x,6]	x=Number protection zone	Contour type of 7th contour element
\$SN_PA_CONT_TYP[x,7]	x=Number protection zone	Contour type of 8th contour element
\$SN_PA_CONT_TYP[x,8]	x=Number protection zone	Contour type of 9th contour element

Name	Index	Description
\$SN_PA_CONT_TYP[x,9]	x=Number protection zone	Contour type of 10th contour element
\$SN_PA_LIM_3DIM[x]	x=Number protection zone	Limitation of protection zone, applicate
\$SN_PA_MINUS_LIM[x]	x=Number protection zone	Lower limit of protection zone, applicate
\$SN_PA_ORI[x]	x=Number protection zone	Plane assignment of protection zone
\$SN_PA_PLUS_LIM[x]	x=Number protection zone	Upper limit of protection zone, applicate
\$SN_PA_T_W[x]	x=Number protection zone	Workpiece or tool-related protection zone
\$TC_ADPT ...		Adapter data
\$TC_ADPTx \$TC_ADPTT	x=1 to 3	Number of parameters per adapter
\$TC_DPCE		Transformed edge correction value
\$TC_DPCx[y,z]	x=ParamNo y=ToolNo, z=EdgeNo	User-defined tool edge parameter
\$TC_DPx[y,z]	x=ParamNo y=ToolNo, z=EdgeNo	Edge correction value
\$TC_DPx[y,z]	x=ParamNo y=ToolNo, z=EdgeNo	Transformed edge correction value
\$TC_ECP ...		Transformed location-dependent setup correction
\$TC_MAMP3		Wear compound strategy
\$TC_MAP1		Type of magazine
\$TC_MAP2		Identifier of the magazine
\$TC_MAP3		Status of magazine
\$TC_MAP4		Link 1 between magazine and next magazine
\$TC_MAP5		Link 2 between magazine and preceding magazine
\$TC_MAP6		Magazine dimension
\$TC_MAP9		Number of active wear group
\$TC_MAPCx[y]	x=ParameterNo y=MagazineNo	Magazine user data for a tool magazine
\$TC_MOP1(x,y) ... \$TC_MOP15(x,y)	x=ToolNo y=EdgeNo	Monitoring data for each tool edge
\$TC_MOPCx[y,z]	x=ParamNo, y=T number z=Edge	Magazine user data for a tool edge
\$TC_MPPCx[y,z]	x=ParamNo y=MagazineNo z=MagPlaceNo	Magazine location user data for a tool magazine
\$TC_MPPx	x=1,...7	No. of parameters per magazine location
\$TC_SCP...		Transformed location-dependent wear correction
\$TC_SCP...		Location-dependent wear correction
\$TC_SCPx	x=13,...21,...71	Number of total offset parameters per total offset block
\$TC_TP1		DUPLO number
\$TC_TP10		Tool search type for replacement tools
\$TC_TP11		Tool information for HMI
\$TC_TP2		Tool designation
\$TC_TP3		Size to left in half locations
\$TC_TP4		Size to right in half locations
\$TC_TP5		Size upwards in half locations
\$TC_TP6		Size downwards in half locations
\$TC_TP7		Magazine location type
\$TC_TP8		Tool status
\$TC_TP9		Tool monitoring method

Name	Index	Description
\$TC_TPCx[y]	x=ParameterNo y=ToolNo	User-defined tool parameter
\$TC_TPG1		Spindle number
\$TC_TPG2		Chaining rule
\$TC_TPG3		Minimum wheel diameter
\$TC_TPG4		Minimum grinding wheel width
\$TC_TPG5		Current width of grinding wheel
\$TC_TPG6		Maximum speed of grinding wheel
\$TC_TPG7		Maximum peripheral speed of grinding wheel (SUG)
\$TC_TPG8		Angle of inclination of inclined grinding wheel
\$TC_TPG9		Compensation parameter for GWPS
\$VA_COUP_OFFS[x]	x=Axis	Offset to leading axis / leading spindle, actual value
\$VA_IS[x]	x=Axis	Safe actual position of axis
\$VA_VACTM[x]	x=Axis	Actual-value, load-end axis velocity in MCS system

**Abbreviations:**

EG	Expansion Unit
ESR	Extended stop and retract
FIFO	First In - First Out: Method of storing and retrieving data in a memory.
MCS	Machine coordinate system
GWPS	Grinding wheel surface speed
WCS	Workpiece coordinate system



## 8.5 List of PI services

PI\_SERVICE(*service, n parameter*)

with:

Service Identifier of PI service  
n parameter List of PI service parameters.  
Individual parameters are separated by commas.

Table 8-7: PI services

Service	Description
<b>Parameters</b>	<b>Example</b>
_N_ASUP_	An interrupt number in the specified channel is assigned to a parts program stored in the NCK (identified by path name and program name). This PI service is identical to program instruction 'SETINT'.
	<i>Par1</i> Interrupt number (0 – 8) <i>Par2</i> Priority (0–8) <i>Par3</i> Liffast (0, 1) <i>Par4</i> Block synchronization (0, 1) <i>Par5</i> Path data with max. 32 positions
	Assignment of interrupt 5 to program MPF_DIR/TEST_MPF in the current channel. The interrupt has priority 3 and is executed without rapid lift on the <i>PI_SERVICE("_N_ASUP_",5,3,0,0,"/_N_MPF_DIR/_N_TEST_MPF")</i>
_N_CANCEL	All alarms with the "Cancel Alarm" classification can be acknowledged with this command. Individual acknowledgment for specific alarms is not possible.
	---
	All alarms with the "Cancel Alarm" classification are reset. <i>PI_SERVICE("_N_CANCEL")</i>

Service	Description
<b>Parameters</b>	<b>Example</b>
_N_CRCEDN	<p>Create tool edge by specifying the cutting edge number. If the T number of an existing tool is entered in parameter "T Number" of this service, then a tool edge for the existing tool is created (in this case, the parameter "D number", i.e., the number of the edge to be created, has a value range of 1-9). If a positive T number is specified as a parameter and the tool for the T number entered does not exist, then the PI service is aborted. If a value of 0 is entered as the T number (model of absolute D numbers), then the D number values can range from 1– 31999. The new cutting edge is set up with the specified D number. If the specified cutting edge already exists, then the PI service is aborted in both cases.</p> <p><i>Par1</i> T number  <i>Par2</i> D number            T number == 0 ==&gt; 1–31999            T number &gt; 0 ==&gt; 1–9</p> <p>A tool edge with number 3 is created for the tool with number 17 in the current TO area.  <i>PI_SERVICE("_N_CRCEDN",17,3)</i></p>
_N_CREACE	<p>Create a new tool edge for a specified tool. The next-higher available D number is automatically set up. The following blocks in the active file system are affected by this PI service:            Tool offsets TO: The relevant tool edge (with ZERO content) is set up            Monitoring data TS: (If available) the relevant tool edge (with ZERO content) is set up            User edge data TUE: (if available) the respective tool edge is created (with content ZERO)            (NCK software versions &lt; 10.x)</p> <p><i>Par1</i> Tool number 1 to 31999</p> <p>After the tool with number 55 has been set up in T area 1, a further 2 edges are created for it. The tool therefore now possesses a total of 3 edges.  <i>PI_SERVICE("_N_CREATO",55)</i>  <i>PI_SERVICE("_N_CREACE",55)</i>  <i>PI_SERVICE("_N_CREACE",55)</i></p>
_N_CREATO	<p>Create a tool with a specific T number.            The following blocks in the active file system are affected by this PI service:</p> <ul style="list-style-type: none"> <li>• Tool directory TV: The tool is entered as an existing tool.</li> <li>• Tool offsets TO: The first tool edge D1 (with ZERO content) is set up.</li> <li>• User edge data TUE: (If available) the first tool edge D1 (with ZERO content) is set up.</li> <li>• User tool data TU: (If available) an empty data block is made available for the tool.</li> </ul>

Service	Description
Parameters	Example
	<i>Par1</i> Tool number 1 to 31999
	After the tool with number 55 has been set up in T area 1, a further 2 edges are created for it. The tool therefore now possesses a total of 3 edges. <i>PI_SERVICE("_N_CREATO",55)</i> <i>PI_SERVICE("_N_CREACE",55)</i> <i>PI_SERVICE("_N_CREACE",55)</i>
_N_DELECE	Delete a cutting edge: If the T number of an existing tool is entered in parameter "T Number" of this PI service, then the tool edge for the relevant tool is deleted (in this case, the parameter "D number, i.e., the number of edge to be deleted, has a value range of 1-9). If a positive T number is specified as a parameter and the tool for the T number entered does not exist, then the PI service is aborted. If a value of 0 is entered as the T number (model of absolute D numbers), then the D number values can range from 1-31999. If the specified cutting edge does not exist, then the PI service is aborted in both cases.
	<i>Par1</i> T number of the tool for which the associated tool cutting edge is to be deleted. A setting of 0 indicates that there is no reference to a specific tool (absolute D number). <i>Par2</i> Cutting edge number of tool cutting edge that should be deleted. Value range: T number == 0 ⇒ 1 – 31999 T number > 0 ⇒ 1 – 9
	The tool edge with number 3 for the tool with number 17 in the current TO area is deleted: <i>PI_SERVICE("_N_DELECE",17,3)</i>

Service	Description
<b>Parameters</b>	<b>Example</b>
_N_DELETEO	Deletes the tool with all tool edges in all data blocks in which it is stored. The tool is also deleted in the following blocks (if available): TO, TU, TUE, TV, TG (type 400), TD, TS. <i>Par1</i> Tool number 1 to 31999
	Deletion of tool with T number 21 in the current T area. <i>PI_SERVICE("_N_DELETEO",21)</i>
	<i>Par1</i> ID for the search mode 1: Search without calculation 2: Search with calculation 3: Search including the main block analysis
	Start search with calculation in current channel. To start the PI service in a meaningful manner, the data structure for the block search (block SPRAF ; addressing on HMI Embedded sl with variable services via /Channel/Search/..) must be filled in beforehand. <i>PI_SERVICE("_N_FINDBL",2)</i>
_N_LOGIN_	A password, which sets the current access level, is sent to the NCK. <i>Par1</i> Password (precisely 8 characters, if there are fewer than 8 characters, blanks must be added)
	Transfers a password to the NCK, thus setting another access level. <i>PI_SERVICE("_N_LOGIN_", "TESTWORD")</i>
_N_LOGOUT	The current access level is reset. ---
	The current access level is reset. <i>PI_SERVICE("_N_LOGOUT")</i>
_N_SETUFR	The SYSTEM OR USER variables 'linShift', 'mirrorImgActive', 'rotation' and 'scaleFact' in channel-specific data block FU can be used to define up to 8 zero offsets per channel. PI service _N_SETUFR must be called in order to activate these user-defined zero offsets. ---
	Activation of a user frame. <i>PI_SERVICE("_N_SETUFR")</i>

## 8.6 Examples

```

//S(Start)
HS7=("Example", sel, ac7)
PRESS(HS7)
LM("Form4")
END_PRESS
//END

//M(Form4/"Example 4: Machine control panel"/"MST.BMP")

DEF byte=(I/0/0/"Input byte, Standard=0", "Byte
  No.:", ""/wrl,lil///380,40,100/480,40,50)
DEF Feed=(IBB//0/""/"Feed
  Ovrld.", ""/wrl// "EB3"/20,180,100/130,180,100),
  Axistop=(B//0/""/"Feed
  stop", ""/wrl// "E2.2"/280,180,100/380,180,50/0,11)
DEF Spin=(IBB//0/""/"Spindle
  Override", ""/wrl// "EB0"/20,210,100/130,210,100),
  spinstop=(B//0/""/"Spindle
  stop", ""/wrl// "E2.4"/280,210,100/380,210,50/0,11)
DEF custom1=(IBB//0/""/"Customer keys
  1", ""/wrl// "EB6"/20,240,100/130,240,100)
DEF custom2=(IBB//0/""/"Customer keys
  2", ""/wrl// "EB7"/20,270,100/130,270,100)
DEF By1
DEF By2
DEF By3
DEF By6
DEF By7

HS1=("Input byte", SE1, AC4)
HS2=(" ")
HS3=(" ")
HS4=(" ")
HS5=(" ")
HS6=(" ")
HS7=(" ")
HS8=(" ")
VS1=(" ")
VS2=(" ")
VS3=(" ")
VS4=(" ")
VS5=(" ")
VS6=(" ")
VS7=(" ")
VS8=("OK", SE1, AC7)

```

```
LOAD
  By1=1
  By2=2
  By3=3
  By6=6
  By7=7
END_LOAD

PRESS(HS1)
  Byte.wr=2
END_PRESS

CHANGE(Byte)
  By1=byte+1
  By2=byte+2
  By3=byte+3
  By6=byte+6
  By7=byte+7
  Feed.VAR="EB"<<By3
  Spin.VAR="EB"<<Byte
  Custom1.VAR="EB"<<By6
  Custom2.VAR="EB"<<By7
  Axistop.VAR="E"<<By2<<".2"
  Spinstop.VAR="E"<<By2<<".4"
  Byte.wr=1
END_CHANGE

CHANGE(Axistop)
  IF Axistop==0
    Axistop.BC=9
  ELSE
    Axistop.BC=11
  ENDIF
END_CHANGE

CHANGE(Spinstop)
  IF Spinstop==0
    Spinstop.BC=9
  ELSE
    Spinstop.BC=11
  ENDIF
END_CHANGE

PRESS(VS8)
  EXIT
END_PRESS
```

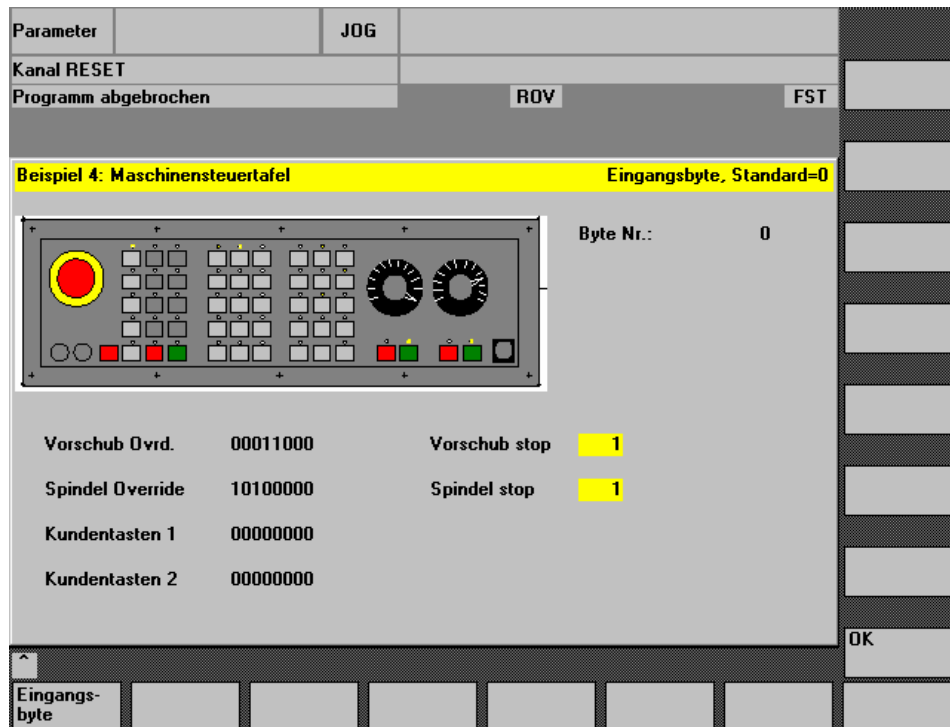


Fig. 8-1 Machine control panel

**Note**

You will find further examples in the supplied tool box.



## Notes



# I Index

## I.1 Subject Index

### A

Access level..... BE1/3-30, BE1/3-62  
 Access mode  
   direct.....BE1/3-51  
   search.....BE1/3-51  
 Action.....BE1/3-65  
 ADDHTX.....BE1/3-41  
 AP (Activate Program).....BE1/3-90  
 Array access.....BE1/3-50  
 Array status .....BE1/3-53  
 Array .....BE1/3-49  
 ASCII editor .....BE1/2-14  
 Attributes ..... BE1/3-25, BE1/3-29

### B

Background color .....BE1/3-31  
 Binary notation.....BE1/3-39  
 Bit operators .....BE1/3-95  
 Block.....BE1/3-84

### C

CALL.....BE1/3-85  
 CFI variable .....BE1/3-90  
 CHANGE .....BE1/3-67  
 CLRHTX .....BE1/3-41  
 CM .....BE1/3-25  
 Color table .....BE1/8-184  
 Colors .....BE1/3-31  
 Column alignment .....BE1/3-25  
 Command channel → *Dialog*  
   *screen*.....BE1/4-111  
 Comment .....BE1/1-10  
 COMMON.COM control file.....BE1/7-165  
 Compare mode.....BE1/3-51  
 Comparison operators.....BE1/3-97

Configuration file  
   HMI Advanced ..... BE1/7-173  
   NC code .....BE1/4-108  
   start softkey .....BE1/7-166  
 Configuration file .....BE1/7-174  
 Configuring example .....BE1/5-134  
 Configuring rules.....BE1/3-21  
 Configuring.....BE1/5-137  
 Constants .....BE1/3-97  
 Control characters.....BE1/7-163  
 CP (COPY PROGRAM).....BE1/3-89  
 Custom  
   features .....BE1/6-157  
   Hotkey .....BE1/6-157  
   Integration.....BE1/6-158  
 CUSTOM.COM .....BE1/6-160  
 CVAR .....BE1/3-81

### D

Dialog line .....BE1/3-82  
 Dialog screen .....BE1/4-111  
 Directory.....BE1/5-130  
 Display  
   Deselect.....BE1/5-141  
 Display image  
   creating .....BE1/3-45  
 Display line.....BE1/3-21  
 Display number .....BE1/5-139  
 Display .....BE1/3-37  
 DLGL.....BE1/3-82  
 DLL file .....BE1/3-94  
 DOS conventions .....BE1/7-164  
 DP (Delete Program) .....BE1/3-92

### E

ELSE .....BE1/3-98  
 ENDIF .....BE1/3-98  
 ENTRY .....BE1/3-81

- EP ..... BE1/3-92  
 Error selecting display ..... BE1/5-141  
 Error variable ..... BE1/3-80  
 EVAL ..... BE1/3-83  
 Event ..... BE1/3-65  
 EXE ..... BE1/3-75, BE1/3-103  
 EXIT ..... BE1/3-73  
 EXITLS ..... BE1/3-74  
 Expand user interface (Wizard) ..... BE1/1-8  
 Exponential notation ..... BE1/3-39  
 External functions ..... BE1/3-93
- F**
- FALSE ..... BE1/3-97  
 FCT ..... BE1/3-93  
 File  
   Check ..... BE1/3-92  
   Copy ..... BE1/3-89  
   Deleting ..... BE1/3-92  
   Selecting ..... BE1/3-91  
   transferring ..... BE1/3-90, BE1/3-91  
 File functions ..... BE1/3-87  
 FILE\_ERR ..... BE1/3-90  
 Find a character ..... BE1/3-100  
 Find text ..... BE1/4-107  
 FOC-Focus ..... BE1/3-80  
 FOCUS ..... BE1/3-68  
 Font size ..... BE1/3-30  
 Foreground color ..... BE1/3-31  
 Functional principle ..... BE1/1-9  
 Functions ..... BE1/3-71
- G**
- GC ..... BE1/3-87  
 Graphic position ..... BE1/3-24  
 Graphics text ..... BE1/3-29  
 Graphics ..... BE1/3-24
- H**
- Hardkey block ..... BE1/5-128  
 Hardkey event ..... BE1/5-131, BE1/5-132  
 Hardkeys ..... BE1/5-128  
 Header ..... BE1/3-24  
 Help display ..... BE1/3-30  
 Help loop ..... BE1/3-32  
 Help variable ..... BE1/3-44  
 Help ..... BE1/3-32, BE1/3-41  
 Hexadecimal notation ..... BE1/3-39  
 HMI Advanced alarm text files ..... BE1/7-173  
 HMI Advanced ..... BE1/1-8
- HMI byte ..... BE1/5-139  
 HMI Embedded ..... BE1/1-8  
 HMI software scope of delivery ..... BE1/7-162
- I**
- I/O field ..... BE1/3-27  
 ID parameter ..... BE1/1-10  
 Identifier  
   Parameter ..... BE1/3-21  
   Variable ..... BE1/3-28  
 IF ..... BE1/3-98  
 Image  
   Display as short text ..... BE1/3-34  
   Start screen form ..... BE1/6-159  
 INI files Sections ..... BE1/5-136  
 Input mode ..... BE1/3-29  
 Integer extensions ..... BE1/3-37  
 Intermediate storage ..... BE1/5-138  
 Interpreter ..... BE1/1-8  
 ISNUM ..... BE1/3-38  
 ISSTR ..... BE1/3-38
- K**
- Key words ..... BE1/1-10  
 KeyConfiguration ..... BE1/5-137  
 Keys.ini ..... BE1/5-131
- L**
- LA ..... BE1/3-49  
 LB ..... BE1/3-85  
 Limit values ..... BE1/3-28, BE1/3-57  
 LM ..... BE1/3-71, BE1/3-103  
 LOAD ..... BE1/3-67  
 Loading a block ..... BE1/3-85  
 Long text ..... BE1/3-29  
 LS ..... BE1/3-72, BE1/3-103
- M**
- M key ..... BE1/5-134  
 Main body of screen form ..... BE1/3-24  
 Master screen form ..... BE1/3-72  
 Mathematical functions ..... BE1/3-97  
 Mathematical operators ..... BE1/3-95  
 Memory utilization ..... BE1/3-37  
 Menu Select ..... BE1/5-134  
 Merge ..... BE1/3-73  
 Methods ..... BE1/3-65  
 MRNP ..... BE1/3-77

Multiple keypress.....BE1/5-131

## N

### NC code

Generating .....BE1/3-87

NC functions.....BE1/3-93

### NC variable

Read.....BE1/3-75

Write.....BE1/3-75

## O

### Operating area

changeover .....BE1/6-160

Custom.....BE1/6-156

Displays.....BE1/5-141

OUTPUT.....BE1/3-69

## P

Parameters .....BE1/1-10

Performance range.....BE1/1-8

PI services..... BE1/3-93, BE1/8-195

PI\_SERVICE .....BE1/8-195

PLC byte.....BE1/5-139

PLC display .....BE1/5-139

PLC interface .....BE1/5-138

PLC interface.....BE1/5-139

PLC program .....BE1/5-138

PLC softkey .....BE1/5-142

### PLC variable

Read.....BE1/3-75

Write.....BE1/3-75

Position of I/O field .....BE1/3-31

Position of short text.....BE1/3-31

Position of the cursor.....BE1/3-81

Positional parameters.....BE1/1-10

Positions .....BE1/3-41

PP (Passivate Program).....BE1/3-91

PRESS .....BE1/3-66

Protection levels .....BE1/3-63

## R

RECALL key.....BE1/3-64

Recompile.....BE1/4-108

REG.....BE1/3-78

Register status.....BE1/3-79

Register value.....BE1/3-78

Registers .....BE1/3-78

Replacing characters.....BE1/3-102

Return to standard application..... BE1/2-14

RETURN .....BE1/3-86

RNP .....BE1/3-75

Round decimal places .....BE1/3-95

ROUND.....BE1/3-95

## S

SB .....BE1/4-107

### Screen form

Defining.....BE1/3-21

Definition block .....BE1/3-21

multi-column .....BE1/3-34

Testing .....BE1/7-173

Screen form change mode .....BE1/3-71

Screen form elements.....BE1/3-27

Screen form header .....BE1/6-159

Screen form properties .....BE1/3-24

Screen form tree .....BE1/2-14

Search method.....BE1/7-177

Search sequence.....BE1/7-174

SF.....BE1/4-107

Short text.....BE1/3-29

### Softkey

Defining menu.....BE1/3-61

Horizontal.....BE1/3-21

Properties .....BE1/3-62

Vertical.....BE1/3-21

SP .....BE1/3-91

Start softkeys .....BE1/3-20

Start softkeys .....BE1/2-15

State.....BE1/5-133

Status.....BE1/3-62

String length.....BE1/3-99

Strings.....BE1/3-98

SUB.....BE1/3-84

### Subprogram

Aborting .....BE1/3-86

Calling.....BE1/3-85

Define .....BE1/3-84

Sub-screen form .....BE1/3-72

Substring from left.....BE1/3-101

Substring from mid-string.....BE1/3-102

Substring from right.....BE1/3-101

Syntax rules .....BE1/1-10

System variable .....BE1/3-31, BE1/3-44

## T

Table .....BE1/3-49

Task .....BE1/5-132

Terms.....BE1/8-180

Text alignment .....BE1/3-30

Text file  
  Language-dependent ..... BE1/1-12  
Text ..... BE1/3-29  
Toggle field  
  Extension ..... BE1/3-42  
  Limit value ..... BE1/3-28  
Transfer variable ..... BE1/3-73  
TRUE ..... BE1/3-97

## U

Unit text ..... BE1/3-29  
UNLOAD ..... BE1/3-67  
User interface ..... BE1/3-21  
User variable ..... BE1/3-31

## V

val ..... BE1/3-44, BE1/3-78  
Value range ..... BE1/5-132

Variable  
  Calculation ..... BE1/3-45  
  Changing properties ..... BE1/3-44  
  Check ..... BE1/3-81  
  defining ..... BE1/3-27  
  S\_CHAN ..... BE1/3-81  
Variable status ..... BE1/3-43

Variable type  
  variant ..... BE1/3-38  
Variable type ..... BE1/3-28  
Variable value ..... BE1/3-43  
Variables  
  default setting ..... BE1/3-29  
Virtual keys ..... BE1/5-132  
vld ..... BE1/3-45, BE1/3-79

## W

WNP ..... BE1/3-75  
Write mode ..... BE1/3-31



# SINUMERIK 840D sl

## SINUMERIK 840D/840Di/810D

### Commissioning CNC Part 2 (HMI)

### Online Help (HE1)

#### Valid for

##### *Control*

SINUMERIK 840D sl / 840DE sl  
SINUMERIK 840D powerline / 840DE powerline  
SINUMERIK 840Di powerline / 840DiE powerline  
SINUMERIK 810D powerline / 810DE powerline

##### *Software*

##### *Software version*

HMI Advanced	7.1
HMI-Embedded	7.1



<b>1 Programming Help Systems .....</b>	<b>HE1/1-5</b>
1.1 Overview .....	HE1/1-6
1.2 Quick "ASCII" help .....	HE1/1-9
1.2.1 Extending/replacing the help function .....	HE1/1-9
1.2.2 Entries in the file "IF.INI" .....	HE1/1-9
1.2.3 Extension of text files .....	HE1/1-12
1.2.4 Extension of a topic .....	HE1/1-14
1.2.5 Extension of an instruction .....	HE1/1-15
1.2.6 Parameterization of extended "PDF" help .....	HE1/1-16
1.2.7 Extension with alarm text variables .....	HE1/1-19
1.3 Extended "PDF" help .....	HE1/1-19
1.3.1 Function .....	HE1/1-19
1.3.2 Extension of programming help .....	HE1/1-20
1.4 ISO mode help .....	HE1/1-22
1.4.1 Function .....	HE1/1-22
1.4.2 Extension of ISO mode help .....	HE1/1-23
<b>2 Alarm/Machine Data Help .....</b>	<b>HE1/2-27</b>
2.1 Overview .....	HE1/2-28
2.2 Alarm help .....	HE1/2-31
2.2.1 Function .....	HE1/2-31
2.2.2 Extension of alarm help .....	HE1/2-32
2.2.3 User comments on help .....	HE1/2-33
2.2.4 Change "Win-Help" alarm help to "PDF help" .....	HE1/2-33
2.3 Machine data help .....	HE1/2-34
2.3.1 Function .....	HE1/2-34
2.3.2 Extension of machine data help .....	HE1/2-35
<b>I Index .....</b>	<b>HE1/I-37</b>
I.1 Subject index .....	HE1/I-37





# 1

## 1 Programming Help Systems

1.1 Overview .....	HE1/1-6
1.2 Quick "ASCII" help .....	HE1/1-9
1.2.1 Extending/replacing the help function.....	HE1/1-9
1.2.2 Entries in the file "IF.INI" .....	HE1/1-9
1.2.3 Extension of text files.....	HE1/1-12
1.2.4 Extension of a topic .....	HE1/1-14
1.2.5 Extension of an instruction.....	HE1/1-15
1.2.6 Parameterization of extended "PDF" help .....	HE1/1-16
1.2.7 Extension with alarm text variables .....	HE1/1-19
1.3 Extended "PDF" help .....	HE1/1-19
1.3.1 Function .....	HE1/1-19
1.3.2 Extension of programming help .....	HE1/1-20
1.4 ISO mode help .....	HE1/1-22
1.4.1 Function .....	HE1/1-22
1.4.2 Extension of ISO mode help .....	HE1/1-23

## 1.1 Overview

The extendible programming help systems listed below are available in the PCU 50.3 controller:

- Quick "ASCII" help
- Extended "PDF" help
  - Additional help (PDF file) from the quick "ASCII" help
  - Additional help on input fields of cycle parameterization screens
  - ISO mode help

The functionality and extendability of these help systems is described in the chapters below.

### Software

- HMI Advanced
- Adobe Reader 4.x (can be obtained free of charge from the Internet/specialist shops)  
The "Adobe Acrobat Reader 4.x" software is supplied as standard with the SINUMERIK 840D sl controller.
- Adobe Acrobat 4.x  
You will need the "Adobe Acrobat Distiller 4.x" software for converting documents to PDF and for editing PDF documents.

### Storage locations of help files

Programming help	F:\DH\CST.DIR\HLP.DIR (may contain HLP, PDF and TXT files)
Alarm/MD help	F:\MMC2\HLP (may contain HLP, PDF and TXT files)
OPI help	F:\HMI_ADV\HLP (may contain HLP, PDF and TXT files)

## Principle of PDF help

The PDF help requires the following components at the help file storage location:

1. Call file, e.g. PG\_GR.TXT, with the information  
<PDF file name> <ID> <search string> for each call
2. Associated PDF file

Proceed as follows to call up the help:

- Start Adobe Acrobat Reader (supplied with the controller)
- Jump to page number (must be defined by index in document)
- Search for term (Adobe software search function)

## Creation procedure:

1. Define jumps in the help document (e.g. using standard index function of DTP systems) and generate
2. Store jump file as text file
3. Assign ID numbers to the programming commands
4. Generate PDF file

### Result:

- A call file (text file) containing the data  
<PDF file name> <ID> <search string> for each call
- One or more associated PDF files

---

### Note

In the event of multiple jumps for the same topic, each page must be entered on a separate line.

**Example:**

840D_PG.PDF	44	155
840D_PG.PDF	45	180
840D_PG.PDF	46	199

---

### Hint:

In the case of a larger volume of indices, the call file can be created automatically using Microsoft Excel.

- Insert "index file" (\*.TXT) in Excel
- Insert column
- Enter the name of the PDF file on the first line
- Select the menu items "Edit" – "Fill" – "Down"
- Copy the data and
- save as "TXT" file

The syntax below is left over:

```
<PDF file name> <ID> <search string>
840D_PG.PDF 44 155
```

## Testing the jumps in the document

Proceed as follows to test whether all the search strings can be found:

- Start Adobe Acrobat
- Select the menu item "Edit" → "Find" (CTRL + F)
- Enter the search string.

---

### Note

If the search strings cannot be found, possible causes include:

- Incorrect input
- The texts were not converted properly during PDF generation and cannot be found by the search engine for that reason.

*Analysis:* Select a string you are looking for using Adobe Acrobat, copy it and paste it into a Word document.

*Remedy:*

- Generate the PDF again or
  - Generate the postscript file using Adobe printer driver AdobePS 5.1.1 and then convert it using the Adobe Distiller.
- 

## Search sequence

The help adheres to the search sequence below:

1. F:\ USER\HLP
2. F:\ OEM\HLP
3. F:\ ADD\_ON\HLP
4. F:\ MMC2\HLP
5. F:\ HMI\_ADV\HLP
6. F:\ DH\CUS.DIR\HLP.DIR (User directory)
7. F:\ DH\CMA.DIR\HLP.DIR (Manufacturer directory)
8. F:\ DH\CST.DIR\HLP.DIR (Siemens directory)
9. F:\ DH\COM.DIR\HLP.DIR

## 1.2 Quick "ASCII" help

You can call up a number of help functions in the editor to provide programming support with the editing of part programs via the "*Information*" key:

- Quick "ASCII" help
  1. **Descriptive text for instructions**, context-sensitive based on the cursor position in the part program (e.g. G9 "Exact stop – deceleration")
  2. **Overview of topics** (e.g. "Preparatory functions", "Path commands", "Path action", etc.) to which instructions are assigned.
  3. **Overview of instructions with descriptive texts**
  4. **Search** in help window via topic or text
  5. **Transfer** of selected instruction to the editor
- Extended "PDF" help
  1. Call up **additional help** (PDF file) **from the quick "ASCII" help**, e.g. jump to Programming Guide Fundamentals
  2. **Call up additional help on input fields of cycle screens** (PDF file)

### 1.2.1 Extending/replacing the help function

The machine manufacturer/end user can integrate additional M and H functions, cycles or macros in the help system.

Text files designed for this purpose which are accessed by the help function can be extended (see Subsection "Extension of text files").

The paths of the generated ASCII text files can be parameterized or added in initialization file "IF.INI" or under "*Settings*" (user file name).

### 1.2.2 Entries in the file "IF.INI"

The following sections are available in the "IF.INI" file for parameterization of the help function:

- TECHNOLOGY
- TEXTFILES
- CONTEXT
- TEXTSEARCH

### TECHNOLOGY section

The relevant technology category of the connected machine can be entered in the [TECHNOLOGY] section so as to limit the display of instruction texts to relevant technology entries, i.e. only entries pertinent to the relevant technology are displayed.

The following technology codes can be specified:

- "m" = milling
- "t" = turning
- "g" = grinding
- "s" = nibbling
- "p" = punching
- "a" = display all technologies

## TEXTFILES section

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

The help function accesses the following text files:

- Siemens "DH\CST.DIR\IFS\_GR.COM"
- Machine manufacturer "DH\CMA.DIR\IFM\_GR.COM" (any file name)
- End user "DH\CST.DIR\USER\_GR.COM" (any file name)  
(entered under "Settings")



### Important

The path for the Siemens text file "DH\CST.DIR\IFS\_GR.COM" **must** exist otherwise an error message is output.

---

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

The following table lists the parameterization options and their interpretation by the help function.

Table 1-1

Parameters for the text file	Interpretation
FileName.com	The language-dependent text file is read out
File name_XX.com (XX=abbreviation for language, e.g. "GR" for German, see Chapter 2 for language abbreviations: Alarm help, other languages)	The language-specific text file is read out

**Example:** ENDUSER=CUS.DIR\IFM\_.COM

First of all, the non-language-specific text file "IFM.COM" is opened in the path "C:\DH\CUS.DIR\" by default or, if the non-language-specific text file does not exist, the relevant language-specific text file (e.g. "IFM\_GR.COM") is opened.

Extract from TXT file "IFM\_GR.COM":

```
...  
a2//G00/Linear interpolation with rapid traverse (m) Quick help  
a2//G01/Linear interpolation with feed (m)  
...  
a/20/"PGA.PDF",2,"="="/Assignment operator with extended help  
...  
a/3/"PG.PDF",17,"ANG"/ANG/Contour angle(s) with extended cycle help  
...  
m/14/"PGZ.PDF",302,"CYCLE77"/CYCLE77/Circular spigot milling  
cycle with extended cycle help  
...
```

## CONTEXT section

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

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

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

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

## TEXTSEARCH section

The text search type can be specified in the [TEXTSEARCH] section. The following search methods can be parameterized:

- 1 = instruction texts only
- 2 = descriptive texts only
- 3 = instruction and descriptive texts

(entered under "Search").

## File IF.INI

### [TECHNOLOGY]

"m" = milling  
"t" = turning  
"g" = grinding  
"s" = nibbling  
"p" = punching  
"a" = display all technologies

### [TEXTFILES]

Siemens = DH\CST.DIR\IFS\_GR.COM  
Machine manufacturer = DH\CMA.DIR\IFM\_GR.COM  
Machine manufacturer = DH\CMA.DIR\IFI\_GR.COM  
End user = DH\CUS.DIR\USER\_GR.COM

### [CONTEXT]

= 0 Context sensitivity not active  
= 1 Context sensitivity active

### [TEXTSEARCH]

= 1 Text search for instruction texts only  
= 2 Text search for descriptive texts only  
= 3 Text search for instruction and descriptive texts

## 1.2.3 Extension of text files

These two categories are available in the text files:

1. Topic definition
2. Instruction definition

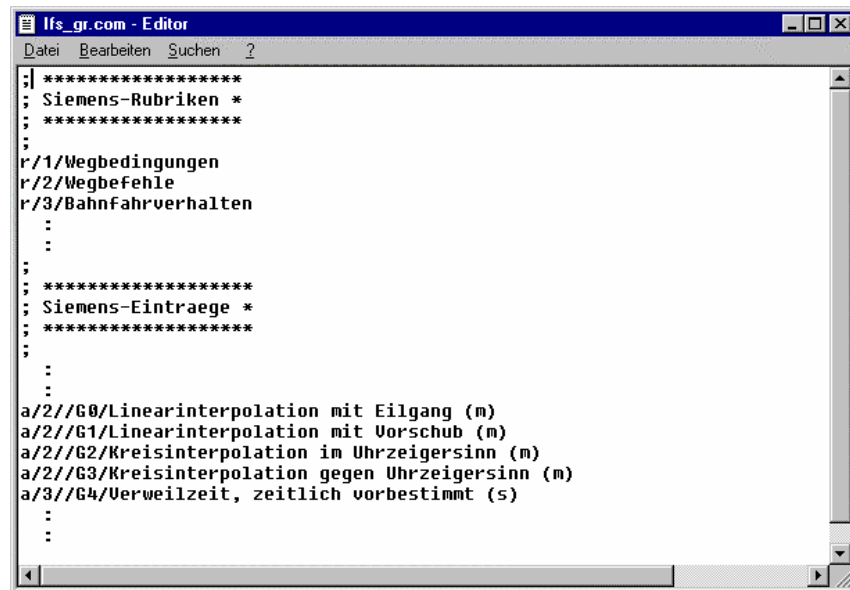
The functions below can be entered in the \*.COM text files of the help in the specified format:

- Topics as IDs of an **instruction grouping**
- Instructions with **descriptive texts**
- Instructions with call for extended help
  - **"PDF"** editor help
  - Cycle help on **input fields of cycle screens** in which parameters are supplied to the instruction
- Instructions with **alarm text variables** as descriptive texts

One line is available for each of these in the associated text file.



The screenshot below shows an example:



```

lfs_gr.com - Editor
Datei Bearbeiten Suchen ?
;| *****
; Siemens-Rubriken *
; *****
;
r/1/Wegbedingungen
r/2/Wegbefehle
r/3/Bahnfahrverhalten
:
:
;
; *****
; Siemens-Eintraege *
; *****
;
:
:
a/2//G0/Linearinterpolation mit Eilgang (m)
a/2//G1/Linearinterpolation mit Vorschub (m)
a/2//G2/Kreisinterpolation im Uhrzeigersinn (m)
a/2//G3/Kreisinterpolation gegen Uhrzeigersinn (m)
a/3//G4/Verweilzeit, zeitlich vorbestimmt (s)
:
:

```

Fig. 1-1 Siemens text file

The topic definition contains the following parameters:

### Topic definition

Technology code "r"/topic ID/topic name

The instruction definition contains the following parameters:

### Instruction definition

Technology/topic ID/extended help  
/instruction text/descriptive text

---

#### Note

1. Within each of the individual text files, the Siemens topics are displayed first, followed by those of the machine manufacturer and finally those of the end user.
  2. The specified sequence only appears if the **topics** are declared **first** and then the instructions of the text file.
  3. The instructions in the text files must be stored in **alphabetical** order if the standard, machine manufacturer and end user instructions are to be sorted correctly.
-

## 1.2.4 Extension of a topic

### Extension of a topic

A topic represents an instruction grouping which can be used for displaying instructions.

One line for the topic definition contains the following 3 parameters:

### Topic definition

*Technology code "r"/topic ID/topic name*

The following line could then appear in the text file, for example:

`r/2/Motion commands`

<code>r</code>	Permanently defined in topic definitions
<code>2</code>	Topic ID used for assigning instructions
<code>Motion commands</code>	Topic name

The "/" character separates the parameters and may only be used as a text character in the last parameter.

The length of the topic name is limited by the display and must not exceed 40 characters with a standard text size of 8 pt, for example.

The following topic IDs are reserved:

Table 1-2 Topic IDs

Topic ID	User
0..49	Siemens
50...99	Machine manufacturer
from 100	End user

The following topic IDs exist as standard in the Siemens text file:

Table 1-3 Predefined topic IDs

Topic ID	Topic name	Topic ID	Topic name
1	Preparatory functions	13	Program handling
2	Motion commands	14	Standard cycles
3	Path action	15	Measuring cycles
4	Feed	16	Measurement
5	Spindle	17	Synchronized actions
6	Axis handling	18	Interrupt handling
7	M functions	19	String/field operations
8	H functions	20	Math. functions/operators
9	Offsets	21	Definitions
10	Transformations	22	High-level language elements
11	Tool management/ compensation	23	Parameters
12	Technology functions	24	System Variables

## 1.2.5 Extension of an instruction

### Extension of an instruction

An instruction contains descriptive texts, e.g. "G0/Linear interpolation with rapid traverse". An instruction is assigned to a topic via the topic ID.

### Parameters of instruction definition

An instruction definition contains the following parameters:

*Technology/topic id/extended help/instruction text/descriptive text*

#### Example of editor help

```
a/20/"PGA.PDF",2,"="/Assignment operator
a                               Milling technology
20                              Topic ID Standard cycles
"PGA.PDF",2,"="                Extended help on command "="
=                               Instruction text
Assignment operator            Descriptive text
```

If the above line is entered in one of the user text files (e.g. IFM\_GR.COM), help file "PGA.PDF" is called up for the command "=" and the relevant help section is displayed.

#### Example of cycle help:

```
m/14/"PGZ.PDF",302,"CYCLE83"/CYCLE83/Deep-hole drilling
milling cycle
m                               Milling technology
14                              Topic ID Standard cycles
"PGZ.PDF",302,"RTP"           Extended help on parameter "RTP"
                               of the cycle parameterization screen
CYCLE83                        Instruction text CYCLE83
Deep-hole drilling
milling cycle                  Descriptive text
```

If the above line is entered in one of the user text files, user cycle "CYCLExx" is supported by the help system. It is displayed if

- either technology-independent code "a"
- or cutter-specific code "m" has been entered in initialization file "IF.INI".

It is assigned to "Standard cycles" via topic ID "14" and can therefore also be found via the corresponding entry in the topic search.

The last parameter contains the descriptive text which is displayed in the relevant column of the help system and which is designed to give the user an initial indication as to the function of the cycle.

The "/" character separates the parameters and may only be used as a text character in the last parameter.

## Technology

The following characters are defined for specifying the technology category:

"m"	= milling
"t"	= turning
"g"	= grinding
"s"	= nibbling
"p"	= punching
"a"	= display for all technologies

---

### Note

Entries are separated by commas so as to enable them to be assigned to several technologies and different topics.

---

## Topic ID

The topic ID is used for assigning instructions of a particular topic to the topic definition and for displaying these under the topic.

## Integrating additional commands

If additional commands and a brief description are to be integrated in the help, these must be incorporated using the syntax described above in file IFM\_GR.COM (any file name), which is stored in directory DH\CMA.DIR\. If an additional help (extended "PDF" help) is to be called up for these commands, the parameter "*Extended help*" must be edited as described in the section below.

### 1.2.6 Parameterization of extended "PDF" help

#### Calling extended help

A special entry in the "Extended help" parameter of the instruction definition makes it possible to call up an extended "PDF" help as an online help in addition to the quick help. This may contain detailed information on the selected command, for example. The following calls are possible:

- **Editor help:** Jump directly to the help (PDF file) from the quick "ASCII" help for a term (command) in the program editor.
- **Cycle help:** Jump to the help (PDF file) from an input field of the cycle screen.

#### Parameters of instruction definition

*Technology/topic id/extended help/instruction text/descriptive text*

## Extended help

The following entries are possible in the "Extended help" parameter:

- Editor help  
/"PDF file name", help ID, "search string"/
- Cycle help  
/"PDF file name", help ID, "search string" (file name, help screen)/

Two commas must be inserted in the "Extended help" parameter even if no entries need to be made.

### Example:

The declaration line requires the following parameters, for example:

```
m,t/14/"PDF filename", help ID, "search string" (file name, help screen)/CYCLE82/drilling cycle, drilling, counterboring
```

The meanings of the entries for the "Extended help" parameter are as follows:

PDF file name

= File from which the descriptive text is to be displayed

Help ID

= Unique assignment to term for which the help is to be displayed

Search string

= Search string in the PDF file which is to be displayed in the help text

(File name, help screen)

= Name of file in which the help screens are stored and the name of the help screen which is to be displayed (→ only relevant for cycle help)

CYCLE82

= Instruction text

Drilling cycle, drilling, counterboring

= Descriptive text

Call example:

```
m,t/14/"PGZ.PDF",1,"RTP"(drilling,CYC82)/CYCLE82/drilling cycle, drilling, counterboring
```

---

### Note

If you only want to access the PDF file, only these parameters should then appear in the "Extended help" section.

e.g.: m,t/14/"PGZ.PDF",1,"RTP"/CYCLE82/drilling cycle, drilling, counterboring

If you only want to display the help screen, only these parameters should then appear in the "Extended help" section.

e.g.: m,t/14/,,(drilling,CYC82)/CYCLE82/drilling cycle, drilling, counterboring

---

## Instruction text

An instruction text could be "G17", for example. This text is transferred to the part program currently being edited when the *"Transfer to editor"* command is issued. The instruction text can be approx. 12 characters long.

## Descriptive text

The descriptive text is the help text which describes the instruction in more detail. The length of the descriptive text is limited by the display and must not exceed 40 characters with a standard text size of 8 pt, for example.

## Multiple jumps

Several help references can be stored in ring formation (i.e. the first help is displayed again after the last help) for each help element.

Example: `m,t/14/"Hlp.pdf",1,"Help1",,2,"Help2","Help3.pdf",3,,/  
CYCLE82/Drilling cycle, drilling, counterboring`

In other words:

1. Entry in help text ring  
File: HLP.PDF  
Index: 1  
Text: Help1
2. Entry in help text ring  
File: HLP.PDF  
Index: 2  
Text: Help2
3. Entry in help text ring  
File: HELP3.PDF  
Index: 3  
Text: Help2

The components of a block are remembered by the previous block, as illustrated in the example, so as to avoid repetitions. Only complete blocks are transferred.

---

### Note

Two commas must be inserted in the "Extended help" parameter even if no entries need to be made.

---

## 1.2.7 Extension with alarm text variables

### Extension with alarm text variables

A special situation with respect to parameterization of the section for descriptive texts or topic names is to store these in the section instead of the descriptive texts ("alarm text variables").

These variables begin with "\$".

Example: **m/14//CYCLExx/\$888888**

Instead of entering the descriptive text directly in the help text file, this is entered under alarm variable "\$888888", for example.

The actual declaration of the descriptive text is therefore stored in a user-specific alarm text file.

The result is a language-independent text file.

## 1.3 Extended "PDF" help

### 1.3.1 Function

You can call up a number of help functions in the editor to provide programming support with the editing of part programs via the "*Information*" key:

- Quick "ASCII" help
  1. **Descriptive text for instructions**, context-sensitive based on the cursor position in the part program (e.g. G9 "Exact stop – deceleration")
  2. **Overview of topics** (e.g. "Preparatory functions", "Motion commands", "Path action", etc.) to which instructions are assigned
  3. Overview of instructions with descriptive texts
  4. **Search** in help window via topic or text
  5. **Transfer** of selected instruction to the editor
- Extended "PDF" help
  1. Call up **additional help** (PDF file) **from the quick "ASCII" help**, e.g. jump to Programming Guide Fundamentals
  2. **Call up additional help on input fields of cycle parameterization screens** (PDF file)

### 1.3.2 Extension of programming help

A special entry in the "Extended help" parameter of the instruction definition makes it possible to integrate an extended help in addition to the quick help (e.g. detailed information on the selected command).

The following calls are possible:

- **Editor help:** Jump directly to the help (PDF file) from the quick "ASCII" help for a term (command) in the program editor.

**Prerequisite:**

A list of additional commands which are to be displayed in the quick help has already been created as described in the "Extension of text files" subsection (e.g. IFM\_GR.COM) and is stored in the controller. This allows you to call up the extended help after the quick help using the Info key. Please note that the list of commands for the quick help is identical to the list for the extended help (concordance).

- **Cycle help:** Jump to the help (PDF file) from an input field of the cycle screen.

#### Parameters of instruction definition:

```
Technology/topic id/extended help/instruction text/  
                                descriptive text
```

"Extended help":

The following entries are possible in the "Extended help" parameter:

- Editor help  
/"PDF file name", help ID, "search string"/  
Example: a/20/"PGA.PDF",2,"="/=Assignment operator
- Cycle help:  
/"PDF file name", help ID, "search string" (file name,  
help screen)/  
Example: a/3/"PG.PDF",17,"ANG"/ANG/Contour angle (s)

#### Necessary files

- IFM\_GR.COM Text file with program commands and brief description
- \*.PDF Help file e.g. 840D\_PG\_GR.PDF
- \*.txt Jump file e.g. 840D\_PG\_GR.TXT

## 1

### Define jump labels in document

The jumps must be identified by an index entry and a single-column index list (right-justified page numbers) must be generated in the document (e.g. Winword) from which sections are to be displayed on the controller as a PDF.

If several jumps are required for one command, the same number of index labels must be created.



## 2 Generate text file

Copy the single-column index generated above (right-justified page numbers) and save it in a text file (\*.TXT).

**Example:** MEAW            155  
                  <search string> <page>

The text file for the PDF call requires the following syntax:  
          <PDF file name> <ID> <page>

Enter the name of the PDF file (**without the language code**) at the start of each line.

**Example:** 840D\_PG.PDF MEAW            155  
                  <PDF name>    <ID>            <page>

## 3 Define ID numbers

ID numbers unique to each document (e.g. 1 ... n) must be defined for the search strings. The ID numbers correspond to the list of instructions of the quick "ASCII" help, as the detailed descriptions corresponding to the brief descriptions are to be displayed:

**Example:** 840D\_PG.PDF            44    MEAW            155  
                  <PDF file name>            <ID> <search string>    <page>

Now delete the "search string" column such that the syntax below remains:

**Result:**            <PDF file name>            <ID>    <page>  
                  840D\_PG.PDF            44        155

## 4 Generate PDF file

Print your document as a postscript file and convert this to PDF format using Adobe Acrobat Distiller.

## 5 Read files into controller

The generated files (\*.PDF and \*.TXT) must now be read into the controller, e.g. into directory F:\DH\CUS.DIR\HLP.

---

### Note

In the case of a larger volume of indices, the call file can be created automatically using Microsoft Excel.

- Insert text file (\*.TXT) in Excel
  - Insert column
  - Enter the name of the PDF file on the first line
  - Select "Edit > Fill"
  - Now delete the "search string" column such that the syntax below remains:  

<PDF file name>	<ID>	<page>
840D_PG.PDF	44	155
  - Copy the data and
  - Save as "TXT" file
- 

## 1.4 ISO mode help

### 1.4.1 Function

The 840D/840D sl-specific help systems are displayed as standard:

- Help in the editor
- Alarm help
- Machine data help

If ISO dialect 0 help systems are to be displayed, it must be differentiated which data are to be displayed, as the same commands have different meanings, for example.

The [TEXTFILES] section of the "IF.INI" file has been extended by the entry SiemensI=CST.DIR\IFIT\_COM (turning technology) or SiemensI=CST.DIR\IFIM\_COM (milling technology) for this reason. This specifies the path (CST.DIR) and the name (IFIT\_COM or IFIM\_COM) for the ISO mode quick "ASCII" help.

The extended "PDF" help can thus be called from the quick "ASCII" help.

### IF.INI [TEXTFILES]

```
[TEXTFILES]
;text files containing the information shown in the help
system
Siemens=cst.dir\ifs_.com
SiemensI=cst.dir\ifit_.com or ifim_.com
Manufacturer=
End user=
```

## Necessary files

- List of "ISO dialect 0" commands with the associated brief description
- TXT file for the jump to the PDF file
- PDF file which is to be displayed in the help

## Program execution

The help checks the following machine data on startup:

- MD 1880 MM\_EXTERN\_LANGUAGE  
If the MD is activated, the controller switches over to the ISO dialect 0 mode.  
If MD 1880 = 1, MD 20150 is evaluated.
- MD 20150 GCODE\_RESET\_VALUES[n(active channel),47]  
If MD 20150 [47] = 2, the file entered in the INI file under "SiemensI" (ISO mode) is used  
(otherwise the file under "Siemens" is used).
- MD 10880 MM\_EXTERN\_CNC\_LANGUAGE  
MD 10880 specifies whether the controller is to be designed for turning or milling in ISO mode.

### 1.4.2 Extension of ISO mode help

If additional descriptions are to be displayed in the ISO mode help, these must be inserted in the form specified below:

- As a list of ISO mode commands with a brief description in the directory "SiemensI=CST.DIR\" as file "IFIT\_COM" or file "IFIM\_COM "  
(see Subsection "Extension of text files")
- As a help description (PDF file) and a TXT file of the same name in the directory "F:\USER\HLP"

### Parameters of instruction definition:

```
Technology/topic id/extended help/instruction text/  
descriptive text
```

"Extended help":

The following entries are possible in the "Extended help" parameter:

- Editor help  
/"PDF file name", help ID, "page number"/
- Cycle help:  
/"PDF file name", help ID, "page number" (file name, help screen)/



---

**Note**

In the case of a larger volume of indices, the call file can be created automatically using Microsoft Excel.

- Insert text file (\*.TXT) in Excel
- Insert column
- Enter the name of the PDF file on the first line
- Select "Edit > Fill"
- Now delete the "search string" column such that the syntax below remains:  

<PDF file name>	<ID>	<page>
840D_PGIT.PDF	44	155
- Copy the data and
- Save as "TXT" file





# 2

## 2 Alarm/Machine Data Help

2.1 Overview .....	HE1/2-28
2.2 Alarm help .....	HE1/2-31
2.2.1 Function .....	HE1/2-31
2.2.2 Extension of alarm help .....	HE1/2-32
2.2.3 User comments on help .....	HE1/2-33
2.2.4 Change "Win-Help" alarm help to "PDF help" .....	HE1/2-33
2.3 Machine data help .....	HE1/2-34
2.3.1 Function .....	HE1/2-34
2.3.2 Extension of machine data help .....	HE1/2-35

## 2.1 Overview

The functionality and extendability of the alarm and machine data help are described in the chapters below.

### Software

- HMI Advanced
- Adobe Reader 4.x (can be obtained free of charge from the Internet/specialist shops)  
The "Adobe Acrobat Reader 4.x" software is supplied as standard with the SINUMERIK 840D / 840D sl controllers.
- Adobe Acrobat 4.x  
You will need the "Adobe Acrobat Distiller 4.x" software for converting documents to PDF and for editing PDF documents.

### Storage locations of help files

Programming help F:\DH\CST.DIR\HLP.DIR  
(may contain HLP, PDF and TXT files)

Alarm/MD help F:\MMC2\HLP (may contain HLP, PDF and TXT files)

OPI help F:\HMI\_ADV\HLP (may contain HLP, PDF and TXT files)



## Principle of PDF help

The PDF help requires the following components at the help file storage location:

1. Call file, e.g. PG\_GR.TXT, with the information  
     <PDF file name> <ID> <search term> for each call
2. Associated PDF file

Proceed as follows to call up the help:

- Start "Adobe Acrobat Reader" (supplied with the controller)
- Jump to page number (must be defined by index in document)
- Search for term (Adobe software search function)

## Creation procedure:

1. Define jumps in the help document (e.g. using standard index function of DTP systems) and generate
2. Store jump file as text file
3. Generate PDF file

### Result:

- A call file (text file) containing the data  
     <PDF file name> <ID> <search string> for each call
- One or more associated PDF files

---

### Note

In the event of multiple jumps for the same topic, each page must be entered on a separate line.

**Example:**

840D_PG.PDF	44	155
840D_PG.PDF	45	180
840D_PG.PDF	46	199

---

### Hint:

In the case of a larger volume of indices, the call file can be created automatically using Microsoft Excel.

- Insert "index file" (\*.TXT) in Excel
- Insert column
- Enter the name of the PDF file on the first line
- Select the menu items "Edit" – "Fill" –> "Down"
- Copy the data and
- Save as "TXT" file

The syntax below is left over:

<PDF file name>	<ID>	<search string>
840D_PG.PDF	44	155

## Testing the jumps in the document

Proceed as follows to test whether all the search strings can be found:

- Start Adobe Acrobat
- Select the menu item "Edit" → "Find" (CTRL + F)
- Enter the search string.

---

### Note

If the search strings cannot be found, possible causes include:

- Incorrect input
- The texts were not converted properly during PDF generation and cannot be found by the search engine for that reason.

*Analysis:* Select a string you are looking for using Adobe Acrobat, copy it and paste it into a Word document.

*Remedy:*

- Generate the PDF again or
  - Generate the postscript file using Adobe printer driver Adobe PS 5.1.1 and then convert it using the Adobe Distiller.
- 

## Search sequence

The help adheres to the search sequence below:

1. F:\ USER\HLP
2. F:\ OEM\HLP
3. F:\ ADD\_ON\HLP
4. F:\ MMC2\HLP
5. F:\ HMI\_ADV\HLP
6. F:\ DH\CUS.DIR\HLP.DIR (User directory)
7. F:\ DH\CMA.DIR\HLP.DIR (Manufacturer directory)
8. F:\ DH\CST.DIR\HLP.DIR (Siemens directory)
9. F:\ DH\COM.DIR\HLP.DIR

## 2.2 Alarm help

### 2.2.1 Function

When help is called up, a PDF file is opened and the relevant alarm is displayed on the controller in Adobe Reader.

The files containing the alarms are stored in the directories below:

- Siemens alarms F:\HMI\_ADV\HLP
- Manufacturer alarms F:\DH\CMA.DIR\HLP
- User alarms F:\DH\USER\HLP

### File names

The file names below must be adhered to (must **not** be changed), even for the purposes of additional alarms:

- ALARM\_GR.PDF [German alarms]
- ALARM\_UK.PDF [English alarms]
- ALARM\_FR.PDF [French alarms]
- ALARM\_IT.PDF [Italian alarms]
- ALARM\_SP.PDF [Spanish alarms]

The names of the associated TXT files (contain the jump list) must therefore be as follows:

- ALARM\_GR.TXT [German TXT file]
- ALARM\_UK.TXT [English TXT file]
- ALARM\_FR.TXT [French TXT file]
- ALARM\_IT.TXT [Italian TXT file]
- ALARM\_SP.TXT [Spanish TXT file]

### Other languages

The alarm texts are available in the following languages:

CH:	Simplified Chinese
TW:	Standard Chinese
KO:	Korean
JA:	Japanese
SW:	Swedish
HU:	Hungarian
PO:	Portuguese
CZ:	Czech
TR:	Turkish
RU:	Russian
PL:	Polish
NL:	Dutch

The naming convention specified above for language abbreviations must be used for the file names of the user alarms (must **not** be changed).

## 2.2.2 Extension of alarm help

If additional user alarms are to be displayed in the alarm help, these must be inserted in the directory C:\USER\HLP in the form of a PDF file and a TXT file of the same name.

### Necessary files

- \*.PDF e.g. ALARM\_GR.PDF
- \*.TXT e.g. ALARM\_GR.TXT

### 1

#### Define jump labels in document

The jumps must be identified by an index entry and a single-column index list (right-justified page numbers) must be generated in the document (e.g. Winword) which is to be displayed on the controller as a PDF.

### 2

#### Generate text file

Copy the single-column index generated above (right-justified page numbers) and save it in a TXT file.

**Example:** 10702            90  
                  <ID>            <page>

The search string (alarm number) is identical to the ID number in this case.

The text file for the PDF call now requires the following syntax:

<PDF file name> <ID> <page>

Enter the name of the PDF file at the start of each line.

**Example:** ALARM.PDF    10702   90  
                  ALARM.PDF    10703   90  
                  ALARM.PDF    10704   91  
                  <PDF name>   <ID> <page>

### 3

#### Generate PDF file

Print your document as a postscript file and convert this to PDF format using Adobe Acrobat Distiller.

### 4

#### Read files into controller

The generated files (\*.PDF and \*.TXT) must now be read into the controller, e.g. into directory F:\DH\USER\HLP.

### 2.2.3 User comments on help

The alarm help ([i] key in diagnostics) provides the "Note block" function for entering user-specific texts for individual alarms. These texts are stored in the directory:

F:\USER\RH

---

#### Note

Alarm-specific texts can be stored if the HMI help is called up using the [i] key from the alarm overview or from the alarm log for the corresponding alarm. This is not possible when the alarm is selected from the table of contents.

---

### 2.2.4 Change "Win-Help" alarm help to "PDF help"

To change the help function from "Win-Help" to "PDF help", proceed as described in Subsection "Extension of alarm help".

1. **Define jump labels for each alarm number in the document:**  
The jumps must be identified by an index entry and a single-column index list (right-justified page numbers) must be generated in the document (e.g. Winword) which is to be displayed on the controller as a PDF.
2. **Generate text file:** Copy the single-column index generated above (right-justified page numbers) and save it in a TXT file.  
The text file for the PDF call requires the following syntax:  

```
<PDF file name> <ID> <page>
```

 The search string (alarm number) is identical to the ID number in this case.
3. Enter the name of the PDF file at the start of each line.  
**Example:**

```
ALARM.PDF 10702 90
ALARM.PDF 10703 90
ALARM.PDF 10704 91
<PDF name> <ID> <page>
```
4. **Print the file** in which the alarm texts are stored **as a postscript file** (possible from all standard platforms). The file extension \*.prn is used in Winword, \*.ps in Interleaf.
5. Convert the postscript file to "PDF" using the Adobe Acrobat Distiller software.
6. **Read the files** (\*.PDF and \*.TXT) into directory F:\USER\HLP on the controller.

---

**Note**

If additional user alarms are to be displayed in the alarm help, these must be inserted in the directory F:\USER\HLP in the form of a PDF file and a TXT file of the same name.

Please adhere to the naming conventions listed in the next chapter under the section "File names".

---

## 2.3 Machine data help

### 2.3.1 Function

The machine data help is based on the "PDF help", i.e. when help is called up, a PDF file is opened and the relevant alarm is displayed on the controller in Adobe Reader.

The files containing the machine data are stored in the directories below:

- Siemens MD F:\MMC2\HLP
- User MD F:\USER\HLP

### File names

The file names below must be adhered to (must not be changed), even for the purposes of additional user MD:

- |                         |                        |
|-------------------------|------------------------|
| • N_NCK_GR.PDF [German] | N_NCK_UK.PDF [English] |
| • C_NCK_GR.PDF          | C_NCK_UK.PDF           |
| • A_NCK_GR.PDF          | A_NCK_UK.PDF           |
| • S_NCK_GR.PDF          | S_NCK_UK.PDF           |
| • DISPL_GR.PDF          | DISPL_UK.PDF           |
| • DROT_GR.PDF           | DROT_UK.PDF            |
| • DLIN_GR.PDF           | DLIN_UK.PDF            |
| • HYDRM_GR.PDF          | HYDRM_UK.PDF           |

The names of the associated TXT files (contain the jump list) must therefore be as follows:

- |                         |                        |
|-------------------------|------------------------|
| • N_NCK_GR.TXT [German] | N_NCK_UK.TXT [English] |
| • C_NCK_GR.TXT          | C_NCK_UK.TXT           |
| • A_NCK_GR.TXT          | A_NCK_UK.TXT           |
| • S_NCK_GR.TXT          | S_NCK_UK.TXT           |
| • DISPL_GR.TXT          | DISPL_UK.TXT           |
| • DROT_GR.TXT           | DROT_UK.TXT            |
| • DLIN_GR.TXT           | DLIN_UK.TXT            |
| • HYDRM_GR.TXT          | HYDRM_UK.TXT           |

## Structure of text file

PDF file without language code	MD number	Page
DROT.PDF	1000	1
DROT.PDF	1001	1
DROT.PDF	1002	1
...	...	...

## Meanings of names

Name	Meaning
N_NCK_GR.PDF	General MD
C_NCK_GR.PDF	Channel-specific MD
A_NCK_GR.PDF	Axis-specific MD
S_NCK_GR.PDF	Setting data
DISPL_GR.PDF	Display MD
DROT_GR.PDF	Drive MD, rotary
DLIN_GR.PDF	Drive MD, linear
HYDRM_GR.PDF	Drive MD, hydraulic module

### 2.3.2 Extension of machine data help

If additional user machine data are to be displayed in the MD help, these must be inserted in the directory C:\USER\HLP in the form of a PDF file and a TXT file of the same name.

#### Necessary files

- \*.PDF e.g. N\_NCK\_GR.PDF
- \*.TXT e.g. N\_NCK\_GR.TXT

#### 1

#### Define jump labels in document

The jumps must be identified by an index entry and a single-column index list (right-justified page numbers) must be generated in the document (e.g. Winword) which is to be displayed on the controller as a PDF.

#### 2

#### Generate text file

Copy the single-column index generated above (right-justified page numbers) and save it in a TXT file.

**Example:** 10000 170  
<ID> <page>

The search string (MD number) is identical to the ID number in this case.  
The text file for the PDF call now requires the following syntax:

*<PDF file name> <search string> <page>*

Enter the name of the PDF file at the start of each line.

**Example:** N\_NCK.PDF 10000 170  
N\_NCK.PDF 10002 170  
N\_NCK.PDF 10010 171  
*<PDF name> <ID> <page>*

### 3

#### Generate PDF file

Print your document as a postscript file and convert this to PDF format using Adobe Acrobat Distiller.

### 4

#### Read files into controller

The generated files (\*.PDF and \*.TXT) must now be read into directory F:\USER\HLP on the controller.





# I Index

## I.1 Subject index

### A

Add alarm text variables ..... HE1/1-19  
 Alarm help ..... HE1/2-31  
   Comments ..... HE1/2-33  
   Extend ..... HE1/2-32  
   File name ..... HE1/2-31  
 Alarm texts  
   Language abbreviations ..... HE1/2-31

### C

CONTEXT ..... HE1/1-11

### D

Description  
   Text property ..... HE1/1-18

### E

Extended PDF help ..... HE1/1-19  
   Parameterize ..... HE1/1-16

### H

Help file  
   Storage location ..... HE1/1-6  
 Help function ..... HE1/1-9  
   Extend ..... HE1/1-9  
 Help references  
   Several ..... HE1/1-18

### I

IF.INI ..... HE1/1-12  
 IFM\_GR.COM ..... HE1/1-11  
 Initialization file, file IF.INI ..... HE1/1-9  
 Instruction  
   Definition, sections ..... HE1/1-15  
   Extend ..... HE1/1-15  
   Text property ..... HE1/1-18  
 ISO mode help ..... HE1/1-22  
   Extend ..... HE1/1-23

### J

Jump file  
   Create alarm/MD help ..... HE1/2-29  
   Create programming help ..... HE1/1-7  
   Test ..... HE1/1-8, 2-30

### M

Machine data help  
   Extend ..... HE1/2-35  
   File name ..... HE1/2-34  
 Maschine data help ..... HE1/2-34

### P

Programming help  
   Extend ..... HE1/1-20

### Q

Quick ASCII help ..... HE1/1-9

**S**

Search sequence

Help ..... HE1/1-8, 2-30

**T**

Technology ..... HE1/1-16

TECHNOLOGY ..... HE1/1-9

Text files

Extend..... HE1/1-12

TEXTFILES..... HE1/1-10

TEXTSEARCH..... HE1/1-11

Topic

Definition..... HE1/1-13

Extend..... HE1/1-14

Topic ID..... HE1/1-14

# **SINUMERIK 840D sl**

## **Commissioning CNC Part 2 (HMI)**

### **Installing HMI Embedded (IM2)**

Valid for

Controller  
SINUMERIK 840D sl

Software            Software version  
HMI Embedded    7.1



<b>1 General Preparations</b> .....	<b>IM2/1-7</b>
1.1 Hardware/software prerequisites .....	IM2/1-8
1.1.1 Hardware .....	IM2/1-8
1.1.2 Software.....	IM2/1-9
1.2 Accessories.....	IM2/1-9
1.3 Documentation.....	IM2/1-10
1.4 EMC/ESD measures.....	IM2/1-10
1.4.1 Electromagnetic Compatibility (EMC) .....	IM2/1-10
1.4.2 Electrostatically Endangered Modules (ESD) .....	IM2/1-11
<b>2 Power On and Ramp-Up</b> .....	<b>IM2/2-13</b>
2.1 Start-up.....	IM2/2-14
<b>3 Communication/Initial Requirements</b> .....	<b>IM2/3-15</b>
3.1 System-internal communication .....	IM2/3-16
3.2 Licensing.....	IM2/3-16
3.3 Network operation with HMI-Embedded .....	IM2/3-18
3.3.1 Parameterizing the drive connection from the HMI user interface.....	IM2/3-18
3.3.2 Network configuration .....	IM2/3-21
3.3.3 Examples: network configuration .....	IM2/3-22
3.3.4 Configuring a drive connection via MD .....	IM2/3-24
3.3.5 Parameterizing the drive connection using LOGDRIVE.INI .....	IM2/3-25
3.3.6 Accepting network data.....	IM2/3-28
3.4 "Blue Screen" error status screen .....	IM2/3-28
3.4.1 Restart .....	IM2/3-29
<b>4 Functions/Parameterization</b> .....	<b>IM2/4-31</b>
4.1 Access protection using password and key switch .....	IM2/4-33
4.1.1 Password .....	IM2/4-34
4.1.2 Keyswitch.....	IM2/4-35
4.2 Functions .....	IM2/4-37
4.2.1 Configuring Caps Lock.....	IM2/4-37
4.2.2 Activating the screen saver .....	IM2/4-37
4.2.3 Changing the language.....	IM2/4-38
4.2.4 Activating/deactivating calculator Help.....	IM2/4-38
4.2.5 Setting date/time .....	IM2/4-38
4.2.6 Setting the editor.....	IM2/4-39
4.2.7 Geometry processor .....	IM2/4-40
4.2.8 File manager .....	IM2/4-40
4.2.9 Displaying the acknowledgement icon for PLC alarm .....	IM2/4-41
4.2.10 Alarm display .....	IM2/4-41
4.2.11 Inch/metric switchover.....	IM2/4-41
4.3 Switching channels .....	IM2/4-42
4.3.1 Several channels on 1 NCU.....	IM2/4-42

4.3.2 Configuration example of a NETNAMES.INI.....	IM2/4-43
4.4 Selecting the tool management.....	IM2/4-47
4.5 Tool fine correction .....	IM2/4-48
4.6 Activating tool correction immediately .....	IM2/4-48
4.7 Selecting tools via a D number (flat D No.) .....	IM2/4-48
4.8 Workpiece template .....	IM2/4-49
4.9 Analog spindles.....	IM2/4-49
4.10 Work offset.....	IM2/4-50
4.10.1 Changing access protection to base offset Softkey .....	IM2/4-50
4.10.2 Fine work offset and base offset .....	IM2/4-50
4.10.3 Presetting actual value memory, preset, Scratching .....	IM2/4-51
4.10.4 Hiding display of the machine axes.....	IM2/4-53
4.10.5 Displaying the geometry axes .....	IM2/4-54
4.11 Specifying the machine and rotary axis position .....	IM2/4-54
4.12 Program simulation .....	IM2/4-55
4.12.1 Rotary simulation and program test .....	IM2/4-55
4.12.2 Turning/milling simulation.....	IM2/4-56
4.12.3 Milling simulation 3D graphic .....	IM2/4-56
4.13 Measuring cycle support in the G code editor .....	IM2/4-56
4.14 Expanding user displays from the operator interface .....	IM2/4-57
4.15 Coordinate systems .....	IM2/4-58
4.15.1 Actual value display .....	IM2/4-58
4.15.2 Position of the coordinate system .....	IM2/4-58
4.16 Displaying and editing system resources .....	IM2/4-60
4.16.1 Static user memory (SRAM) .....	IM2/4-60
4.16.2 Dynamic user memory (DRAM) .....	IM2/4-64
4.17 Remote diagnosis "RCS Host Embedded / RCS Viewer Embedded" .....	IM2/4-67
4.17.1 Configuring "RCS Host Embedded" .....	IM2/4-69
4.17.2 Installing and configuring "RCS Viewer Embedded" .....	IM2/4-71
4.17.3 Start remote diagnostics .....	IM2/4-73
4.18 Available display machine data.....	IM2/4-75
<b>5 Creating In-House Alarm Texts.....</b>	<b>IM2/5-95</b>
5.1 Alarm, user, message textfiles .....	IM2/5-96
5.1.1 Storage of the text files .....	IM2/5-96
5.2 Number ranges of the text files .....	IM2/5-98
5.3 Syntax of the text file.....	IM2/5-99
5.3.1 File name .....	IM2/5-99
5.3.2 Example of an alpu.txt file.....	IM2/5-99

5.4 Supported languages.....	IM2/5-102
5.5 Bitmaps and icons.....	IM2/5-103
5.5.1 Storage of the bitmaps and icons.....	IM2/5-103
5.5.2 User status display.....	IM2/5-103
5.6 Editing in-house text files .....	IM2/5-105
5.7 Plain text for PLC machine data.....	IM2/5-107
<b>6 Series startup.....</b>	<b>IM2/6-109</b>
6.1 Series startup.....	IM2/6-110
6.1.1 Exporting data.....	IM2/6-110
6.1.2 Importing data .....	IM2/6-111
<b>7 Data Backup .....</b>	<b>IM2/7-113</b>
7.1 Data backup.....	IM2/7-114
7.1.1 Introduction .....	IM2/7-114
7.1.2 Data backup with HMI Embedded.....	IM2/7-116
7.1.3 Data backup for changed machine data.....	IM2/7-116
<b>8 Diagnostics.....</b>	<b>IM2/8-119</b>
8.1 Action log .....	IM2/8-120
8.1.1 Log file: "action.log" .....	IM2/8-122
8.1.2 Displaying the log file / setting the action log .....	IM2/8-123
8.1.3 Outputting the log file .....	IM2/8-123
8.2 Software version display .....	IM2/8-124
<b>I Index .....</b>	<b>IM2/I-125</b>
I.1 Subject Index .....	IM2/I-125





# 1

## 1 General Preparations

1.1 Hardware/software prerequisites .....	IM2/1-8
1.1.1 Hardware .....	IM2/1-8
1.1.2 Software.....	IM2/1-9
1.2 Accessories.....	IM2/1-9
1.3 Documentation.....	IM2/1-10
1.4 EMC/ESD measures.....	IM2/1-10
1.4.1 Electromagnetic Compatibility (EMC).....	IM2/1-10
1.4.2 Electrostatically Endangered Modules (ESD).....	IM2/1-11

## Introduction

This Installation and Startup Guide describes the procedures for installing the HMI Embedded software.

For further reading on special NCK, NCU, PLC, or drive functions, please consult the Descriptions of Functions/Manuals (see Section: Documentation).

## 1.1 Hardware/software prerequisites

It is a requirement that all system components be fully mounted, installed and configured.

For operating, programming, and visualization, the HMI software is integrated in the NCU software.

The HMI-Embedded software can only be operated via an operator panel with TCU.

### 1.1.1 Hardware

At least the following components are needed in order to run the HMI Embedded sl (Human Machine Interface Embedded Solution Line) software:

- Numerical Control Unit (NCU sl)
- Operator panel (OP)
- Machine control panel (MCP)
- Thin Client Unit (TCU)
- SINAMICS 120 converter system

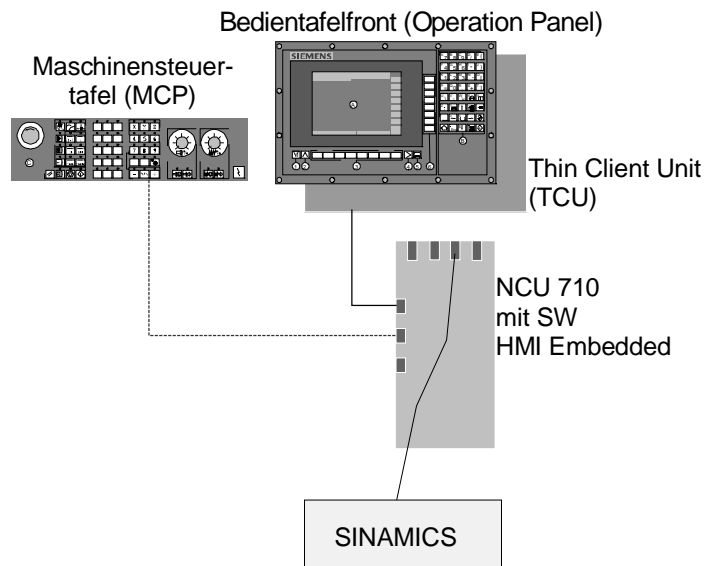


Figure 1-1 Example: Hardware configuration of an 840D sl

In this Sinumerik sl system, one or more TCUs (Thin Client Units) are used to operate and monitor the control. The actual HMI Embedded operator software runs on the NCU, while the TCUs make up the operator interface.

The TCUs are equipped with USB interfaces, to which you can connect USB devices (e.g., USB-FlashDrive, USB diskette drive, etc.) to download data onto the control or to back up the control data.

A description of the individual components and how they can be combined, connected and installed can be found in various manuals.

(see Section: Documentation).

### 1.1.2 Software

A CompactFlash card containing the complete HMI Embedded sl software package is supplied with the NCU.

#### Licensing

You need a license release (authorization to use the software) to use various options of HMI Embedded and the NCU. Activation is required before the protected software can be used.

See Chapter 3: Licensing

## 1.2 Accessories

### Memory media

#### Memory expansion of the CompactFlash card

You can use the CompactFlash card as a storage location for backing up data. To obtain additional memory expansion, you must purchase the "256 MB user memory" option with order number: 6FC5800-0AP12-0YB0.

This option is then released via the license manager (addit. 256 MB HMI user memory on CF card of NC,...).

#### USB FlashDrive

You can use a 512 MByte USB FlashDrive as an additional memory medium:  
Order number: 6ES57648-0DC20-0AA0

### Keyboard

With HMI-Embedded sl you have to distinguish between upper and lower case. As it is not possible to switch to lower case mode on the keyboard of the machine control panel /OP, you require an external keyboard (with USB connector) for installation and startup.

Plug in the keyboard in the USB interface on the front of the operator panel. The system immediately detects the keyboard.

See also Chapter 4: Configuring Caps Lock

## 1.3 Documentation

You require the following documentation to install and start up the SINUMERIK 840D sl:

### References:

/GDS/ Sinumerik 840D sl, NCU Manual  
/BHsl/ Operator Component Manual  
/IDsl/ Installation and Startup Manual CNC Part 1 (NCK, PLC, Drive)  
/IAM 2/IM5/ TCU Base Software, HT 8  
/FB1/ Description of Functions – Basic Machine  
/LIS1/ Lists  
/DA/Diagnostics Manual  
/BEMsl/ Operator's Guide HMI Embedded  
/FBTsl/ Description of Functions - ShopTurn  
/FBSPsl/ Description of Functions - ShopMill

## 1.4 EMC/ESD measures

### 1.4.1 Electromagnetic Compatibility (EMC)

#### RI suppression measures

##### Shielded signal cables

- To ensure safe, interference-free operation of the system, the cables specified in the individual drawings must be used.
- Both ends of the shield must always be connected conductively with the housings.

##### Exception:

If non-Siemens devices are connected (printers, programming devices, etc.), you can also use standard shielding cables, which are connected at one end. These external devices may not be connected to the control during normal operation. However, if the system cannot be operated without them, then the cable shields must be connected at both ends. Furthermore, the external device must be connected to the control via an equipotential bonding cable.

##### Rules for routing cables

In order to achieve the best-possible noise immunity for the complete system (control, power section, machine) the following EMC measures must be observed:

- Signal cables and load cables must be routed at the greatest possible distance from one another.
- Only cables supplied by SIEMENS may be used as signal cables to and from the NCU and PLC.
- Signal cables must not be laid in the close vicinity of strong magnetic fields (e.g., motors and transformers).
- Pulse-carrying HC/HV cables must always be laid completely separately from all other cables.
- The distance (interference injection area) between the following cables must be as close as possible:
  - Signal cable and signal cable
  - Signal cable and associated equipotential bonding lead
  - Equipotential bonding lead and protective conductor incorporated in lead.

**Important**

For further information about suppression measures and the connection of shielded cables see

**References:** /EMC/EMC Directive

---

## 1.4.2 Electrostatically Endangered Modules (ESD)

---

**Caution**

The modules contain electrostatically sensitive devices. Discharge yourself of electrostatic energy before touching the components. The easiest way to do this is to touch a conductive, grounded object immediately beforehand (for example, the bare metal part of a control cabinet or the protective ground contact of a socket outlet).

---

**Important**

Handling of modules containing devices sensitive to electrostatic discharge:

- When handling electrostatically sensitive devices, make sure that operator, workplace and packing material are properly grounded.
  - Generally, electronic modules must not be touched unless work has to be carried out on them. When handling PC boards make absolutely sure that you do not touch component pins or printed conductors.
  - Components must only be touched if
    - You are permanently grounded with an ESD arm band
    - You are wearing ESD shoes or ESD shoe grounding strips that are in contact with an ESD floor.
  - Modules must be placed on a conductive surface (table with ESD surface, conductive ESD foam plastic, ESD packing bag, ESD transport container).
  - Never place modules in the vicinity of display units, monitors, or television sets (minimum distance < 10 cm).
  - Modules must never be brought into contact with statically chargeable and highly insulating materials, such as plastic sheeting, insulating table tops, clothing made of synthetic fibers.
  - Measurements of the module must only be performed if
    - The measuring instrument is grounded (e.g., via a protective conductor) or
    - Before measuring with a potential-free measuring device, the measuring head is discharged briefly (e.g., touch unpainted metal on the control housing).
- 





# 2

## 2 Power On and Ramp-Up

2.1 Start-up ..... IM2/2-14

## 2.1 Start-up

---

### Note

Before a cold restart all USB devices must be removed from the TCU and not connected again until start-up is complete.

---

Insert the CompactFlash Card containing the HMI Embedded software into the slot marked "CF" (interface X109) before switching on the NCU.  
Now switch on the power supply. The NCU starts up automatically and automatically detects the CompactFlash card.

An 8-digit HMI Embedded version number is displayed during start-up. The seconds-counter display can be used to monitor when the connection to the NCK/PLC has been established.

Further information about the start-up is contained in the submanual of the CNC Installation and Startup Manual Part 5 (Base Software):

**References:** IAM2/IM5: Installation and Startup TCU, Section: Operation during start-up.





# 3

## 3 Communication/Initial Requirements

3.1 System-internal communication.....	IM2/3-16
3.2 Licensing.....	IM2/3-16
3.3 Network operation with HMI-Embedded.....	IM2/3-18
3.3.1 Parameterizing the drive connection from the HMI user interface.....	IM2/3-18
3.3.2 Network configuration.....	IM2/3-21
3.3.3 Examples: network configuration.....	IM2/3-22
3.3.4 Configuring a drive connection via MD.....	IM2/3-24
3.3.5 Parameterizing the drive connection using LOGDRIVE.INI ..	IM2/3-25
3.3.6 Accepting network data .....	IM2/3-28
3.4 "Blue Screen" error status screen.....	IM2/3-28
3.4.1 Restart .....	IM2/3-29

## 3.1 System-internal communication

Internal system communication in the NCU between the HMI Embedded software, NCK, drive, and PLC is carried out via an integrated PROFIBUS DP (VDP).

For a description, please refer to the following documentation:

**References:** /IDsl/ Installation and Startup Manual CNC Part 1 (NCK, PLC, Drive)

## 3.2 Licensing

A suitable license is needed in order to use the options.

In the "Startup" operating area, HMI Embedded has three menus, which contain information about the status of the options and their licenses.

### Requirements

You need at least "write access" authorization in order to set or reset options.

### Procedure

To call up the following three menus in the "Startup" operating area, select the ETC key ">" and press the "License" softkey:

- Overview
- All options
- Missing licenses

### Licensing: Overview

The following information appears when you press the "Overview" softkey:

Hardware serial number:            Shows the unique number of the Sinumerik sl

Type of hardware, e.g.:            "Sinumerik 840D sl"

Status message of the license key, e.g., "License key is NOT sufficient!"

When you obtain a new license key, it must be entered after the prompt "Please enter a license key:".

Hyphens are inserted automatically after every fourth character. Their only purpose is to improve legibility.

When you press the "Apply" softkey, the data are transmitted to the NCU and stored there.

Press the "NCK Reset" softkey to activate the licenses.

### Licensing: All options

When you press the "All Options" softkey, all NCU and HMI options are displayed. A description and order number are displayed for each option. Options shown red are activated, however, not yet licensed or not licensed in adequate number. You can set or reset the availability in the adjacent "Set" field. After changing the option, press the "Apply" softkey.

### Licensing: Missing licenses

When you press the "Missing Licenses" softkey, any insufficiently licensed options will be listed. You can set and reset the option in the adjacent field, as in the "All Options" menu.

---

#### Note

Use of non-licensed options.  
Before licensing options, you can also activate them temporarily without the license keys and use them for test purposes. In this case, the control displays alarm 8080 periodically, indicating that the corresponding license has not been allocated to the hardware.

---

The allocation of licenses to hardware is described in the following documentation:  
**References:** /IDsl/ Installation and Startup Manual CNC Part 1 (NCK, PLC, Drive)

### 3.3 Network operation with HMI-Embedded

The NCU hardware has its own network card and can be integrated in an existing network.

If the appropriate option is set, additional horizontal softkeys can be integrated in HMI Embedded with icons for a mapped drive or a local drive.

#### Possible connections

The number of connections between the HMI Embedded system and different servers or local drives is limited to eight.

These connections are also called "logical drives" and can be parameterized in three different ways.

1. from the HMI user interface
2. using machine data
3. using the "LOGDRIVE.INI" file

#### 3.3.1 Parameterizing the drive connection from the HMI user interface

The drive connections (local, network and USB drives) and their softkeys are set up in the settings window of the startup operating area.

Network connections can be defined or parameterized in the screen displayed below. These connections, also known as "logical drives", can refer to either a network connection or an internal drive, such as a PCMCIA memory-card interface (CompactFlash Card) and USB interfaces (including a diskette drive connected via USB).

#### Procedure

- Press the "Startup" softkey.
- Press the "Connections" softkey.
- Press the "Logical drives" softkey.

You will now get the "Network Connections" window in which you can set up the connections:

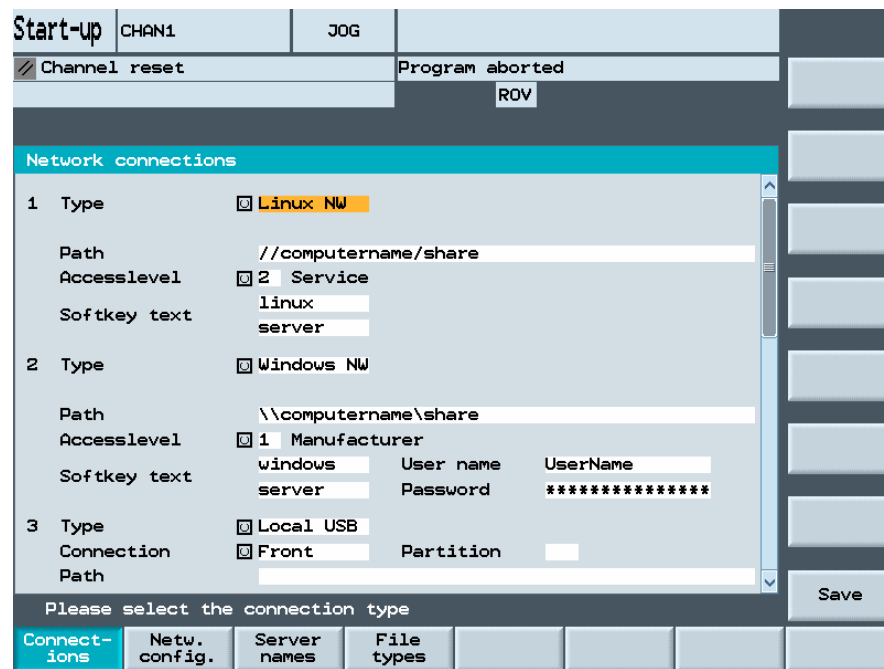


Figure 3-1: Drive connections

Use the Select key to select entries for the "Type", "Connection" and "Access level" fields.

Type: No drive, USB local, USB global  
for set option: Flash Card (delivered CompactFlash Card with HMI software)

Connection: Front (preassigned), X203, X204

Access level: 1 Manufacturer  
2 Service  
3 User  
4 Key Switch Settings 3  
5 Key Switch Settings 2  
6 Key Switch Settings 1  
7 Key Switch Settings 0

---

### Note

You need the memory expansion (optional) to access the CompactFlash card, see Chapter 1: Accessories.

---

Enter the other fields manually. You can choose any softkey text.

The following methods can be used to create the connections:

Type	Connection	Path (path entry)	
No drive	-	-	
Local USB	Front (preassigned)	acttcu://0,2	
	X203	acttcu://2	
	X204	acttcu://1	
USB global	X203 X204	Connection to other TCUs	
Flash Card		.../user/sinumerik/data/...	CompactFlash card in the NCU
NW Linux	-	// Linux ComputerName/Linux ShareName	
NW Windows	-	// WindowsComputerName /WindowsShareName	

- A numeric setting for DOS (Windows) is made in the "Partition" field.
- Two lines, with a maximum of nine characters each, are available for the softkey text.
- You must allocate a user name and a password for Windows shares. If no setting is entered, a default user (PCU20\_USER) is entered and a default password is used.
- No user names or passwords are needed for Linux (NFS) shares and local/global network paths (USB).
- The first four softkeys for the configured connections are displayed in the horizontal menu bar in the "Program" and "Services" operating areas, and the other four softkeys are displayed in the second horizontal menu bar, which can be opened using the ETC key ">".
- The selection of the drive can be prevented by setting the appropriate access level.

### 3.3.2 Network configuration

#### Settings

The data required for networking the components are entered in the "Network Configuration" window. Only the TCP/IP protocol is supported and can be configured within the network setup screens.

The window below appears when you press the "Network config." softkey in the "Startup" operating area.

IBN	CHAN1	Jog	
Kanal Reset		Programm abgebrochen	
		ROV	
Netzwerk-Konfiguration (X130-Netz)			
HMI Hochlauf:	mit Netz		
DHCP:	<input checked="" type="checkbox"/> Nein		
IP Adresse:	192	168	11 34
Subnet Mask:	255	255	255 0
Gateway:	192	168	11 1
Rechnername:	TEST_HMIEMBEDDED		
DNS	1:	192	168 10 4
	2:	192	168 10 5
	3:	0	0 0 0
DNS Domain:	TEST_DOMAIN		
MAC Adresse:	08 00 06 73 6F 41		
Ermittlung der IP-Adresse über Select ausv.			
Verbindungen	Netz-Konfig.	Server-Namen	Datei-Typen
			Speichern

Fig. 3-2 Configuring the network

1. DHCP:
  - Use the "Select" key to select the "Yes" or "No" setting, depending on whether or not a DHCP server is available.
2. IP addresses:
  - A unique IP address must be selected for each computer.
1. Subnet mask:
  - You need to enter a subnet mask that is unique within the entire network.
2. Gateway:
  - A gateway is not needed for a local area network.
  - Please ask your administrator for the settings of a company network.

3. Computer name:  
The name is optional, as it is only a symbolic identifier and can be replaced by the IP address at any time.
4. DNS:  
If the DNS (Domain Name Server) has the appropriate assignment of computer and IP address, no entry is required here.  
  
MAC address: is unique (present on the NCU 710 after X130). If the NCU is already known, it will be displayed automatically.

Use the "Save" softkey to save the parameter settings for the network connection.

### 3.3.3 Examples: network configuration

This section describes two possible configurations:

1. A stand-alone local area network
2. A large company network.

#### Local area network

A small stand-alone network is the easiest option.



#### Important

As no "hub" is used, you will need a special twisted-pair cable for the point-to-point connection.

---

Select for DHCP: "Yes", the IP address, subnet mask and gateway data are transmitted and displayed automatically via a DHCP server. These data cannot be overwritten.

You have to enter the computer name and DNS domain manually.

Select for DHCP: "No", because no server exists, for example, you have to determine and enter the data yourself.

For stand-alone local area networks, RFC 1597 provides for three address ranges so that there are no conflicts while using the Internet at the same time:

10.0.0.0	>	10.255.255.255	(Class A network)
172.16.0.0	>	172.31.255.255	(Class B network)
192.168.0.0	>	192.168.255.255	(Class C network)

The IP addresses for small networks should be selected from these ranges.



## Company network

Since the company network already exists, you will need to ask the administrator for the settings and addresses.

A fixed IP address is used to address the company network. You will need to ask your administrator for an IP address for each NCU.

The computer name is also provided by the network administrator and corresponds to the IP address that is set.

Further, it is necessary to have the correct subnet mask for the network and a default gateway (router).

If you intend to use a DNS server, you will also need its IP address and the DNS domain.

## Server name (setting not mandatory)

In order to assign symbolic names to the computers, the relevant nodes to be addressed within the network can be entered so that their names can be resolved:

### Example

Server name:

192.168.0.20          R2345

192.168.0.21          R2346

With these settings computers R2345 and R2346 can be addressed by means of their symbolic names.

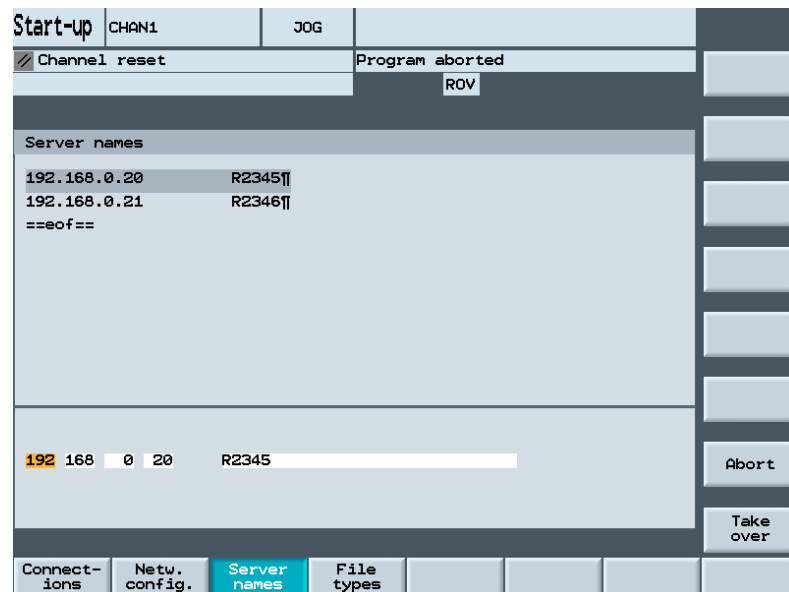


Fig. 3-3 Server names

### Operation:

The first entry in the line is the IP address.

The second entry is the computer name. The data already specified appears in the upper section of the screen.

By selecting the "New" and "Edit" softkeys you can go to the lower half of the screen where you can edit entries.

You can change back to the upper section by pressing the "Cancel" softkey (data is not saved) or the "Save" softkey (data is saved).

With external drives (network connections) you need to specify the full network path (\\Rechnername\Freigabenamen\Pfadangabe), whereby the path name is optional:

\\R4711\Workpieces\Template

\\R4812\Workpieces

These names must be available on the network computer. The network computer, for example, is called "\\R4812". The share name WORKPIECES must be specified on any hard disk drive on this computer.

## File types

The file types are set up as needed and are evaluated and displayed by the NCU. The Program Manager will use the file extensions to filter the files to be displayed and those to be hidden. This filter function can be set as follows:

If the "New" or "Edit" softkeys are selected, a switch is made to the lower half of the window where a new extension can be entered or an existing extension edited. To exit, press "Cancel" (without changes) or "Save" (with changes).

### 3.3.4 Configuring a drive connection via MD

If you do not need more than four drive connections, you can set these up using machine data.

For the authorization of the input for configuration data, the following display machine data must be set up:

MD 9509: MM\_USER\_CLASS\_DIRECTORY\_CHG (value 0 ... 7)

## Display machine data

The following display machine data is available for authorizing the use of the created drives:

For the "Program" and "Services" areas

MD 9510: MM\_USER\_CLASS\_DIRECTORY1\_P

MD 9511: MM\_USER\_CLASS\_DIRECTORY2\_P

MD 9512: MM\_USER\_CLASS\_DIRECTORY3\_P

MD 9513: MM\_USER\_CLASS\_DIRECTORY4\_P

(value 0...7)

For the "Machine" area

MD 9516: MM\_USER\_CLASS\_DIRECTORY1\_M

MD 9516: MM\_USER\_CLASS\_DIRECTORY2\_M

MD 9516: MM\_USER\_CLASS\_DIRECTORY3\_M

MD 9516: MM\_USER\_CLASS\_DIRECTORY4\_M

The network drives are configured in the display machine data:

MD 9676: MM\_DIRECTORY\_SOFTKEY\_PATH1  
MD 9676: MM\_DIRECTORY\_SOFTKEY\_PATH2  
MD 9676: MM\_DIRECTORY\_SOFTKEY\_PATH3  
MD 9676: MM\_DIRECTORY\_SOFTKEY\_PATH4

The display machine data for the access rights are effective immediately, i.e., after switching to the relevant application. A restart is not required.

### 3.3.5 Parameterizing the drive connection using LOGDRIVE.INI

You can also use LOGDRIVE.INI to set up all drives (NW Linux, USB interface, CompactFlash card) that do not require any password.

The display machine data are ignored and the entries are used from this file in accordance with the HMI search strategy (USER, OEM, ADD\_ON).

---

#### Important

The change or set up of a Windows network drive (prefix: smb) cannot be made in the LOGDRIVE.INI, because the password must be entered encrypted.

---

#### Opening LOGDRIVE.INI

You need access level 0 in order to call up the file.

#### Procedure:

To open the file, proceed as follows:

- Select the "Startup" operating area.
- Press the "HMI" softkey.
- Press the "Editor" softkey (HMI only).
- Press the "Flash Card" vertical softkey.  
You will find the file in: \user\sinumerial\hmi\cfg\logdrive.ini.
- Press the "Input" key to open the file.

#### LOGDRIVE.INI file

##### [CONNECTIONS]

Connection<Index>=<entry>

SK-Connection<Index>=<Softkeytext, 2-lines separated by %n>

Username<Index>=<Username>

Password<Index>=<password, coded, cannot be changed with the editor!>

Type <Index>=

AccessMachine<Index>=<access>

AccessProgram<Index>=<access>

AccessService<Index>=<access>

Index: 1 to 8 connections (possibly with gaps).  
If "Connection" is not set or is greater than 8, the number of entries is initially set to 8.

Access: Access rights for machine, program and services possible from **0 – 7**.  
If nothing is entered, the default is 7 (= access for all).

1st entry: Configuring of the drive in the following form.  
Path details for the following interfaces:  
FRONT: tcu:// acttcu://0,2  
X203: tcu:// acttcu://2  
X204: tcu:// acttcu://1

---

### Note

For a connection to a Linux connection, "**nfs:**" is prefixed to the computer name.

For a connection to a Windows computer, "**smb:**" is prefixed to the computer name.

---

### Example of a LOGDRIVE.INI

```
;------  
;Copyright (C) Siemens AG 2005 All Rights Reserved.  
;------  
  
; Set up drives using ini file  
  
;[CONNECTIONS]  
  
;Connection<Index>  
;Connection<index>=<entry>  
;AccessMachine<index>=<access>  
;AccessProgram<index>=<access>  
;AccessServices<index>=<access>  
  
[CONNECTIONS]  
Connection = 1  
Connection1 = "acttcu://0,2/"  
SK_Connection1 = "USB%nFront"  
Username1 = " "  
Password1 = " "  
Type1 = "0"  
AccessMachine1 = "5"  
AccessProgram1 = "5"  
AccessServices1 = "5"  
  
Connection = 2  
Connection2 = "acttcu://2/"  
SK_Connection2 = "USB2"  
Username2 = " "  
Password2 = " "  
Type2 = "0"  
AccessMachine2 = "6"  
AccessProgram2 = "6"  
AccessServices2 = "6"  
  
Connection = 3  
Connection3 = "acttcu://1/"  
SK_Connection3 = "USB3"
```

```
Username3 = " "  
Password3 = " "  
Type3 = "0"  
AccessMachine3 = "6"  
AccessProgram3 = "6"  
AccessServices3 = "6"  
  
Connection = 4  
Connection4 = ""  
SK_Connection4 = ""  
Username4 = " "  
Password4 = " "  
Type4 = ""  
AccessMachine4 = ""  
AccessProgram4 = ""  
AccessServices4 = ""  
  
Connection = 5  
Connection5 = ""  
SK_Connection5 = ""  
Username5 = " "  
Password5 = " "  
Type5 = ""  
AccessMachine5 = ""  
AccessProgram5 = ""  
AccessServices5 = ""  
  
Connection = 6  
Connection6 = ""  
SK_Connection6 = ""  
Username6 = " "  
Password6 = " "  
Type6 = ""  
AccessMachine6 = ""  
AccessProgram6 = ""  
AccessServices6 = ""  
  
Connection = 7  
Connection7 = ""  
SK_Connection7 = ""  
Username7 = " "  
Password7 = " "  
Type7 = ""  
AccessMachine7 = ""  
AccessProgram7 = ""  
AccessServices7 = ""  
  
Connection = 8  
Connection8 = ""  
SK_Connection8 = ""  
Username8 = " "  
Password8 = " "  
Type8 = ""  
AccessMachine8 = ""  
AccessProgram8 = ""  
AccessServices8 = ""  
  
[EXTENSION_FILTER]
```

---

**Note**

You must not edit the password, as it is encrypted in this file.

---

### 3.3.6 Accepting network data

You can edit all screens, connections, Netw. Config., server names and file types in succession. After completing your inputs and activating the "Save" softkey, a prompt is displayed asking whether you want to accept the data.

## 3.4 "Blue Screen" error status screen

If the system crashes, an error status screen ("Blue Screen") showing current system information is displayed.

The error status screen displays a brief description of the error, which has occurred, plus all relevant processor registers.

### Backup option

A complete system data image is transferred to the CompactFlash card (optional).

### System data display

In the event of a system crash, the crash address, the register allocation and any additional information needed for crash analysis are displayed on the screen.

```
Sorry, but HMI Embedded has caused an exception in task XXXXX
Function at XXXXXXXXX with code XXXXX
The type of exception is: XXXXXXXXXXXXXXXXXXXXXXXX
The exception has occurred at cs: XX eip : XXXXXXXX!
EAX= XXXXXXXX EBX= XXXXXXXX ECX= XXXXXXXX EDX= XXXXXXXX
EDI= XXXXXXXX ESI= XXXXXXXX FLAGS= XXXXXXXX DS= XXXX ES= XXXX
SS= XXXX ESP= XXXXXXXX EBP= XXXXXXXX FS= XXXX GS= XXXX
Additional information:
XXXXXXXXXX loaded at: XXXXXXXX XXXXXXXX loaded at: XXXXXXXX
XXXXXXXXXX loaded at: XXXXXXXX XXXXXXXX loaded at: XXXXXXXX
...
XXXXXXXXXX loaded at: XXXXXXXX XXXXXXXX loaded at: XXXXXXXX
CODE = XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX
      XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX
EIP:  XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX
      XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX
STACK= XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX
      XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX
      XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX
      XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX

Please send the above or saved information to the SINUMERIK Hotline!
email: ad.support@siemens.com
Fill in subject: "HMI-Embedded exception (SW XX.XX.XX; date)"
Press 'S' to save data or press 'R' to reboot immediately
```

When you select key:  
"S" (Save) you open the menu for saving data,  
"R" to restart the NCU.

### Menu for saving data

The selected system data file is saved to the CompactFlash card (optional) or can be exported to a USB FlashDrive (available soon).

```
Press 'L' to save exception data to the local compact flash card.  
You can also press 'R' to reboot immediately but we recommend to first  
save the data.
```

The following status message also appears:

```
Please wait - data transmission in progress.  
If you have problems you can  
press key 'S' to stop transmission.
```

### Transmission finished

Once all the data have been transferred successfully, the following message appears:

```
Transmission finished, XXXXXX bytes saved.  
  
.br/>The exception data was saved in folder /user/sinumerik/hmi  
Please send the saved information to the SINUMERIK Hotline!  
email: ad.support@siemens.com  
Fill in subject: "HMI-Embedded exception (SW XX.XX.XX; date)"  
  
Press 'S' to return to save menu or 'R' to reboot
```

### 3.4.1 Restart

The exception handler attempts an automatic restart in Restart mode.  
If the NCU has not reacted after 60 seconds, you must restart it manually by switching the power supply to the system off and then on again.

```
Try rebooting - in the event of no reaction after 1 minute,  
please switch off/on
```







# 4

## 4 Functions/Parameterization

4.1 Access protection using password and key switch.....	IM2/4-33
4.1.1 Password .....	IM2/4-34
4.1.2 Keyswitch.....	IM2/4-35
4.2 Functions .....	IM2/4-37
4.2.1 Configuring Caps Lock .....	IM2/4-37
4.2.2 Activating the screen saver.....	IM2/4-37
4.2.3 Changing the language.....	IM2/4-38
4.2.4 Activating/deactivating calculator Help .....	IM2/4-38
4.2.5 Setting date/time .....	IM2/4-38
4.2.6 Setting the editor.....	IM2/4-39
4.2.7 Geometry processor .....	IM2/4-40
4.2.8 File manager .....	IM2/4-40
4.2.9 Displaying the acknowledgement icon for PLC alarm .....	IM2/4-41
4.2.10 Alarm display .....	IM2/4-41
4.2.11 Inch/metric switchover .....	IM2/4-41
4.3 Switching channels .....	IM2/4-42
4.3.1 Several channels on 1 NCU .....	IM2/4-42
4.3.2 Configuration example of a NETNAMES.INI .....	IM2/4-43
4.4 Selecting the tool management .....	IM2/4-47
4.5 Tool fine correction .....	IM2/4-48
4.6 Activating tool correction immediately .....	IM2/4-48
4.7 Selecting tools via a D number (flat D No.) .....	IM2/4-48
4.8 Workpiece template .....	IM2/4-49
4.9 Analog spindles.....	IM2/4-49
4.10 Work offset.....	IM2/4-50
4.10.1 Changing access protection to base offset Softkey .....	IM2/4-50
4.10.2 Fine work offset and base offset.....	IM2/4-50
4.10.3 Presetting actual value memory, preset, Scratching .....	IM2/4-51
4.10.4 Hiding display of the machine axes .....	IM2/4-53
4.10.5 Displaying the geometry axes.....	IM2/4-54
4.11 Specifying the machine and rotary axis position .....	IM2/4-54
4.12 Program simulation.....	IM2/4-55

4.12.1 Rotary simulation and program test.....	IM2/4-55
4.12.2 Turning/milling simulation .....	IM2/4-56
4.12.3 Milling simulation 3D graphic .....	IM2/4-56
4.13 Measuring cycle support in the G code editor .....	IM2/4-56
4.14 Expanding user displays from the operator interface .....	IM2/4-57
4.15 Coordinate systems .....	IM2/4-58
4.15.1 Actual value display .....	IM2/4-58
4.15.2 Position of the coordinate system.....	IM2/4-58
4.16 Displaying and editing system resources .....	IM2/4-60
4.16.1 Static user memory (SRAM).....	IM2/4-60
4.16.2 Dynamic user memory (DRAM).....	IM2/4-64
4.17 Remote diagnosis "RCS Host Embedded / RCS Viewer Embedded" .....	IM2/4-67
4.17.1 Configuring "RCS Host Embedded" .....	IM2/4-69
4.17.2 Installing and configuring "RCS Viewer Embedded" .....	IM2/4-71
4.17.3 Start remote diagnostics .....	IM2/4-73
4.18 Available display machine data .....	IM2/4-75

## 4.1 Access protection using password and key switch

Access levels for enabling data areas are implemented in the SINUMERIK 840D sl. These are subdivided into:

- 4 password levels for Siemens, machine manufacturer and end user.
- 4 keyswitch positions for end user.

This provides a multilevel safety concept for controlling access rights.

### Access rights

- 0 to 3 are locked by means of a password and
- 4 to 7 by means of keyswitch positions.

Table 4-1 Protection level concept

Protection level	Locked by	Area
0	Password	Siemens
1	Password: SUNRISE (default)	Machine manufacturer: Development
2	Password: EVENING (default)	Machine manufacturer: Startup engineer
3	Password: CUSTOMER (default)	End user: Service
4	Keylock switch position 3	End user: Programmer, machine setter
5	Keylock switch position 2	End user: Skilled operator without programming knowledge
6	Keylock switch position 1	End user: Trained operator without programming knowledge
7	Keylock switch position 0	End user: Semi-skilled operator

### Access features

- Protection level 0 provides the greatest number of access rights, protection level 7 the least.
- If certain access rights are granted to a protection level, these protection rights automatically apply to any higher protection levels.
- Conversely, protection rights for a certain protection level can only be altered from a higher protection level.
- Access rights on protection levels 0 to 3 are set as Siemens defaults.
- Access rights can be set by querying the current keyswitch positions and comparing the passwords entered. When a password is entered it overwrites the access rights of the keyswitch position.

- Options can be protected on each protection level. However, option data can only be entered in protection levels 0 and 1.
- Access rights for protection levels 4 to 7 are only suggestions and can be altered by the machine tool manufacturer or end user.

### 4.1.1 Password

#### Setting/resetting password

The passwords for the 4 available password levels with their respective access authorization can be entered in the "Diagnostics" operating area through selection of the "Set password" softkey.

**References:** /BEM/Operator's Guide HMI Embedded

A password remains valid until the access authorization is reset with the "Delete password" softkey.

Access authorization is therefore not automatically deleted by Power On!

#### Permitted characters

Up to eight characters can be used in the password. We recommend that you restrict yourself to the character set of the operator panel in selecting a password. Where a password consists of less than eight characters, the additional characters are interpreted as blanks.

#### Default passwords

The following default passwords are defined for protection levels 1 to 3:

Protection level 1 SUNRISE  
Protection level 2 EVENING  
Protection level 3 CUSTOMER

---

#### Note

The passwords are entered as the defaults on system ramp-up in startup mode (NCK startup switch in position 1).

The default passwords should be changed in order to guarantee reliable access protection.

---

### 4.1.2 Keyswitch

Protection levels 4 to 7 require a corresponding keyswitch position on the machine control panel. Three keys of different colors are provided for this purpose. Each of these keys provides access to certain areas only. The keyswitch positions are transferred at the PLC interfaces. The associated interface signals are in the DB10, DBB56 and can be evaluated by the PLC user program.

#### Keyswitch application

The keyswitch can be used to inhibit access to certain data areas. In this way, for example, unintentional changes by the operator in the geometry data (e.g., work offsets) can be excluded.

Keyswitch position 0 has access rights of the lowest priority and position 3 access rights of the highest priority.

Table 4-2 Meaning of the keyswitch positions

Key color	Switch position	DB10, DBB56	Protection level
All (no key used)	0 = Key removal position	Bit 4	7
Black	0 and 1	Bit 5	6-7
green	0 to 2	Bit 6	5-7
Red	0 to 3	Bit 7	4-7

#### Influencing the PLC user program

The PLC interface signals "Keyswitch positions 0 to 3" can be input either directly via the keyswitch on the machine control panel or from the PLC user program. If more than one interface signal is set at the same time, the control sets keyswitch position 3.

#### Changing the protection levels for machine and setting data

The user can change the priority of the protection levels. Only protection levels of lower priority can be assigned to the machine data, setting data can also be assigned protection levels of higher priority.

The file becomes active when the next `_N_INITIAL_INI` is read in. Different protection levels are specified for writing (changing) or reading (parts program or PLC).

##### Example:

```
%_N_SGUD_DEF ; File for global variables
; $PATH=/_N_DEF_DIR
REDEF $MA_CTRLOUT_SEGMENT_NR APR 7 APW 4 ; (APR ... read access)
REDEF $MA_ENC_SEGMENT_NR APR 2 APW 2 ; (APW ... write access)
REDEF $SN_JOG_CONT_MODE_LEVELTRIGGRD APR 2 APW 2
M30
```

The MD 10000 general machine data has protection level 2 / 7, i.e. protection level 2 (corresponds to password) and protection level 7 are required to write or read, respectively. In order to access the machine data area, key switch position 3 or more is required.

## Undoing protection level changes

To undo protection level changes, the original values have to be restored.

### Example:

```
%_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
```

For a detailed description of how to change protection levels or the REDEF command for assigning protection levels, please refer to:

**References:** /PGA/Programming Guide Advanced  
Chapter: "Protection levels for NC commands"

## 4.2 Functions

### 4.2.1 Configuring Caps Lock

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

Via the display machine data MD 9009: KEYBOARD\_STATE sets the key behavior for Caps Lock:

- 0: CAPSLOCK off
- 1: Setting not relevant!
- 2: CAPSLOCK on (default setting)

If the display machine data is set to "CAPSLOCK" you can switch between upper and lower case with key combination "Ctrl" and the Shift key.

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

### 4.2.2 Activating the screen saver

If a screen with high contrast is displayed unchanged for longer than an hour, the screen must be switched dark (screensaver) in order to protect the TFT display against so-called "freezing" of the last displayed screen.

#### Activating via PLC

You can activate the screen darkening with the PLC in DB19.DBB0. The PLC has priority over the entry in the display machine data.

**References:** /FB1/A2: Various Interface Signals

#### Activating via the machine data

Via the display machine data MD 9006: DISPLAY\_BLACK\_TIME is used to specify the duration (max. 60 minutes), after which the screen is automatically darkened if no key is pressed on the keyboard.

In this display machine data, the "Screen dark from HMI Embedded" function can be started with the following entry:

- MD 9006 = 1 < time in minutes, e.g., 1 >
- MD 9006 = 60 < time in minutes, e.g., 60 >
- MD 9006 = 0 Deactivation of the function (default setting)

### 4.2.3 Changing the language

The HMI software is available in 6 languages as standard:

You can access language selection via the "Language Selection" softkey in the "Startup" operating area.

Select the first and second languages from the "Language Selection" menu.

During operation, the switchover between the set languages is performed via the "Change Language" softkey in the "Startup" operating area.

The language displayed during the control ramp-up is set via the MD 9003 display machine data. FIRST\_LANGUAGE

### 4.2.4 Activating/deactivating calculator Help

Via the display machine data MD 9991: HMI\_HELP\_SYSTEMS can be used to activate/deactivate the calculator help.

Bit 0 = 0 help display not active

Bit 0 = 1 help display active (default setting)

### 4.2.5 Setting date/time

You can set the date and time in the PLC and synchronize the PLC and HMI Embedded date and time.

This function is accessed via the Startup operating area in the PLC menu by activating the "Set time/date" softkey.

If any changes are made to the time and date, the time and date from HMI Embedded are transferred to the PLC.

The synchronization can be verified in the "Current time:" output field.

The cyclical synchronization of the time can be

- activated/deactivated via the "Synchronous AUTO/ON" or "Synchronous AUTO/OFF" softkeys and
- the duration of the synchronization set via the "Synchronous manual" softkey. 10 minutes should be entered as default.  
Minimum value: 0, maximum value: 99.

The set values are retained when the control is rebooted.

---

#### Note

Changes can only be made to the PLC operands with the appropriate password.

---



## 4.2.6 Setting the editor

The editor is available in the "Program" and "Startup" operating areas.  
You can set the editor settings as follows:

1. Via the display machine data MD 9460: PROGRAM\_SETTINGS
2. Via the HMI user interface

### Settings in the display machine data

The following section describes the settings as used in HMI Embedded:

Bit 2	= 0 = 1	No automatic enabling for programs Automatic enabling for programs created using the "New" softkey.
Bit 4	= 0 = 1	Line feed will be displayed in an opened file in the edit field. The LF (line feed) icon will be hidden in an opened file in the edit field.
Bit 5	= 0 = 1	Hidden lines marked with "**HD" (hidden) will be displayed. Hidden lines marked with "**HD" (hidden) remain hidden.
Bit 6	= 0 = 1	Active program disabled for editing. Active program enabled for editing.
Bit 7	= 0 = 1	Perform line test Disable the line test of the cycle support for programs with a file size in the MD 9464 display machine data greater than: MAX_PROGRAMM_SIZE_CHECK. The size is entered in KBytes. New lines can also be added within the G-Code created by the "Wizard" or the geometry processor.

The settings in the Program/Startup operating areas are retained after a reset.

### Settings via the HMI user interface

From the HMI user interface, you can make the same settings in the "Editor settings" window pane as for the display machine data.

Open a program for editing in the "Program" operating area.  
To obtain the settings window, press the "Settings" horizontal softkey and the "Editor Settings" vertical softkey.

You can mark the following functions in the "Yes" field at the side.

- Release new programs
- Hide LF in the program
- Display hidden lines in the program
- Enable active programs for editing.
- Special handling for mold making      File size in KB

## 4.2.7 Geometry processor

You have two possibilities of specifying the text output at the end of a contour and to set the representation of the contour elements:

1. Via the HMI user interface
2. Via the display machine data MD 9460: PROGRAM\_SETTINGS

### Settings from the HMI user interface

Press the "Contour settings" vertical softkey.

The window pane: "Free contour programming settings" will be opened:

- Enter after "Text output at the end of the contour programming" the text that is to appear after each contour, e.g. "Contour end"

You can select/deselect other settings:

- Technology specifications: "Rotate"
- Show graphically softkey for geometry elements

### Settings in the display machine data

The following section describes the settings in the display machine data as used in HMI Embedded:

Bit 0	= 1	Technology rotating (contour support)
Bit 3	= 0	The softkeys for selecting the contour elements are assigned text.
	= 1	The softkeys for selecting the contour elements are assigned icons.

## 4.2.8 File manager

You can use the file manager to select a connection as standard directory for all created drives.

You can specify the selection of this standard directory using a display machine data or via the HMI user interface:

1. Via the display machine data MD 9005: PRG\_DEFAULT\_DIR:  
Use the Select key to select all created drive connections (1 to max. 8).
2. From the HMI user interface:  
Click the "Settings" vertical softkey and the "Program overview" vertical softkey. All created drives will be shown. Select the connection you want to use as the standard directory and then press the "OK" softkey to save the directory.

### 4.2.9 Displaying the acknowledgement icon for PLC alarm

Via the display machine data MD 9055: PLC\_ALARM\_PICTURE, activate or deactivate the following icons:

Value: -1 No icon display  
Value: 0 Icon labeled "PLC" is displayed.  
Value: 1 "Cancel" icon is displayed.

This functionality refers only to PLC alarms, not to PLC messages.

### 4.2.10 Alarm display

If several alarms are pending at the same time, only the last alarm is displayed in the alarm and message line. If you want to display the pending NCK, PLC or HMI alarms alternatively, you can set this in the display machine data MD 9056: ALARM\_ROTATION\_CYCLE.

The following settings can be made:

< 500: No change of the alarm display, only the last alarm is displayed  
500 – 10000: Display duration of an alarm in milliseconds

If a valid display duration is set, each alarm is displayed for this time before being replaced by the next alarm.

Messages will not be displayed alternatively.

### 4.2.11 Inch/metric switchover

The availability of the softkey, and thus the associated functionality, can be configured using the general NCK machine data MD 10260. In the "Machine" operating area, you can switch between the inches measuring system and the metric measuring system via a softkey. If you want to display the "Switch to inch" or "Switch to mm" softkey, set the machine data as follows:

MD 10260: CONVERT\_SCALING\_SYSTEM ≠ 0

**References:** /FB1/A3: Axis Monitoring, Protection Zones  
/FB1/G2: Velocities, setpoint / actual value systems, closed-loop control

## 4.3 Switching channels

### 4.3.1 Several channels on 1 NCU

#### Function

You can establish a connection between HMI Embedded and the PLC in every operating area, using the user interface.

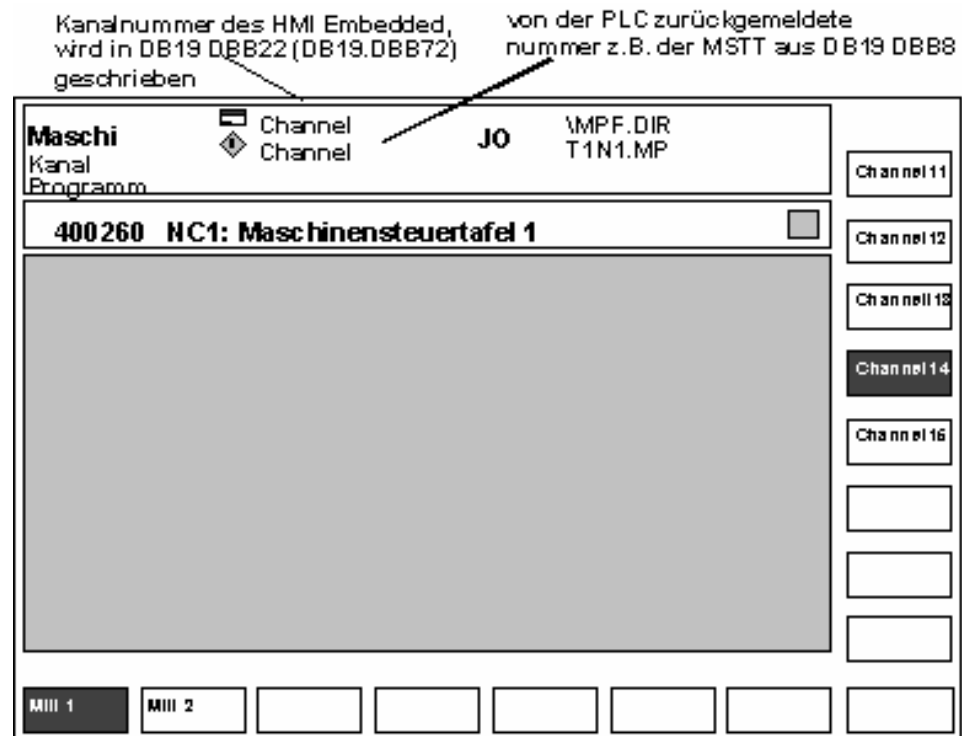


Fig. 4-1 Channel menu

Only the channels of the respective group are displayed.  
Activate the channel switchover key. The currently existing connection is displayed by means of the highlighted softkeys (horizontal, vertical) if the channel menu is active.

#### Channel switchover

You can change to other channels by means of the vertically arranged softkeys.

## Group switchover

Use the horizontal softkeys to switchover to a different group; the channels of the current group are now displayed on the vertical softkeys. The switchover to another channel takes place only when a vertical softkey is activated.

---

### Note

The softkeys only offer the connections that are really assigned and whose channels are active in the respective NCK.

---

## 4.3.2 Configuration example of a NETNAMES.INI

The channel menu is configured by extending the "NETNAMES.INI" file. The file is located on the CompactFlash Card in the following path:  
user/sinumerik/hmi/cfg.

### Procedure

Proceed as follows to modify the file for your application:

- Press the "Startup" softkey to enter the Startup operating area.
- Press the "HMI" softkey and
- the "Editor softkey (only HMI)".
- Now press the "Edit.net-names.ini" vertical softkey.  
A window pane with the following questions opens:  
"Do you want to edit your changes that have not yet been activated?"  
"Do you want to edit the current data?"
- Press the "Inactive file" vertical softkey. This opens an ini file where you can make your settings.
- Press the "Close editor" softkey. Your entries are checked and the "Test completed" window pane is displayed together with the number of errors and messages.
- Press the "Data active" softkey, this initiates a reset and configuring is completed.

### NETNAMES.INI file

1. Define groups (a max. of 8) in "logChanSetList"
2. Define the NCKs and the channels with unambiguous names for the groups in "logChanList"
3. For the channels (max. 8/group) specify
  - The NC name in "logNCName",
  - The channel number on the NCK in "ChanNum" and
  - Reserve softkey yes/no (via the command IsAGap" = True/False )

## 1. Example: Operator panel front as main operator panel

```

; NETNAMES.INI                               Version 04.03.11           30.01.05

[own]
owner ==      MMC_1

; Description of possible connections

[conn MMC_1]
conn_1=      NCU_1

; Description of significant net parameters

[param network]
bus=         mpi

[param MMC_1]
mmc_address = 1
              ;only the first letter N or H will be
              evaluated
name=VNC ;NCU -> connection using IP to 710s, ip=must
              be specified
              ;HT8 -> connection using SPC2 to PCU with Linux,
              ip will be ignored
              ;VNC -> or no name, connection to 710s with
              internal HMI, ip will be ignored

[param NCU_1]
nck_address= 3 ,ip=127.0.0.1
plc_address= 2 ,ip=127.0.0.1
name          = NCU710

; Description of a standard channel configuration

[chan MMC_1]
ShowChanMenu          = TRUE
DEFAULT_logChanSet    = ChannelArea1
DEFAULT_logChan       = Channel111
logChanSetList        = ChannelArea1

[ChannelArea1]
logChanList           = Channel111,Channel112,Channel113

[Channel111]
logNCName             = NCU_1
ChanNum               = 1

[Channel112]
logNCName             = NCU_1
ChanNum               = 2

[Channel113]
logNCName             = NCU_1
ChanNum               = 3

```

## Foreign language texts

In particular for the channel menu, the "chan.txt" file is available in several language variants that are stored language-dependent in the text directories e.g. for German text at: card/user/sinumerik/hmi/lng/deu", for English text at: card/user/sinumerik/hmi/lng/eng, etc. The content of "chan.txt" files, which do not yet contain any texts, is as follows.

### Example:

```
// CP = 1252
// IDEO = NO
// ANSI = YES
// UNICODE = NO
// VERSION = $VERSION

[SWITCHES]/NATIVE
File with OEM name specifications for channel-areas and
channels

/* Lengths of softkey text 2 * 9 characters */
/* For next line use %n in the string */
/* You must not change any number but only insert the texts!
*/
/* Text-definitions for softkeys in channel-switch-menu */

/* Name of channel area 1 and names of channels of channel
area 1 */
3080193 "NCU_1" //2x9 T_CHAN_AREA_1
3080194 "CHANNEL_1" //2x9 T_CHAN_AREA_1_CHANNEL_1
3080195 "CHANNEL_2" //2x9 T_CHAN_AREA_1_CHANNEL_2
3080196 "CHANNEL_3" //2x9 T_CHAN_AREA_1_CHANNEL_3
3080197 " " //2x9 T_CHAN_AREA_1_CHANNEL_4
3080198 " " //2x9 T_CHAN_AREA_1_CHANNEL_5
3080199 " " //2x9 T_CHAN_AREA_1_CHANNEL_6
3080200 " " //2x9 T_CHAN_AREA_1_CHANNEL_7
3080201 " " //2x9 T_CHAN_AREA_1_CHANNEL_8

/* Name of channel area 2 and names of channels of channel
area 2 */
3080202 " " //2x9 T_CHAN_AREA_2
3080203 " " //2x9 T_CHAN_AREA_2_CHANNEL_1
3080204 " " //2x9 T_CHAN_AREA_2_CHANNEL_2
3080205 " " //2x9 T_CHAN_AREA_1_CHANNEL_3
3080206 " " //2x9 T_CHAN_AREA_2_CHANNEL_4
3080207 " " //2x9 T_CHAN_AREA_2_CHANNEL_6
3080208 " " //2x9 T_CHAN_AREA_2_CHANNEL_7
3080209 " " //2x9 T_CHAN_AREA_2_CHANNEL_8
```

```

/* Name of channel area 3 and names of channels of channel
area 3 */
3080210 "" //2x9 T_CHAN_AREA_3
3080211 "" //2x9 T_CHAN_AREA_3_CHANNEL_1
3080212 "" //2x9 T_CHAN_AREA_3_CHANNEL_2
3080213 "" //2x9 T_CHAN_AREA_3_CHANNEL_3
3080214 "" //2x9 T_CHAN_AREA_3_CHANNEL_4
3080215 "" //2x9 T_CHAN_AREA_3_CHANNEL_5
3080216 "" //2x9 T_CHAN_AREA_3_CHANNEL_6
3080217 "" //2x9 T_CHAN_AREA_3_CHANNEL_7
3080218 "" //2x9 T_CHAN_AREA_3_CHANNEL_8

/* Name of channel area 4 and names of channels of channel
area 4 */
3080219 "" //2x9 T_CHAN_AREA_4
3080220 "" //2x9 T_CHAN_AREA_4_CHANNEL_1
3080221 "" //2x9 T_CHAN_AREA_4_CHANNEL_2
3080222 "" //2x9 T_CHAN_AREA_4_CHANNEL_3
3080223 "" //2x9 T_CHAN_AREA_4_CHANNEL_4
3080224 "" //2x9 T_CHAN_AREA_4_CHANNEL_5
3080225 "" //2x9 T_CHAN_AREA_4_CHANNEL_6
3080226 "" //2x9 T_CHAN_AREA_4_CHANNEL_7
3080227 "" //2x9 T_CHAN_AREA_4_CHANNEL_8

/* Name of channel area 5 and names of channels of channel
area 5 */
3080228 "" //2x9 T_CHAN_AREA_5
3080229 "" //2x9 T_CHAN_AREA_5_CHANNEL_1
3080230 "" //2x9 T_CHAN_AREA_5_CHANNEL_2
3080231 "" //2x9 T_CHAN_AREA_5_CHANNEL_3
3080232 "" //2x9 T_CHAN_AREA_5_CHANNEL_4
3080233 "" //2x9 T_CHAN_AREA_5_CHANNEL_5
3080234 "" //2x9 T_CHAN_AREA_5_CHANNEL_6
3080235 "" //2x9 T_CHAN_AREA_5_CHANNEL_7
3080236 "" //2x9 T_CHAN_AREA_5_CHANNEL_8

/* Name of channel area 6 and names of channels of channel
area 6 */
3080237 "" //2x9 T_CHAN_AREA_6
3080238 "" //2x9 T_CHAN_AREA_6_CHANNEL_1
3080239 "" //2x9 T_CHAN_AREA_6_CHANNEL_2
3080240 "" //2x9 T_CHAN_AREA_6_CHANNEL_3
3080241 "" //2x9 T_CHAN_AREA_6_CHANNEL_4
3080242 "" //2x9 T_CHAN_AREA_6_CHANNEL_5
3080243 "" //2x9 T_CHAN_AREA_6_CHANNEL_6
3080244 "" //2x9 T_CHAN_AREA_6_CHANNEL_7
3080245 "" //2x9 T_CHAN_AREA_6_CHANNEL_8

```



```

/* Name of channel area 7 and names of channels of channel
area 7 */
3080246  " "           //2x9 T_CHAN_AREA_7
3080247  " "           //2x9 T_CHAN_AREA_7_CHANNEL_1
3080248  " "           //2x9 T_CHAN_AREA_7_CHANNEL_2
3080249  " "           //2x9 T_CHAN_AREA_7_CHANNEL_3
3080250  " "           //2x9 T_CHAN_AREA_7_CHANNEL_4
3080251  " "           //2x9 T_CHAN_AREA_7_CHANNEL_5
3080252  " "           //2x9 T_CHAN_AREA_7_CHANNEL_6
3080253  " "           //2x9 T_CHAN_AREA_7_CHANNEL_7
3080254  " "           //2x9 T_CHAN_AREA_7_CHANNEL_8

/* Name of channel area 8 and names of channels of channel
area 8 */
3080255  " "           //2x9 T_CHAN_AREA_8
3080256  " "           //2x9 T_CHAN_AREA_8_CHANNEL_1
3080257  " "           //2x9 T_CHAN_AREA_8_CHANNEL_2
3080258  " "           //2x9 T_CHAN_AREA_8_CHANNEL_3
3080259  " "           //2x9 T_CHAN_AREA_8_CHANNEL_4
3080260  " "           //2x9 T_CHAN_AREA_8_CHANNEL_5
3080261  " "           //2x9 T_CHAN_AREA_8_CHANNEL_6
3080262  " "           //2x9 T_CHAN_AREA_8_CHANNEL_7
3080263  " "           //2x9 T_CHAN_AREA_8_CHANNEL_8

```

The OEM user can enter texts by inserting them in the blank character strings (see above " ")!

8 characters are possible per softkey bar; the carriage return is created with characters "%n": e.g., "NC3Turn%nChann1".

## 4.4 Selecting the tool management

The ShopTurn, ShopMill tool management function is activated using the display machine data MD 9414: TM\_KIND\_OF\_TOOLMANAGEMENT for the representation of ShopMill/ShopTurn.

MD 9414 = 1	ShopMill/ShopTurn tool management
MD 9414 = 0	Standard tool management (default setting)

All the information needed for setup can be found in:

**References:**        /FBW/Description of Functions - Tool Management  
                          /FBSP/Description of Functions - ShopMill  
                          /FBT/Description of Functions - ShopTurn

## 4.5 Tool fine correction

The limit values of the tool fine correction are contained in the display machine data MD 9450: WRITE\_TOA\_FINE\_LIMIT.

When entering the tool wear fine, the difference between the previous value and the new value may not exceed the limit entered here.

MD 9450 can be changed only when the display machine data MD 9202: USER\_CLASS\_TOA\_WEAR contains a valid protection level.

**References:** /FB1/A2: Various Interface Signals

## 4.6 Activating tool correction immediately

The display machine data MD 9440: ACTIVATE\_SEL\_USER\_DATA can be used to ensure that the active tool offset is activated as soon as the parts program is in the "Reset" or "Stop" state.

When the function is used in the reset state, the channel-specific machine data MD 20110: RESET\_MODE\_MASK Bit 0 = reset mode must be set appropriately. Then the tool offset is not reset on a reset.

**References:** /FB1/K2: Axis Types, Coordinate Systems, Frames,...

## 4.7 Selecting tools via a D number (flat D No.)

The general machine data MD 18102: TYPE\_OF\_CUTTING\_EDGE can be used to activate the "flat D number management".

Value = 0 NCK manages the T and D numbers = default setting

Value = 1 Only tools with the D number structure can be selected.

The D number can be assigned just once per tool, i.e., each D number stands for one and no more than one offset data block.

**References:** /FB1/W1: Tool offset

## 4.8 Workpiece template

Workpiece templates for a program can be used for similar programs. For that purpose, you must create a workpiece with the name `_TEMPL_`. If you press the "New" softkey, this workpiece is copied and renamed. All files in this directory are also copied into the new directory.

**Example:**

Suppose you want to create a workpiece template with the content "Daten.ini" and "TEST.MPF":

```
Workpiece: _TEMPL_  
_TEMPL_.MPF  
DATEN.INI  
TEST.MPF
```

Operating sequence:

1. Press the "New" softkey
2. Name: Define "ACHSE"
3. The new workpiece "ACHSE" is created with the following files:  
ACHSE.MPF  
DATEN.INI  
TEST.MPF

## 4.9 Analog spindles

The spindle load from 0 to 100% can be displayed in the spindle window for up to two analog spindles. For that, the following bytes must be set in DB 19:

Analog spindle: DB19.DBB 6

Analog spindle: DB19.DBB 7

**References:**        /BEM/Operator's Guide HMI Embedded

## 4.10 Work offset

---

### Note

The functions of HMI Embedded can be altered in the INI files, display machine data or the user interface.

---

### 4.10.1 Changing access protection to base offset Softkey

#### Parameters operating area

The display machine data

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 base frames are also shown/hidden in the "Work offset" window and in the "Active WO + offsets" window.

---

### Note

Display machine data MD\_9210 USER\_CLASS\_WRITE\_ZOA (write coarse offset) has no effect with "active zero offset" in the "Parameters" operating area.

---

#### Machine operating area

The display machine data 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 and/or G500 can be entered in the work offset field.

### 4.10.2 Fine work offset and base offset

This function refers to the offset in the screen form "Settable work offset" and "Base work offset".

Another column for the fine offset is inserted alongside the column for entering the offsets.

The inputs will be validated using the display machine data

- MD 9203: USER\_CLASS\_WRITE\_FINE
- MD 9451: MM\_WRITE\_ZOA\_FINE\_LIMIT

The first data describes the protection level, and the second data the amount of a possible value that can be entered.

The base work offset (base WO) is displayed like a settable work offset and can be selected and modified using the "Base WO" softkey in the screen form "Overview of work offsets".

The general machine data MD 18600: MM\_FRAME\_FINE\_TRANS = 1 activates the fine offset for all settable FRAMES and the base frame.

Set the general machine data

MD 18600: MM\_FRAME\_FINE\_TRANS = 0, the fine offset cannot be entered or programmed.

**References:** /FB1/K2: Axes, Coordinate Systems, Frames,...External Work Offset.

### 4.10.3 Presetting actual value memory, preset, Scratching

#### Introduction

The behavior of the **scratching** and **preset actual value memory, 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.)

**References:** /BEM/HMI Embedded Operator's Guide

With the introduction of system frames in software version 6, two variants are available for the functions. The variants are differentiated using a channel-specific machine data MD 28082: MM\_SYSTEM\_FRAME\_MASK:

Bit:	0	Preset, scratching
	1	External work offset
	2	TCARR, PAROT
	3	TOROT, TOFRAME
	4	Workpiece reference points
	5	Cycles
	6	Frame transformation

Variant 1 **without system frame:**

The MD does not exist (older NCK software versions) or bit 0 for system frame for scratching and preset is not set.

Variant 2 **with system frame:**

Bit 0 for system frame for scratching and preset is not set in the MD.

Display machine data MD 9422: MA\_PRESET\_MODE defines the PRESET/base offset in the JOG mode.

0: No softkey

1: The old PRESET in the Machine operating area (default).

2: Preset actual value memory:

NCK without system frame: Setting only possible with active G500 in the base offset, otherwise error message.

NCK with system frame: Setting in the system frame possible.

The basic frame is not used with G500 in the system frame.

3: Preset actual value memory: NCK with/without system frame: Setting possible in the currently active frame.

---

### Note

To hide the PRESET, preset actual value memory and scratching softkeys, the display machine data 9220: MM\_USER\_CLASS\_PRESET can be set.

---

## Presetting the actual value memory

### Presetting the actual value memory with system frame

G500 active

- The values are written to the system frame

G500 not active

- An error message is output.

### Presetting the actual value memory without system frame

G500 active (adjustable frames reset)

- The values are written to the first basic frame (up to software version 4),  
or (as of software version 5)

depending on the entry in display machine data

MD 9245: MA\_PRESET\_FRAMEIDX = index of basic frame  
to the set basic frame

## Scratching

### Scratching with system frame

G500 active

- The values are written to the system frame.

G500 not active

- The values are written in the currently active frame.

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**

G500 active (adjustable frames reset)

- The values will be written in the set basic frame depending on the entry in the display machine data  
MD 9245: \$MM\_MA\_PRESET\_FRAMEIDX = index of base frame

G500 not active

- The values are written in the currently active frame.

G500 not active

- An error message is output.

You can select any tool you want.

**References:** /FB1/Description of Functions, K2: Frames...**4.10.4 Hiding display of the machine axes****Function**

The operator can use the channel-specific machine data MD 20098: DISPLAY\_AXIS to suppress the display of the axes in the Machine area. All axes are displayed by default.

Bits 16 to 31 Show (=1)/hide (=0) machine axes (MCS).

20098 <b>MD number</b>	DISPLAY_AXIS <b>Display or hide axis on MMC/HMI</b>		
Default setting: 0xFFFFFFFF	Minimum input limit: 0	Maximum input limit: 0xFFFFFFFF	
Changes effective after POWER ON	Protection level: 2/7	Unit: -	
Data type: DWORD	Applies as of SW 4.4		
Significance:	<b>Bits 16 to 31: MCS</b>		
	Bit 16=	1	Display machine axis in actual-value windows
		0	Hide machine axis in actual-value windows
	(Bit 17)		Unassigned
	Bit 18=	1	Display machine axis in base offset window
		0	Hide machine axis in base offset window
	Bit 19=	1	Display machine axis in handwheel selection window
		0	Hide machine axis in handwheel selection window

## Special cases

### Homing and Safety Integrated

The channel-specific machine data MD 20098: DISPLAY\_AXIS is not evaluated in the displays for "REF" homing mode, namely the "Axes Reference Point and Acknowledge Safe Position" display. All machine axes are always displayed.

Therefore the operator cannot change the values for the base work offset without the appropriate access right.

The access right 7 must be entered as default value in the display machine data MD 9247 and 9248.

## 4.10.5 Displaying the geometry axes

Via the display machine data MD 9421: In MA\_AXES\_SHOW\_GEO\_FIRST you specify whether the geometry axes are to be displayed at the first position or not (e.g., before the auxiliary axes).

MD 9421 = 1	Display geometry axes at the first position
MD 9421 = 0	Do not display geometry axes at the first position (default setting)

## 4.11 Specifying the machine and rotary axis position

### Singularities with special kinematics and robots

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

### Setting the numerical basis

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 set via the display machine data:

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

Possible settings 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'

**References:** /FB1/Description of Functions, K2: Frames...

## Show/hide STAT and TU

The STAT and TU values are displayed in the axis screens (actual-value windows) and in teach mode (insert block).

You can choose whether or not to show the STAT and TU values by setting the following display machine data:

STAT and TU are not displayed:

MD 9242: MA_STAT_DISPLAY_BASE	= 0	of articulated position STAT
MD 9243: MA_TU_DISPLAY_BASE	= 0	of rotary axis position TU

STAT and TU are displayed:

MD 9242: MA_STAT_DISPLAY_BASE	= 1	of articulated position STAT
MD 9243: MA_TU_DISPLAY_BASE	= 1	of rotary axis position TU

## 4.12 Program simulation

### 4.12.1 Rotary simulation and program test

PLC signal DB19.DBB20.6 is set to 1 when simulation is activated. This signal can be evaluated in order to switch to the program test and to remove the axis or control enable (to prevent axis movements during the simulation). The signal is reset to 0 when simulation is terminated.

In order to simulate certain operations during a dry run even though the axis/controller pulses have been disabled, the transfer of status "Ext. pulse disable active, terminal 663 open" to the NC has to be prevented.

The machine manufacturer can achieve this by evaluating the above-mentioned PLC signal and activating the dry run simultaneously with NC Start.

If the drive machine data MD 1012: FUNC\_SWITCH, bit 2 = 0 is also set, the "Ext. pulse disable active, terminal 663 open" status will not be forwarded to the NC.

**References:** /FBA/DB1: Operational messages/alarm responses

## 4.12.2 Turning/milling simulation

Via the display machine data MD 9020: TECHNOLOGY, you define the basic configuration for the free contour programming of turning/milling.

MD 9020 = 1 Turning machine configuration  
MD 9020 ≠ 1 Milling machine configuration

**References:** /FB1/A2: Various Interface Signals

## 4.12.3 Milling simulation 3D graphic

This function is a software option named "Milling Simulation", with order number 6FC5253-0AE02-0AA0.

If the option is set, the "3D Simulation" softkey will appear and can be used to select the function.

Operation of milling machine simulation is described in:

**References:** /BEM/Operator's Guide HMI Embedded

## 4.13 Measuring cycle support in the G code editor

In HMI Embedded you can integrate screen forms, which provide for programming measuring cycles, into the G code editor. These cycles can then be recompiled automatically.

The "measuring cycles" function is a software option with order no. 6FC520-0BX00-0AB0.

For further information about measuring cycle support, please refer to:

**References:** /BNM/User's Guide Measuring Cycles

- In the ANWENDER-ZYKLEN\COMMON.COM file, remove the semicolon ";" in front of the following lines:  
;SC326 = AEDITOR.COM ;Turning measuring cycles (horizontal softkey 6 on  
;the extended softkey bar in the G code editor)  
;SC327 = AEDITOR.COM ;Milling measuring cycles (horizontal softkey 7 on  
;the extended softkey bar in the G code editor)

In this way, you establish the connection between the softkeys, with which you call the measuring cycle support, and the configuration file of these support screens.

- In the ANWENDER-ZYKLEN\COMMON.COM file, remove the semicolon ";" in front of the following line:  
;SC617 = STARTUP.COM ;Startup operating area (horizontal  
;softkey 7 on the extended softkey bar  
;in the G code editor)

In this way, you can change the properties of the measuring cycle support in the "Startup" operating area.

Change the properties of the measuring cycle support in the "Startup" ">" "Measuring Cycles" menu.

## 4.14 Expanding user displays from the operator interface

"Expand user interface" is implemented with an interpreter and configuration files that contain the descriptions of the user interfaces. The Expand user interface tools can be used to create user interfaces that display functional expansions designed by the machine manufacturer or end user, or simply to implement your own screen form layout.

User environments configured by Siemens or non-Siemens machine manufacturers can be modified or replaced. The interpreter is available for HMI Embedded sl, ShopMill and ShopTurn on NCU as well as HMI Advanced.

Parts programs, for example, can be edited on user interfaces created by users. Screen forms can be created directly on the control system.

The application diskette and PaintShopPro (<http://www.jasc.com>) are required to create graphics and images.

The tool box supplied contains configuration examples for new screen forms. You can also use these examples as a template for creating your own forms.

You cannot call user-defined dialog windows simultaneously in different channels, i.e., you can only use the MMC command with a 1:1 link.

The exact procedure is described in submanual:

**References:** /IAM/ BE1, Expanding the user interface.

## 4.15 Coordinate systems

### 4.15.1 Actual value display

Via the display machine data MD 9424: MA\_COORDINATE\_SYSTEM you can set whether

- the position of the workpiece coordinate system (programmed position, corresponds to the default setting) or
- the holding position of the active tool relative to the workpiece zero must be shown in the actual-value display.

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

**References:**     /FB1/K2: Axes, Coordinate Systems, Frames,... External Work Offset

### 4.15.2 Position of the coordinate system

The display machine data MD 9650: CMM\_POS\_COORDINATE\_SYSTEM is used to set the position of the coordinate system for milling.

Table 4-3     Machine data 9650

9650 MD number	CMM_POS_COORDINATE_SYSTEM Position of coordinate system	
Default setting 0	Min. input limit: 0	Unit: -
Data type BYTE		Valid as of SW release: SW 05.01.13
Meaning	With this machine data you adapt the coordinate system of the user interface to the coordinate system of the machine. The coordinate system can assume the following positions.	

---

#### Note

The display machine data MD 9610: CTM\_POS\_COORDINATE\_SYSTEM is used to set the position of the coordinate system for milling.

---

**References:**     /FB1/K1: Mode group, channel, program operation mode

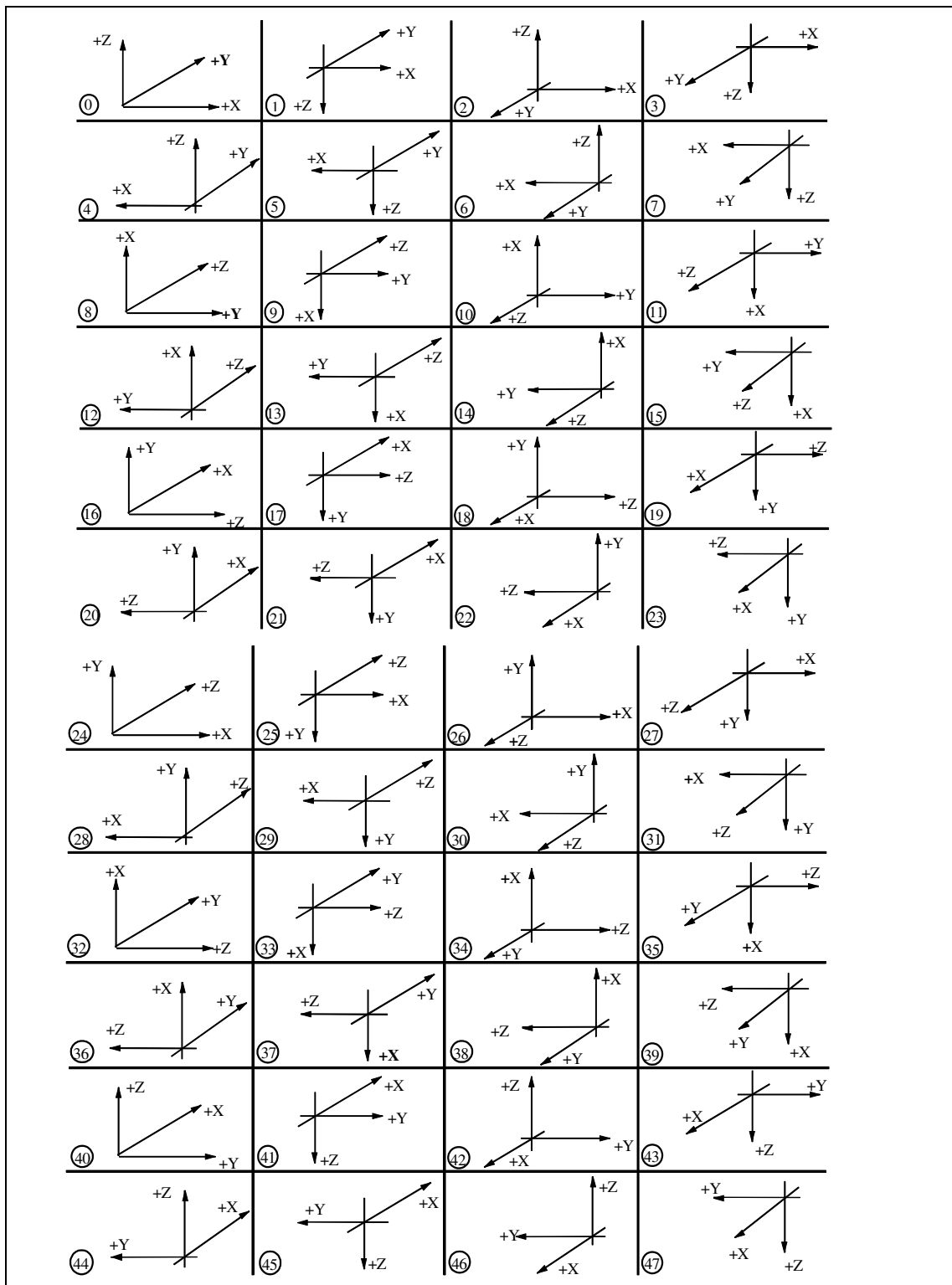


Fig. 4-2 Coordinate system

## 4.16 Displaying and editing system resources

Up-to-date information about the system resources currently used (utilization display) can be displayed and edited for the NCK and HMI Embedded areas.

### Requirements

As different protection levels are assigned to machine data per default, the set access authorization for the editing of machine data must be sufficiently high. Depending on the authorization level, it will be possible to edit data such as cycles and machine data.

### Procedure

Another softkey menu is displayed when you select the ETC key ">" in the Startup operating area.

When you select the "NC memory" softkey, the memory overview screen appears, showing the amount of NC user memory currently available for programs and data. The "SRAM" and "DRAM" areas have been provided to enable you to examine or edit the machine data that configure the memory. These areas in turn are divided into further groups.

#### 4.16.1 Static user memory (SRAM)

When you select the "SRAM" softkey, the total memory allocation is displayed and can be modified at the top of the screen.

The memory-configuring machine data are divided into the following groups:

- Tool management
- Global user data
- Curve tables
- Compensation
- File system / program memory
- Protection zones

IBN	CHAN1	Jog	\MPF.DIR TEST.MPF
Kanal Reset			
Programm abgebrochen			ROV
SRAM-Belegung in Byte (Anwenderspeicher für Programme und Daten)			
-Statischer Anwenderspeicher SRAM nach NC-Reset (in Byte)-			
Gesamt			807936
Mit Filesystem belegt			581084
Frei			258848
-Nicht verwendeter Speicher (in Byte)-			
Werkzeugverwaltung			10504
Globale Anwenderdaten			8544
Kurventabellen			0
Kompensationen			0
Dateisystem / Programmspeicher			18540
Schutzbereiche			0
^			
Speicher- Übersicht	SRAM	DRAM	

Fig. 4-3 SRAM Static user memory

## Detailed view

Click the area of interest with the cursor and select the "Details" softkey. The current data for the relevant machine data are then displayed.

The block header contains the number and name of the machine data. The currently set values are output on the lines below.

You can change the memory setting in the gray field after "New value" or "New number" to the right of the field containing the actual value.

This new setting is entered temporarily. Your entry is automatically checked for limits and an appropriate message output in the dialog line if you make a mistake.

The total available memory, taking into account any modified values, is displayed in the lower half of the screen.

The "Axis+" and "Axis-" softkeys are displayed for axis-specific machine data.

The "Channel+" and "Channel-" are softkeys displayed for channel-specific machine data.

When you select the "Accept" softkey, the temporary values of the displayed machine data are transferred to the NC.

You can exit the detail view by selecting the "Cancel" softkey or the Recall key "^".

---

### Note

Once you have pressed "Accept" softkey, the "Cancel" softkey is no longer operative, i.e., any changes you have made cannot be undone.

---

## Data backup

When you change the allocation of a memory, alarm 4400 "Machine data alteration will cause reorganization of buffered memory (data loss)" appears. Since data are not backed up automatically, to avoid data loss you must back up the data yourself by way of an NCK Reset before you transfer the MD change.

## Memory allocation: Tool management

In the main SRAM screen, click the "Tool management" area with the cursor and then click the vertical "Details" softkey. The following screen appears:

IBN	CHAN1	Auto	NMPF.DIR TEST.MPF		
Kanal Reset					
Programm abgebrochen			ADV		
<b>Werkzeugverwaltung</b>					
-18082 MM_NUM_TOOL					
Speicheränderung					
Momentan eingestellte Anzahl	30	Neue Anzahl	30		
Momentan verwendete Anzahl	0	Änderung SRAM	0	Byte	
-18088 MM_NUM_TOOL_CARRIER					
Speicheränderung					
Momentan eingestellte Anzahl	0	Neue Anzahl	0		
Momentan verwendete Anzahl	0	Änderung SRAM	0	Byte	
Momentan freier Anwenderspeicher				SRAM	226852 Byte
Freier Anwenderspeicher nach NC-Reset				SRAM	258848 Byte
^ Anzahl Werkzeuge in NCK (SRAM)					
Werkzeuge	Werkzeug-schneiden	Magazine	Magazine OEM	Werkzeug-parameter	Werkzeug-verwalt.

Fig. 4-4 SRAM - Tools

By selecting the horizontal softkeys, you can view or edit the general machine data memory values of other subgroups.

- "Tools": MD 18082: MM\_NUM\_TOOL  
MD 18088: MM\_NUM\_TOOL\_CARRIER
- "Tool edges": MD 18100: MM\_NUM\_CUTTING\_EDGES\_IN\_TOA  
MD 18110: MM\_MAX\_SUMCORR\_PER\_CUTTEDGE  
MD 18104: MM\_NUM\_TOOL\_ADAPTER
- "Magazines": MD 18084: MM\_NUM\_MAGAZINE  
MD 18086: MM\_NUM\_MAGAZINE\_LOCATION



"Magazines OEM":	MD 18090: MM_NUM_CC_MAGAZINE_PARAM MD 18092: MM_NUM_CC_MAGLOC_PARAM
"Tool parameter":	MD 18094: MM_NUM_CC_TOA_PARAM MD 18096: MM_NUM_CC_TDA_PARAM MD 18098: MM_NUM_CC_MON_PARAM
"Tool management":	MD 18080: MM_TOOL_MANAGEMENT_MASK

### Memory allocation: Global user data

When you select the "Global user data" area and click the "Details" softkey, you can display and edit the memory settings in the machine data of the following areas:

"GUD files":	MD 18118: MM_NUM_GUD_MODULES MD 18150: MM_GUD_VALUES_MEM
"Number of variables":	MD 18120: MM_NUM_GUD_NAMES_NCK MD 18130: MM_NUM_GUD_NAMES_CHAN

### Memory allocation: Curve tables

When you select the "Curve tables" area and click the "Details" softkey, you can display and edit the memory settings in the following machine data:

MD 18400: MM_NUM_CURVE_TABS
MD 18402: MM_NUM_CURVE_SEGMENTS
MD 18404: MM_NUM_CURVE_POLYNOMS

### Memory allocation: Compensation

When you select the "Compensations" area and click the "Details" softkey, you can display and edit the memory settings in the machine data of the following areas:

"Sag compensens.": MD 18342: MM\_NUM\_CEC\_MAX\_POINTS[ ]  
These configuring machine data are indexed machine data. The index is set in the top half of the screen.

"E. error compensens.": MD 38000: MM\_ENC\_COMP\_MAX\_POINTS[0]  
MD 38000: MM\_ENC\_COMP\_MAX\_POINTS[1]

There are only two tables for the encoder/spindle compensation interpolation points. There is no need to select an index.

"Q. error compensens.": MD 38010: MM\_QEC\_MAX\_POINTS[0]  
There is only one table for quadrant error compensation. There is no need to select an index.

### Memory allocation: File system / program memory

When you select the "File system / program memory" area and click the "Details" softkey, you can display and edit the memory settings in the machine data of the following areas:

"Directories": MD 18310: MM\_NUM\_DIR\_IN\_FILESYSTEM  
MD 18270: MM\_NUM\_SUBDIR\_PER\_DIR

"Files": MD 18320: MM\_NUM\_FILES\_IN\_FILESYSTEM  
MD 18280: MM\_NUM\_FILES\_PER\_DIR

### Memory allocation: Protection zones

When you select the "Protection zones" area and click the "Details" softkey, you can display and edit the memory settings in the machine data of the following areas:

"Global prot. zone": MD 18190: MM\_NUM\_PROTECT\_AREA\_NCK  
"Chann-spec. prot. zone": MD 28200: MM\_NUM\_PROTECT\_AREA\_CHAN  
MD 28210: MM\_NUM\_PROTECT\_AREA\_ACTIVE

### 4.16.2 Dynamic user memory (DRAM)

When you select the "DRAM" softkey, the total memory allocation is displayed and can be modified at the top of the screen.

To help you optimize memory utilization effectively, the memory-configuring machine data are divided into the following groups:

- Local user data
- REORG
- Cycles
- Interpolation buffer
- Execute from external
- Synchronized actions
- Macros
- Tool management
- Protection zones

IBN	CHAN1	Auto	\MPPF.DIR TEST.MPF
Kanal Reset			
Programm abgebrochen			ROV
<b>DRAM-Belegung in Byte (Anwenderspeicher für Daten)</b>			
-Dynamischer Anwenderspeicher DRAM nach NC-Reset (in Byte)-			
Gesamt			10577920
Frei			4300612
-Nicht verwendeter Speicher (in Byte)-			
Lokale Anwenderdaten			30364
REORG			1368
Zyklen			4332
Interpolationspuffer			278740
Abarbeiten von Extern			0
Synchronaktionen			0
Makros			7092
Werkzeugverwaltung			0
Schutzbereiche			0
Speicher- Übersicht			
	SRAM	DRAM	

Fig. 4-5 DRAM dynamic user memory

Operating instructions: See Section "Static user memory (SRAM), Detail view"

Additional elements in DRAM memory allocation screen:

Since the memory of some data areas is allocated to dynamic processes, their memory requirements vary. By selecting "Peak values=0" softkey, you can reset the peak values of the currently displayed machine data to zero.

### Memory allocation: Local user data

In the DRAM main screen, click the "Local user data" area with the cursor and then click the "Details" vertical softkey to view and edit the memory settings of the following machine data:

MD 28020: MM\_NUM\_LUD\_NAMES\_TOTAL  
MD 28040: MM\_LUD\_VALUES\_MEM

### Memory allocation: Reorganization

When you select the "REORG" area and click the "Details" vertical softkey, you can display and edit the memory settings in the following machine data:

MD 28000: MM\_REORG\_LOG\_FILE\_MEM  
MD 28010: MM\_NUM\_REORG\_LUD\_MODULES

### Memory allocation: Cycles

When you select the "Cycles" area and click the "Details" softkey, you can display and edit the memory settings in the following machine data:

MD 18170: MM\_NUM\_MAX\_FUNC\_NAMES  
MD 18170: MM\_NUM\_MAX\_FUNC\_PARAM

### Memory allocation: Interpolation buffer

When you select the "Interpolation buffer" area and click the "Details" softkey, you can display and edit the memory settings in the following machine data:

MD 28060: MM\_IPO\_BUFFER\_SIZE  
MD 28070: MM\_NUM\_BLOCKS\_IN\_PREP

### Memory allocation: Execute from external

When you select the "Execution from external source" area and click the "Details" softkey, you can display and edit the memory settings in the following machine data:

MD 18360: MM\_EXT\_PROG\_BUFFER\_SIZE  
MD 18362: MM\_EXT\_PROG\_NUM

### Memory allocation: Synchronized actions

When you select the "Synchronized actions" area and click the "Details" softkey, you can display and edit the memory settings in the machine data of the following areas:

"Synchronized action 1": MD 28250: MM\_NUM\_SYNC\_ELEMENTS  
MD 28252: MM\_NUM\_FCTDEF\_ELEMENTS  
MD 28258: MM\_NUM\_AC\_TIMER

"Synchronized action 2": MD 28254: MM\_NUM\_AC\_PARAM  
MD 28256: MM\_NUM\_AC\_MARKER

### Memory allocation: Macros

When you select the "Macros" area and click the "Details" softkey, you can display and edit the memory settings in the following machine data:

MD 18160: MM\_NUM\_USER\_MACROS

### Memory allocation: Tool management

When you select the "Tool management" area and click the "Details" softkey, you can display and edit the memory settings in the following machine data:

MD 18105: MM\_MAX\_CUTTING\_EDGE\_NO  
MD 18106: MM\_MAX\_CUTTING\_EDGE\_PERTOOL

### Memory allocation: Protection zones

See Section: Static user memory (SRAM), memory allocation: Protection zones.

## 4.17 Remote diagnosis "RCS Host Embedded / RCS Viewer Embedded"

It is possible for a service engineer to monitor and influence a controller from a remote PC (remote PC = viewer), i.e. the same display is shown on the remote PC as on the operator panel front.

Remote diagnosis has the following services:

- Direct access to the HMI-Embedded via network
- Data exchange (file transfer)

The user has to apply for a license (operator authorization), order number:

6FC5800-0AP30-0YB0

before he can use the remote diagnosis software "RCS Host Embedded" on the NCU.

See Chapter 3: Licensing

The remote diagnosis software "RCS Viewer Embedded V1.1" for PC (Windows) is stored on the remote diagnosis CD "RCS Viewer Embedded", order number: 6FC6000-6DC81-0BA0.

### Requirements

The following hardware and software is required:

- Establish a tunnel connection between NCU 7x0 (Ethernet interface X130) (machine) and the remote PC (viewer), e.g. via hardware router with so-called VPN (Virtual Private Network) functionality, e.g. Cisco 803
- Windows PC (XP/ NT4/ 95/ 98) with network connection
- "Viewer" application (mmcR.exe) from the remote diagnosis CD (remote PC)
- Ethernet cable
- Logic drive for data exchange (file transfer)

### Connection structure

The example shows the connection via the ISDN network (64 kbit/s). This telephone network provides the capability of connecting to remote LANs via a tunnel using the point-to-point protocol (PPP). The NCU 7x0 is connected to the ISDN network via the Ethernet interface with a hardware router (with VPN functionality).

Structure example of an ISDN connection:

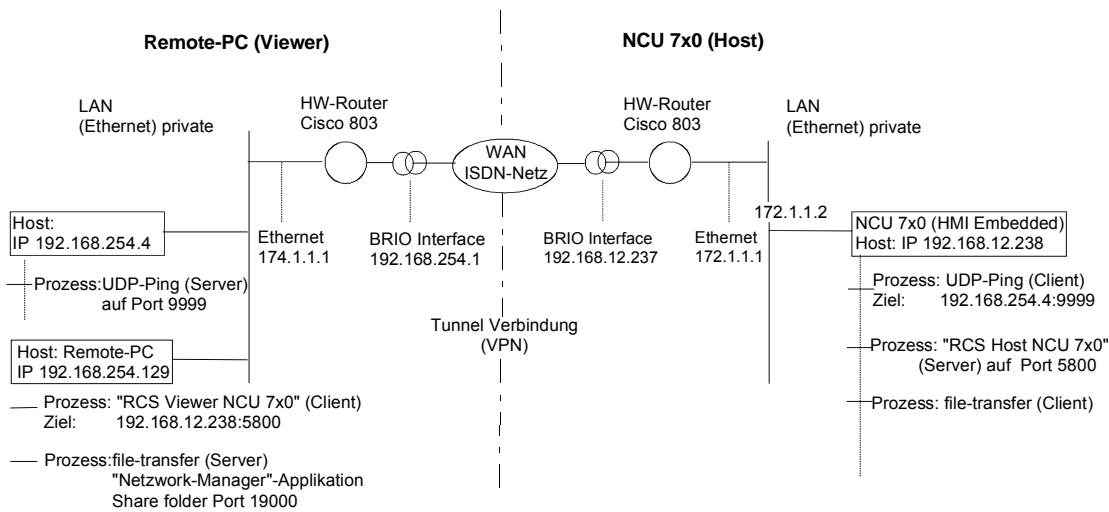


Figure 4-1: Example: Connection structure of the remote diagnosis

## Remote diagnosis sequence

A machine manufacturer who provides this service has selected an appropriate topology for a tunnel connection and will provide support when setting up the remote diagnosis.

The relevant "Ping server" on the remote PC must be activated for this purpose. This server has the task of accepting the incoming "Ping" data (e.g. serial number) of the requesting machine and registering the IP address of the requesting machine. If this server application is not provided by the service provider, it is possible to configure the RCS viewer application so that a simple "Ping" server process can be started on the RCS Viewer.

The IP address, the port, and the protocol used for this Ping server are specified by the service provider and must be set on the machine.

The remote diagnosis is initiated by the machine. The NCU 7x0 is connected to the appropriately configured router via an Ethernet cable and starts the remote diagnosis via the operating interface. The NCU 7x0 then sends cyclical preset "Ping" data to the remote PC and waits for its response. The transmission of the "Ping" causes the router to establish the connection (e.g. switched ISDN connection).

When the response is received from the remote PC, the NCU 7x0 is ready for the data communication for the remote diagnosis. For this, a server process is started on the NCU 7x0, which is responsible for the exchange of graphic information to the remote PC and the keyboard inputs from the remote PC.

The exchange is performed via the TCP/IP protocol and a "permanently" agreed port. A so-called "Viewer" application (mmcR.exe) is installed on a remote PC as communication partner. From now on, all further activities will be initiated by the remote PC.

With the arrival of the "Ping" on the PC, the IP address of the requesting NCU 7x0 is known and the "Viewer" application (mmcR.exe) with this destination IP address can be started on a PC. This application runs as a client. The machine can be monitored and controlled via this remote PC.

The transfer of files between the remote PC and the machine is handled with local drives (see Chapter 3: Network operation with HMI Embedded).

## 4.17.1 Configuring "RCS Host Embedded"

### Adapting NCU 7x0 to router (VPN network)

NCU 7x0 (Ethernet interface X130) and router are connected via an Ethernet cable.

If the NCU 7x0 is already in the network, save the current network configuration. To adapt the network configuration of the NCU 7x0 to the configuration of the Ethernet interface of the router, proceed as follows:

Start the machine, press the "Start-up" -> "HMI" -> "Netw. config." softkeys and make the following entries:

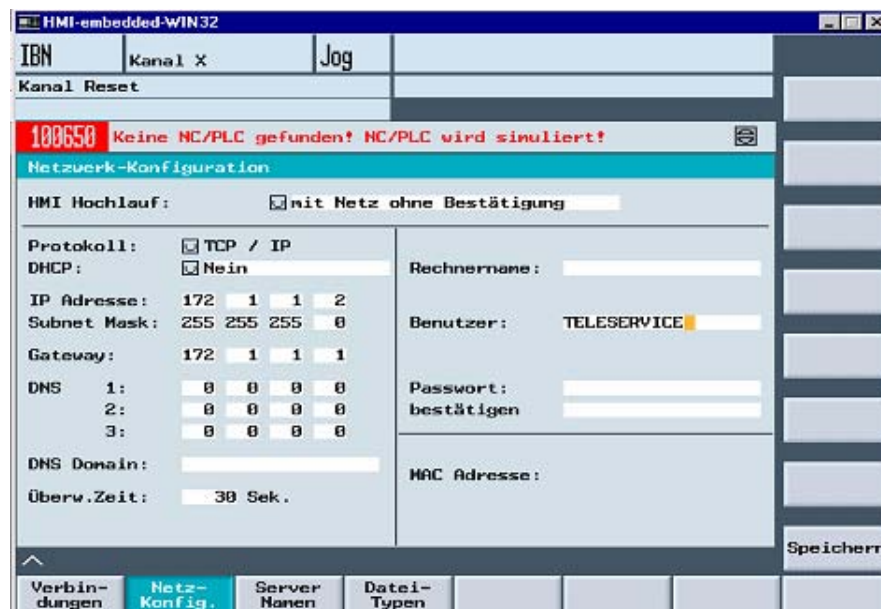


Figure 4-2: Configuration of the Ethernet interface "Router NCU 7x0"

The entries

- IP address
- Subnet mask
- Gateway

are already specified by the router configuration and are only used as examples here.

When you press the "Save" softkey, the NCU 7x0 is restarted.

## Adapting the machine to the remote PC

In the "Start-up" operating area, press the ETC key ">" and the softkey "Remote diagnosis". The following window appears:

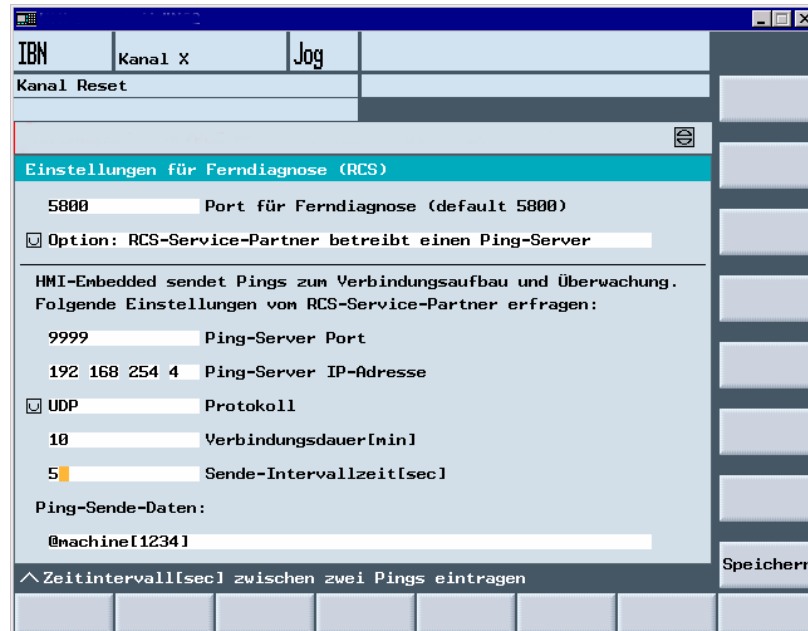


Figure 4-3: Setting up logon server

The port number, default 5800, is entered in the "Port for remote diagnosis (default 5800)" input field for the server process (remote diagnosis). The same port number must be entered in the "Viewer" application on the remote PC.

### Option: "RCS service partner operates a Ping server" selected

By selecting the option "RCS service partner operates a Ping server", the remote PC is triggered to operate a Ping server. The subsequent settings must be obtained from the service partner operating a Ping server:

- Ping server port
- IP address of the "Ping" server runs
- Protocol selection, UDP or TCP/IP
- Connection duration (min): Entry for the maximum time for the duration of the remote diagnosis. After this time expires, the connection is interrupted.
- Transmission interval (sec): The Ping data is transmitted cyclically in this interval from the NCU 7x0 to the remote PC. This is used for the monitoring of the connection.
- Ping transmission data: Entry of an agreed user data string



**Option: "RCS service partner operates a Ping server" deactivated**

This option should only be deselected when the initiative for the remote diagnosis is to come from the remote PC (e.g. when the remote PC and the NCU 7x0 are in the same Intranet and an individual NCU 7x0 in the LAN is to be accessed from a PC).

When this option is deactivated, no other entries are possible.

Pressing the "Save" softkey terminates the configuration of the server.

**Connections for data exchange (file transfer)**

In order to be able to perform a file transfer, connection entries (\\<remote\_ip>\<share folder name>) are automatically made by the system for the remote diagnosis. These connection entries are used by the "logical drives".

If, from the eight possible "logical drives", no connection is available, switch to the "Start-up" operating area, press the softkey "Connections" and delete a connection entry.

**4.17.2 Installing and configuring "RCS Viewer Embedded"****Requirements**

- Remote diagnosis CD with the "RCS Viewer Embedded" software
- Logical drive for the file transfer

**Installation**

The remote diagnosis is installed by running the "Setup.exe" program.

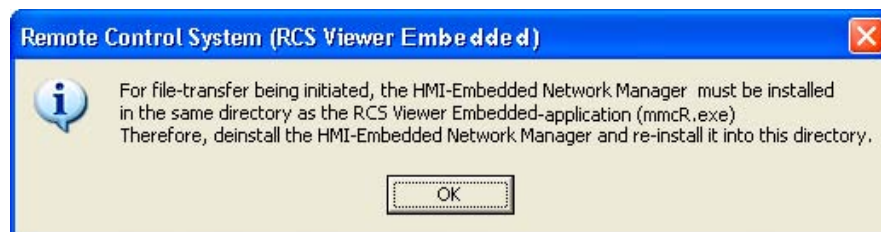


Figure 4-4: Message

Acknowledge the message.

**Note**

The Network Manager is not required with SINUMERIK 840D sl.

## Configuration

Start the "Viewer" application "mmcR.exe" on the remote PC.

If the default for the remote diagnosis port is changed, for example, due to firewall settings when the port is activated for incoming packages only, select the "Configuration" switch in the "RCS Viewer Embedded (Remote Control System)" window (see Figure 4-6).

The following window appears:

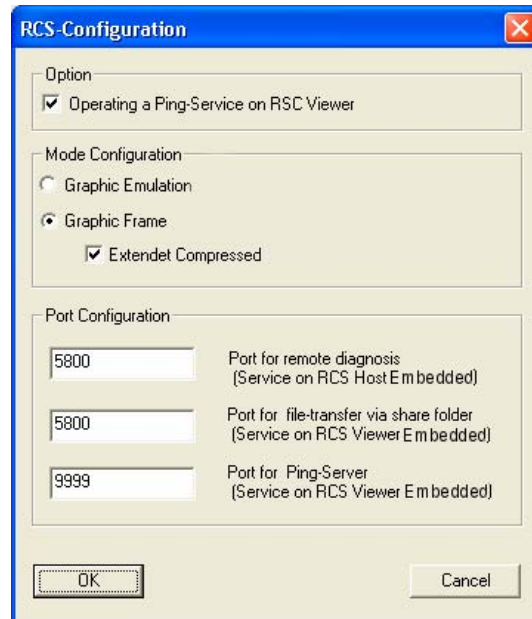


Figure 4-5: Port setting

In the "Option" area, you can specify that a Ping server process is set up on the RCS Viewer, which will wait for the incoming "Pings".

Enter the port numbers in the "Port Configuration" area.

## Additional settings

### Setting graphic transmission mode

The transmission mode for the screenshots is set in the "Mode Configuration" area. "Graphic Frame" and "Extended Compressed" are the default settings and feature the highest transmission rate for this remote diagnosis version.

---

### Note

The field "Graphic Emulation" is not supported with SINUMERIK 840D sl.

---

Click "OK" to save your settings. The following dialog appears in the next window:  
"The port-number has been stored.  
Please, restart your application."

When you click "OK" again, the system is restarted.

---

**Note**

Each time the configuration on the remote PC is changed, you have to restart the NCU.

---

### 4.17.3 Start remote diagnostics

#### NCU 7x0 (Host)

Activate the "RCS Host Embedded" in the "Diagnosis" operating area via the "Remote diagnosis" and "Start" softkeys.  
Status messages provide information on the current state.

When the connection to the remote PC is established, the NCU 7x0 waits for the "Viewer" application (mmcR.exe) to be started on the remote PC and the communication for the data exchange to be activated.

The message: "Ping server responds" appears.

This communication is performed via remote diagnosis which, per default, is applied to port number 5800.

When the message "Communication running" appears, operation of the machine from the remote PC is possible.

#### Remote PC (viewer)

The RCS Viewer can be started either via the Start menu or through a double-click on the "mmcR.exe" file.

If a "Ping" server is operated, then the IP address of the NCU 7x0 (Ethernet interface X130) is known on the remote PC after the first "Ping" has been received (e.g.: 147.54.235.46).

If no Ping server is available, enter the address manually.

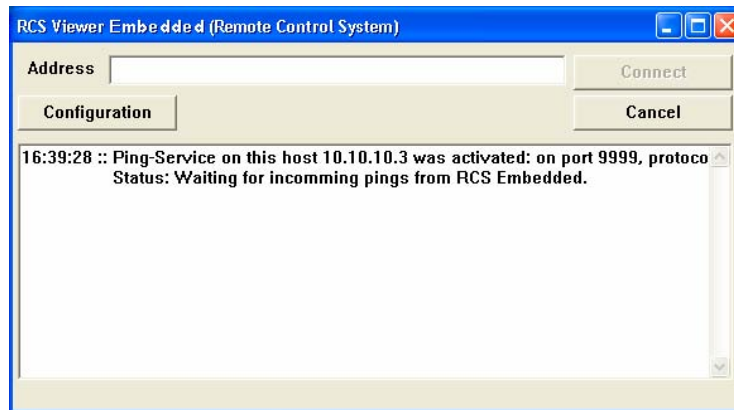


Figure 4-6: RCS Viewer waiting for incoming “Pings”

When you click the "Connect" button, the connection with the machine is activated via the remote diagnosis port. The corresponding status messages are output in the field below. When these are completed, the dialog is minimized and the HMI-Embedded interface appears.

Press the “Configuration” button to display the “RCS Configuration” window. See Fig. 4.5: Port setting

### File transfer

The file transfer is performed in the "Program" operating area. When you press the softkey defined via “logic drives”, the contents of the selected folder are displayed. The transfer can be performed using the vertical soft keys.

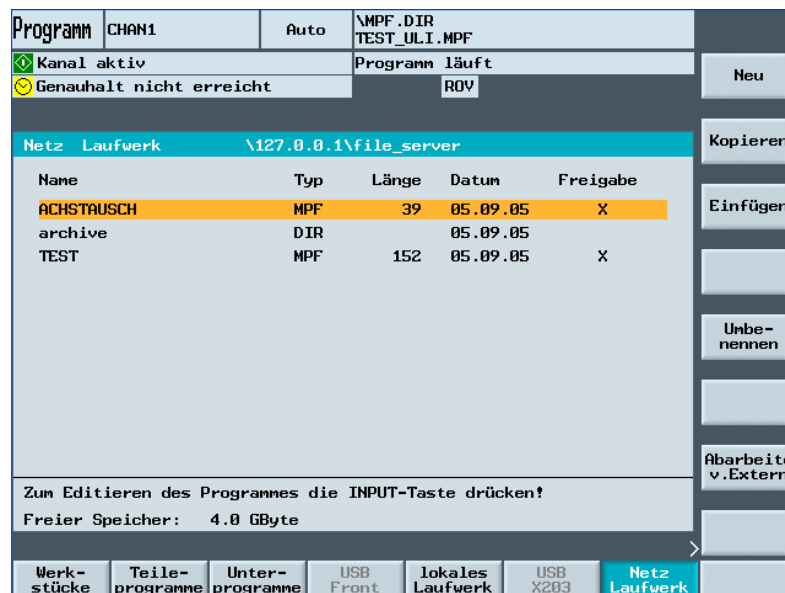


Figure 4-7: Share folder for file transfer

## 4.18 Available display machine data

Please refer to the following documentation, available on DOCONCD, for descriptions of all machine-data displays:

**References:** /AMD/Detailed machine data description

### Legend

#### Target systems:

Adv	HMI Advanced
Emb	HMI-Embedded
HT6	Handheld terminal 6
OP30	Operator panel OP 030
MT	ManualTurn
SM	ShopMill
ST	ShopTurn

#### Filter:

H01	ShopMill
H02	ShopTurn
H03	ManualTurn
H04	Access levels
H05	Standard machine

MD number	Name of identifier			Filter	Cross reference:
Unit	Name			Data type	Read/write protection
PLC	Default value	Min. value	Max. value	Activation	SW version
<b>9000</b>	<b>LCD_CONTRAST</b>			H05	<b>CR: A2</b>
-	Contrast			BYTE	3/4
OP30, Adv, Emb	OP30: 7, Adv: 7, Emb: 7	0	15	Power On	-
<b>9001</b>	<b>DISPLAY_TYPE</b>			H05	<b>CR: A2</b>
-	Type of operator panel 9002 external monitor			BYTE	0/0
OP30, Adv, Emb	OP30: 1, Adv: 1, Emb: 1	0	2	Power On	
<b>9002</b>	<b>DISPLAY_MODE</b>				<b>CR: A2</b>
-	(1: black and white, 2: color)			BYTE	3/4
OP30,	OP30: 0,	0	2	Power On	1
<b>9003</b>	<b>FIRST_LANGUAGE</b>			H05	<b>CR: A2</b>
-	Foreground language			BYTE	3/4
OP30, Emb	OP30: 1, Emb: 1	1	2	Power On	1.1
<b>9004</b>	<b>DISPLAY_RESOLUTION</b>			H05	<b>CR: A2</b>
-	Display resolution			BYTE	3/4
OP30, Adv, Emb	OP30: 3, Adv: 3, Emb: 3	0	5	Power On	-
<b>9005</b>	<b>PRG_DEFAULT_DIR</b>			H05	<b>CR: A2</b>
-	Basic setting Program directory			BYTE	3/4
OP30, Emb	OP30: 1, Emb: 1	1	5	IMMEDIATELY	-

<b>9006</b>	<b>DISPLAY_BLACK_TIME</b>			H05	<b>CR: A2</b>
-	Time for screen darkening			BYTE	3/4
OP30, Emb	OP30: 15, Emb: 15	0	60	Power On	SW 2
<b>9007</b>	<b>TABULATOR_SIZE</b>			H05	<b>CR: A2</b>
-	Tabulator length			BYTE	3/4
OP30, Emb	OP30: 4, Emb: 4	0	30	IMMEDIATE LY	SW 2
<b>9008</b>	<b>KEYBOARD_TYPE</b>			H05	<b>CR: A2</b>
-	Keyboard type (0: OP, 1: MFII/QWERTY)			BYTE	3/4
Adv, Emb	Adv: 0, Emb: 0	0	1	Power On	SW3.6
<b>9009</b>	<b>KEYBOARD_STATE</b>			H05	<b>CR: A2</b>
-	Keyboard shift response during ramp-up (0: single, 2: CAPSLOCK)			BYTE	3/4
Adv, Emb	Adv: 0, Emb: 2	0	2	Power On	SW3.6
<b>9011</b>	<b>DISPLAY_RESOLUTION_INCH</b>			H05	<b>CR: A2</b>
-	Display resolution for INCH system of measurement			BYTE	3/4
Adv, Emb	Adv: 4, Emb: 4	0	6	Power On	SW5.1
<b>9012</b>	<b>ACTION_LOG_MODE</b>			H05	<b>CR: IM2, IM4</b>
-	Set action mode for action log			INTEGER	1/1
Adv, Emb	Adv: 0xFE, Emb: 254	0	0xFFFF	Power On	SW5.2
<b>9013</b>	<b>SYS_CLOCK_SYNC_TIME</b>			H05	<b>CR: IM1</b>
-	Synchronization time for MMC/HMI time with PLC Synchr. time for HMI/PLC time			REAL	0/0
Emb	Emb: 0	0	199	Power On	SW5.3
<b>9014</b>	<b>USE_CHANNEL_DISPLAY_DATA</b>			H05	<b>CR: FBT, FBSP, EMB, ADV</b>
-	Use channel-specific display MDs			INTEGER	3/4
Adv, Emb	Adv: 0, Emb: 0	0	1	IMMEDIATE LY	SW6.3
<b>9015</b>	<b>DARKTIME_TO_PLC</b>				<b>CR: -</b>
-				BOOL	3/4
OP30,	OP30: 0,	***	***	Power On	SW5
<b>9016</b>	<b>SWITCH_TO_AREA</b>			H05	<b>CR: IAM, BE1</b>
-	Default boot menu can be selected			INTEGER	3/4
OP30, Emb	OP30: 20, Emb: 1	-1	10000	Power On	SW6.3
<b>9017</b>	<b>PLC_HOTKEY</b>				<b>CR:</b>
-				STRING	3/4
OP30,	OP30: DB0.DBB0.0,	***	***	Power On	

<b>9020</b>	<b>TECHNOLOGY</b>			H05	<b>CR:</b> A2, FBT
-	Technology for NC prog. and simulation 0: No specific assignment 1: Turning machine configuration <>1: Milling machine configuration			BYTE	3/4
Adv, Emb	Adv: 0, Emb: 1	0	2	Power On	SW6, ST SW6.1
<b>9021</b>	<b>LAYOUT_MODE</b>			H05	<b>CR:</b>
-	Design of the operator interface			BYTE	3/4
Adv, Emb, HT 6	Adv: 1, Emb: 0, HT 6:1	0	1	Power On	SW 6.3
<b>9025</b>	<b>DISPLAY_BACKLIGHT</b>				<b>CR:</b> IM2
-	Display backlight brightness level			BYTE	3/4
	HT6: 15	0	31	Power On	SW 5.3
<b>9026</b>	<b>TEACH_MODE</b>				<b>CR:</b> IM2
-	Teach mode to be activated			REAL	3/4
	HT6: 1	***	***	Power On	SW 5.3
<b>9027</b>	<b>NUM_AX_SEL</b>				<b>CR:</b> IM2
-	Number of axis groups for traversing keys			REAL	3/4
	HT6: 0	0	4	Power On	SW 5.3
<b>9030</b>	<b>EXPONENT_LIMIT</b>			H05	<b>CR:</b> A2
-	Number of digits for display without exponent			BYTE	3/4
Emb	Emb: 6	0	20	Power On	SW 5.1
<b>9031</b>	<b>EXPONENT_SCIENCE</b>			H05	<b>CR:</b> A2
-	Exponent in the technical representation			BYTE	3/4
Emb	Emb: 1	0	1	Power On	SW5.1
<b>9032</b>	<b>HMI_MONITOR</b>			H05	<b>CR:</b> FBT, FBSP, EMB, ADV
-	Determine PLC data for HMI monitor information			STRING	2/4
Adv, Emb	Adv: ", Emb: 0	***	***	Power On	SW6.2
<b>9050</b>	<b>STARTUP_LOGO</b>			H05	<b>CR:</b> FBT, FBSP, EMB, ADV
-	Activate OEM boot screen			BYTE	1/4
Adv, Emb	Adv: 0, Emb: 0	0	1	Power On	SW6.2
<b>9052</b>	<b>SHOW_CHANNEL_SPANNING_STATE</b>			H05	<b>CR:</b> FBT, FBSP
-	Change cross-channel status display			BYTE	2/4
Adv, Emb	Adv: 0, Emb: 0	0	1	Power On	SW6.3
<b>9053</b>	<b>PLC_SYMBOL_SORT</b>			H05	<b>CR:</b>
-	Sorting algorithm for PLC icons			INTEGER	3/4
Adv, Emb	Adv: 0, Emb: 0	0	4	IMMEDIATE LY	SW6.3

<b>9054</b>	<b>PLC_SYMBOL_FILTER</b>			H05	<b>CR:</b>
-	Filter settings for PLC icons			REAL	3/4
Adv, Emb	Adv: 0, Emb: 0	0	0xFFFF	IMMEDIATE LY	SW6.3
<b>9055</b>	<b>PLC_ALARM_PICTURE</b>			H05	<b>CR:</b>
-	Acknowledgement icon selection for PLC alarms			INTEGER	3/4
Adv, Emb	Adv: 1, Emb: 1	-1	1	Power On	SW6.3
<b>9056</b>	<b>ALARM_ROTATION_CYCLE</b>			H05	<b>CR:</b>
-	Rotation cycle time for the alarm display			INTEGER	3/4
Emb	Emb: 0	0	10000	IMMEDIATE LY	SW6.4

<b>9180</b>	<b>USER_CLASS_READ_TCARR</b>			H04, H05	<b>CR: A2</b>
-	Protection level read toolholder offsets			BYTE	3/4
Emb	Emb: 7	0	7	IMMEDIATE LY	SW6.1
<b>9181</b>	<b>USER_CLASS_WRITE_TCARR</b>			H04, H05	<b>CR: A2</b>
-	Protection level write toolholder offsets			BYTE	3/4
Emb	Emb: 7	0	7	IMMEDIATE LY	SW6.1
<b>9182</b>	<b>USER_CLASS_INCH_METRIC</b>			H04, H05	<b>CR: EMB</b>
-	Protection level inch/metric switchover			BYTE	3/4
Emb	Emb: 7	0	7	IMMEDIATE LY	SW6.2
<b>9183</b>	<b>USER_WRITE_TOOLFRAME</b>			H04, H05	<b>CR:</b>
-	Protection level write toolholder			BYTE	3/4
Adv, Emb	Adv: 7, Emb: 0	0	7	IMMEDIATE LY	SW6.4
<b>9184</b>	<b>USER_WRITE_PARTFRAME</b>			H04, H05	<b>CR:</b>
-	Protection level write tool reference point			BYTE	3/4
Adv, Emb	Adv: 7, Emb: 0	0	7	IMMEDIATE LY	SW6.4
<b>9185</b>	<b>USER_WRITE_WPFRAME</b>			H04, H05	<b>CR:</b>
-	Protection level write workpiece reference point			BYTE	3/4
Adv, Emb	Adv: 7, Emb: 0	0	7	IMMEDIATE LY	SW6.4
<b>9186</b>	<b>USER_WRITE_CYCFRAME</b>			H04, H05	<b>CR:</b>
-	Protection level write cycle frame			BYTE	3/4
Adv, Emb	Adv: 7, Emb: 0	0	7	IMMEDIATE LY	SW6.4
<b>9187</b>	<b>USER_WRITE_TRAFRAME</b>			H04, H05	<b>CR:</b>
-	Protection level write transformation frame			BYTE	3/4
Adv, Emb	Adv: 7, Emb: 0	0	7	IMMEDIATE LY	SW6.4
<b>9188</b>	<b>USER_WRITE_EXTRFRAME</b>			H04, H05	<b>CR:</b>
-	Protection level write external WO			BYTE	3/4
Adv, Emb	Adv: 7, Emb: 0	0	7	IMMEDIATE LY	SW6.4



<b>9200</b>	<b>USER_CLASS_READ_TOA</b>			H04, H05	<b>CR: A2</b>
-	Protection level read tool offsets			BYTE	3/4
OP30, Adv, Emb	OP30: 7, Adv: 7, Emb: 7	0	7	IMMEDIATE LY	-
<b>9201</b>	<b>USER_CLASS_WRITE_TOA_GEO</b>			H04, H05	<b>CR: A2</b>
-	Protection level write tool geometry			BYTE	3/4
OP30, Adv, Emb	OP30: 7, Adv: 7, Emb: 7	0	7	IMMEDIATE LY	-
<b>9202</b>	<b>USER_CLASS_WRITE_TOA_WEAR</b>			H04, H05	<b>CR: A2</b>
-	Protection level write tool wear data			BYTE	3/4
OP30, Adv, Emb	OP30: 7, Adv: 7, Emb: 7	0	7	IMMEDIATE LY	-
<b>9203</b>	<b>USER_CLASS_WRITE_FINE</b>			H04, H05	<b>CR: A2</b>
-	Protection level fine			BYTE	3/4
Adv, Emb	Adv: 7, Emb: 7	0	7	IMMEDIATE LY	-
<b>9206</b>	<b>USER_CLASS_WRITE_TOA_SUPVIS</b>			H04, H05	<b>CR: A2</b>
-	Protection level tool monitoring Change limit values fine			BYTE	3/4
Adv, Emb	Adv: 7, Emb: 0	0	7	IMMEDIATE LY	SW 5

<b>9209</b>	<b>USER_CLASS_WRITE_TOA_ADAPT</b>			H04, H05	<b>CR: A2</b>
-	Protection level write tool adapter data			BYTE	3/4
OP30, Adv, Emb	OP30: 7, Adv: 7, Emb: 7	0	7	IMMEDIATE LY	SW5
<b>9210</b>	<b>USER_CLASS_WRITE_ZOA</b>			H04, H05	<b>CR: A2</b>
-	Protection level write settable WO			BYTE	3/4
OP30, Adv, Emb	OP30: 7, Adv: 7, Emb: 7	0	7	IMMEDIATE LY	-
<b>9211</b>	<b>USER_CLASS_READ_GUD_LUD</b>			H04, H05	<b>CR: A2</b>
-	Protection level read user variables			BYTE	3/4
OP30, Adv, Emb	OP30: 7, Adv: 7, Emb: 7	0	7	IMMEDIATE LY	SW6.1
<b>9212</b>	<b>USER_CLASS_WRITE_GUD_LUD</b>			H04, H05	<b>CR: IM2, IM4, A2</b>
-	Protection level write user variables			BYTE	3/4
OP30, Adv, Emb	OP30: 7, Adv: 7, Emb: 7	0	7	IMMEDIATE LY	SW6.1
<b>9213</b>	<b>USER_CLASS_OVERSTORE_HIGH</b>			H04, H05	<b>CR: A2</b>
-	Protection level extended overstore			BYTE	3/4
OP30, Adv, Emb	OP30: 7, Adv: 7, Emb: 7	0	7	IMMEDIATE LY	-
<b>9214</b>	<b>USER_CLASS_WRITE_PRG_CONDIT</b>			H04, H05	<b>CR: A2</b>
-	Protection level program control			BYTE	3/4
OP30, Adv, Emb	OP30: 7, Adv: 7, Emb: 7	0	7	IMMEDIATE LY	-
<b>9215</b>	<b>USER_CLASS_WRITE_SEA</b>			H04, H05	<b>CR: A2</b>
-	Protection level write setting data			BYTE	3/4
OP30, Adv, Emb	OP30: 7, Adv: 7, Emb: 7	0	7	IMMEDIATE LY	-
<b>9216</b>	<b>USER_CLASS_READ_PROGRAM</b>			H04, H05	<b>CR: A2</b>
-	Protection level read parts program			BYTE	3/4
OP30, Adv, Emb	OP30: 7, Adv: 7, Emb: 7	0	7	IMMEDIATE LY	-
<b>9217</b>	<b>USER_CLASS_WRITE_PROGRAM</b>			H04, H05	<b>CR: A2</b>
-	Protection level write parts program			BYTE	3/4
OP30, Adv, Emb	OP30: 7, Adv: 7, Emb: 7	0	7	IMMEDIATE LY	-

<b>9218</b>	<b>USER_CLASS_SELECT_PROGRAM</b>			H04, H05	<b>CR: A2</b>
-	Protection level 'Program selection'			BYTE	3/4
OP30, Adv, Emb	OP30: 7, Adv: 7, Emb: 7	0	7	IMMEDIATE LY	-
<b>9219</b>	<b>USER_CLASS_TEACH_IN</b>			H04, H05	<b>CR: A2</b>
-	Protection level TEACH IN			BYTE	3/4
OP30, Adv, Emb	OP30: 7, Adv: 7, Emb: 7	0	7	IMMEDIATE LY	-
<b>9220</b>	<b>USER_CLASS_PRESET</b>			H04, H05	<b>CR: A2</b>
-	Protection level PRESET			BYTE	3/4
OP30, Adv, Emb	OP30: 7, Adv: 7, Emb: 7	0	7	IMMEDIATE LY	-
<b>9221</b>	<b>USER_CLASS_CLEAR_RPA</b>			H04, H05	<b>CR: A2</b>
-	Protection level delete R parameters			BYTE	3/4
OP30, Adv, Emb	OP30: 7, Adv: 7, Emb: 7	0	7	IMMEDIATE LY	-
<b>9222</b>	<b>USER_CLASS_WRITE_RPA</b>			H04, H05	<b>CR: A2</b>
-	Protection level write R parameters			BYTE	3/4
OP30, Adv, Emb	OP30: 7, Adv: 7, Emb: 7	0	7	IMMEDIATE LY	-
<b>9223</b>	<b>USER_CLASS_SET_V24</b>			H04, H05	<b>CR: A2</b>
-	Protection level RS-232 interface parameterization			BYTE	3/4
OP30, Emb	OP30: 7, Emb: 7	0	7	IMMEDIATE LY	-
<b>9224</b>	<b>USER_CLASS_READ_IN</b>			H04, H05	<b>CR: A2</b>
-	Protection level read in data			BYTE	3/4
OP30, Emb	OP30: 7, Emb: 7	0	7	IMMEDIATE LY	-
<b>9225</b>	<b>USER_CLASS_READ_CST</b>			H04, H05	<b>CR: A2</b>
-	Protection level standard cycles			BYTE	3/4
OP30, Emb	OP30: 7, Emb: 7	0	7	IMMEDIATE LY	SW 2
<b>9226</b>	<b>USER_CLASS_READ_CUS</b>			H04, H05	<b>CR: A2</b>
-	Protection level user cycles			BYTE	3/4
OP30, Emb	OP30: 7, Emb: 7	0	7	IMMEDIATE LY	SW 2
<b>9227</b>	<b>USER_CLASS_SHOW_SBL2</b>			H04, H05	<b>CR: A2</b>
-	Skip single block2 (SBL2)			BYTE	3/4
Emb	Emb: 7	0	7	IMMEDIATE LY	SW3.5
<b>9228</b>	<b>USER_CLASS_READ_SYF</b>			H04, H05	<b>CR: A2</b>
-	Access level for selecting the directory SYF			BYTE	3/4
Emb	Emb: 7	0	7	IMMEDIATE LY	SW4.2
<b>9229</b>	<b>USER_CLASS_READ_DEF</b>			H04, H05	<b>CR: A2</b>
-	Access level for selecting the directory DEF			BYTE	3/4
Emb	Emb: 7	0	7	IMMEDIATE LY	SW4.2
<b>9230</b>	<b>USER_CLASS_READ_BD</b>			H04, H05	<b>CR: A2</b>
-	Access level for selecting the directory BD			BYTE	3/4
Emb	Emb: 3	0	7	IMMEDIATE LY	SW4.2

<b>9242</b>	<b>MA_STAT_DISPLAY_BASE</b>			H05	<b>CR: K2</b>
-	Numerical basis for display of articulated position STAT			WORD	3/4
Adv, Emb	Adv: 10, Emb: 0	0	16	IMMEDIATE LY	SW6.1
<b>9243</b>	<b>MA_TU_DISPLAY_BASE</b>			H05	<b>CR: K2</b>
-	Numerical basis for display of rotary axis position TU			WORD	3/4
Adv, Emb	Adv: 10, Emb: 0	0	16	IMMEDIATE LY	SW6.1

<b>9246</b>	<b>USER_CLASS_SYS_ZERO_OFF</b>			H04, H05	<b>CR:</b>
-	Access level for writing system frames			BYTE	2/2
Adv, Emb	Adv: 7, Emb: 7	0	7	IMMEDIATE LY	Adv. SW6.3, Emb. SW6.2
<b>9247</b>	<b>USER_CLASS_BASE_ZERO_OFF_PA</b>			H04, H05	<b>CR: K2</b>
-	Access level for basic offset PA			BYTE	2/2
Adv, Emb	Adv: 7, Emb: 7	0	7	IMMEDIATE LY	SW5.3
<b>9248</b>	<b>USER_CLASS_BASE_ZERO_OFF_MA</b>			H04, H05	<b>CR: IAM, IM1</b>
-	Access level for basic offset MA			BYTE	2/2
Adv, Emb	Adv: 7, Emb: 7	0	7	IMMEDIATE LY	SW5.3
<b>9249</b>	<b>USER_CLASS_VERT_MODE_SK</b>			H04, H05	<b>CR: K2</b>
-	Protection level for vertical SKs of area SKs Protection for vertical SKs			DOUBLE	3/4
Emb	Emb: 2004318071	0	0x77777777	IMMEDIATE LY	SW6.1
<b>9251</b>	<b>USER_CLASS_TM_SKTLLIST</b>			H04, H05	<b>CR: FBW</b>
-	Display tool list			BYTE	3/4
Emb	Emb: 7	0	7	Power On	S 4.1
<b>9252</b>	<b>USER_CLASS_TM_SKTOOLLOAD</b>			H04, H05	<b>CR: FBW</b>
-	Protection level for loading tools			BYTE	3/4
Emb	Emb: 7	0	7	Power On	SW4.1
<b>9253</b>	<b>USER_CLASS_TM_SKTOOLUNLOAD</b>			H04, H05	<b>CR: FBW</b>
-	Protection level for unloading tools			BYTE	3/4
Emb	Emb: 7	0	7	Power On	SW4.1
<b>9254</b>	<b>USER_CLASS_TM_SKTOOLMOVE</b>			H04, H05	<b>CR: FBW</b>
-	Protection level for relocating tools			BYTE	3/4
Emb	Emb: 7	0	7	Power On	SW4.1
<b>9256</b>	<b>USER_CLASS_TM_SKMGLREPR2</b>			H04, H05	<b>CR: FBW</b>
-	Protection level for displaying 2nd magazine list			BYTE	3/4
Emb	Emb: 7	0	7	Power On	SW4.1
<b>9257</b>	<b>USER_CLASS_TM_SKMGLREPR3</b>			H04, H05	<b>CR: FBW</b>
-	Protection level for displaying 3rd magazine list			BYTE	3/4
Emb	Emb: 7	0	7	Power On	SW4.1

<b>9258</b>	<b>USER_CLASS_TM_SKNCNEWTTOOLE</b>			H04, H05	<b>CR: FBW</b>
-	Protection level for creating new tool edges			BYTE	3/4
Emb	Emb: 7	0	7	Power On	SW4.1
<b>9259</b>	<b>USER_CLASS_TM_SKNCDELTOOL</b>			H04, H05	<b>CR: FBW</b>
-	Protection level for deleting tools			BYTE	3/4
Emb	Emb: 7	0	7	Power On	SW4.1
<b>9260</b>	<b>USER_CLASS_TM_SKMGBUFFER</b>			H04, H05	<b>CR: FBW</b>
-	Protection level for buffer ON/OFF			BYTE	3/4
Emb	Emb: 7	0	7	Power On	SW4.1
<b>9261</b>	<b>USER_CLASS_TM_SKMGFIND</b>			H04, H05	<b>CR: FBW</b>
-	Protection level for search			BYTE	3/4
Emb	Emb: 7	0	7	Power On	SW4.1
<b>9262</b>	<b>USER_CLASS_TM_SKMGLISTPOS</b>			H04, H05	<b>CR: FBW</b>
-	Protection level for positioning			BYTE	3/4
Emb	Emb: 7	0	7	Power On	SW4.1
<b>9263</b>	<b>USER_CLASS_TM_SKMGNEXT</b>			H04, H05	<b>CR: FBW</b>
-	Protection level for paging to next magazine			BYTE	3/4
Emb	Emb: 7	0	7	Power On	SW4.1
<b>9264</b>	<b>USER_CLASS_TM_SKTLNEWTOOL</b>			H04, H05	<b>CR: FBW</b>
-	Protection level for creating tools			BYTE	3/4
Emb	Emb: 7	0	7	Power On	SW4.1
<b>9265</b>	<b>USER_CLASS_TM_SKTLLREPR1</b>			H04, H05	<b>CR: FBW</b>
-	Protection level for displaying 1st tool list			BYTE	3/4
Emb	Emb: 7	0	7	Power On	SW4.1
<b>9266</b>	<b>USER_CLASS_TM_SKTLLREPR2</b>			H04, H05	<b>CR: FBW</b>
-	Protection level for displaying 2nd tool list			BYTE	3/4
Emb	Emb: 7	0	7	Power On	SW4.1
<b>9267</b>	<b>USER_CLASS_TM_SKTLLREPR3</b>			H04, H05	<b>CR: FBW</b>
-	Protection level for displaying 3rd tool list			BYTE	3/4
Emb	Emb: 7	0	7	Power On	SW4.1
<b>9269</b>	<b>USER_CLASS_TM_SKFINDPLACE</b>			H04, H05	<b>CR: FBW</b>
-	Softkey empty location, displ. Tool list			BYTE	3/4
Emb	Emb: 7	0	7	Power On	SW4.1
<b>9270</b>	<b>USER_CLASS_TM_SKACTPLACE</b>			H04, H05	<b>CR: FBW</b>
-	Protection level for loading to curr. location			BYTE	3/4
Emb	Emb: 7	0	7	Power On	SW4.1

<b>9271</b>	<b>USER_CLASS_TM_SKLDTOOLDAT</b>			H04, H05	<b>CR: FBW</b>
-	Viewing and editing tool data			BYTE	3/4
Emb	Emb: 7	0	7	Power On	SW4.1
<b>9272</b>	<b>USER_CLASS_APPLICATION</b>			H04, H05	<b>CR:</b>
-	Protection level for selecting the operating area			BYTE	3/4
Emb	Emb: 7	0	7	IMMEDIATE LY	SW6.4
<b>9300</b>	<b>V24_USER_XON</b>			H05	<b>CR: C4</b>
-	User: X on character			REAL	3/4
OP30, Emb	OP30: 17, Emb: 17	0	0xFF	IMMEDIATE LY	-
<b>9301</b>	<b>V24_USER_XOFF</b>			H05	<b>CR: C4</b>
-	User: X off character			REAL	3/4
OP30, Emb	OP30: 19, Emb: 19	0	0xFF	IMMEDIATE LY	-
<b>9302</b>	<b>V24_USER_EOF</b>			H05	<b>CR: C4</b>
-	User: End of transmission character			REAL	3/4
OP30, Emb	OP30: 26, Emb: 26	0	0xFF	IMMEDIATE LY	-
<b>9303</b>	<b>V24_USER_CONTROLS</b>			H05	<b>CR: C4</b>
-	User: Special bits			REAL	3/4
OP30, Emb	OP30: 76, Emb: 76	0	0x3FF	IMMEDIATE LY	-
<b>9304</b>	<b>V24_USER_RTS</b>			H05	<b>CR: C4</b>
-	User: Line-controlled			BYTE	3/4
OP30, Emb	OP30: 1, Emb: 0	0	1	IMMEDIATE LY	-
<b>9305</b>	<b>V24_USER_BAUD</b>			H05	<b>CR: C4</b>
-	User: Baud rate (300, 600, 1200, 2400, 4800, 9600, 19200) 0 1 ...			BYTE	3/4
OP30, Emb	OP30: 4, Emb: 5	0	8	IMMEDIATE LY	-
<b>9306</b>	<b>V24_USER_DATABITS</b>			H05	<b>CR: C4</b>
-	User: Data bits			BYTE	3/4
OP30, Emb	OP30: 1, Emb: 1	0	1	IMMEDIATE LY	-
<b>9307</b>	<b>V24_USER_PARITY</b>			H05	<b>CR: C4</b>
-	User: Parity bits			BYTE	3/4
OP30, Emb	OP30: 0, Emb: 0	0	2	IMMEDIATE LY	-
<b>9308</b>	<b>V24_USER_STOPBIT</b>			H05	<b>CR: C4</b>
-	User: Stop bits			BYTE	3/4
OP30, Emb	OP30: 0, Emb: 0	0	1	IMMEDIATE LY	-
<b>9309</b>	<b>V24_USER_LINE</b>			H05	<b>CR: C4</b>
-	User: RS-232-C interface (COM1/COM2) (COM1/COM2)			BYTE	3/4
Emb	Emb: 1	1	2	IMMEDIATE LY	SW5

<b>9310</b>	<b>V24_PRINTER_XON</b>			H05	<b>CR: C4</b>
-	Printer: X on character			REAL	3/4
OP30, Emb	OP30: 17, Emb: 17	0	0xFF	IMMEDIATE LY	-
<b>9311</b>	<b>V24_PRINTER_XOFF</b>			H05	<b>CR: C4</b>
-	Printer: X off character			REAL	3/4
OP30, Emb	OP30: 19, Emb: 19	0	0xFF	IMMEDIATE LY	-
<b>9312</b>	<b>V24_PRINTER_EOF</b>			H05	<b>CR: C4</b>
-	Printer: End of transmission character			REAL	3/4
OP30, Emb	OP30: 12, Emb: 12	0	0xFF	IMMEDIATE LY	-
<b>9313</b>	<b>V24_PRINTER_CONTROLS</b>			H05	<b>CR: C4</b>
-	Printer: Special bits			REAL	3/4
OP30, Emb	OP30: 76, Emb: 76	0	0x3FF	IMMEDIATE LY	-
<b>9314</b>	<b>V24_PRINTER_RTS</b>			H05	<b>CR: C4</b>
-	Printer: Line-controlled			BYTE	3/4
OP30, Emb	OP30: 0, Emb: 0	0	1	IMMEDIATE LY	-
<b>9315</b>	<b>V24_PRINTER_BAUD</b>			H05	<b>CR: C4</b>
-	Printer: Baud rate (300, 600, 1200, 2400, 4800, 9600, 19200) 0 1 ...			BYTE	3/4
OP30, Emb	OP30: 5, Emb: 5	0	8	IMMEDIATE LY	-
<b>9316</b>	<b>V24_PRINTER_DATABITS</b>			H05	<b>CR: C4</b>
-	Printer: Data bits			BYTE	3/4
OP30, Emb	OP30: 1, Emb: 1	0	1	IMMEDIATE LY	-
<b>9317</b>	<b>V24_PRINTER_PARITY</b>			H05	<b>CR: C4</b>
-	Printer: Parity bits			BYTE	3/4
OP30, Emb	OP30: 0, Emb: 0	0	2	IMMEDIATE LY	-
<b>9318</b>	<b>V24_PRINTER_STOPBIT</b>			H05	<b>CR: C4</b>
-	Printer: Stop bits			BYTE	3/4
OP30, Emb	OP30: 0, Emb: 0	0	1	IMMEDIATE LY	-
<b>9319</b>	<b>V24_PRINTER_LINE</b>			H05	<b>CR: C4</b>
-	Printer: RS-232-C interface (COM1/COM2) (COM1/COM2)			BYTE	3/4
Emb	Emb: 1	1	2	IMMEDIATE LY	-
<b>9320</b>	<b>V24_PG_PC_XON</b>			H05	<b>CR: C4</b>
-	PG: X on character 9321 PG: X off character			REAL	3/4
OP30, Emb	OP30: 17, Emb: 17	0	0xFF	IMMEDIATE LY	-
<b>9321</b>	<b>V24_PG_PC_XOFF</b>			H05	<b>CR: C4</b>
-				REAL	3/4
OP30, Emb	OP30: 19, Emb: 19	0	0xFF	IMMEDIATE LY	-

<b>9322</b>	<b>V24_PG_PC_EOF</b>			H05	<b>CR: C4</b>
-	PG: End of transmission character			REAL	3/4
OP30, Emb	OP30: 26, Emb: 26	0	0xFF	IMMEDIATE LY	-
<b>9323</b>	<b>V24_PG_PC_CONTROLS</b>			H05	<b>CR: C4</b>
-	PG: Special bits 9324 PG: Line-controlled			REAL	3/4
OP30, Emb	OP30: 144, Emb: 144	0	0x3FF	IMMEDIATE LY	-
<b>9324</b>	<b>V24_PG_PC_RTS</b>			H05	<b>CR: C4</b>
-				BYTE	3/4
OP30, Emb	OP30: 0, Emb: 0	0	1	IMMEDIATE LY	-
<b>9325</b>	<b>V24_PG_PC_BAUD</b>			H05	<b>CR: C4</b>
-	PG: Baud rate (300, 600, 1200, 2400, 4800, 9600) 0 1 ...			BYTE	3/4
OP30, Emb	OP30: 5, Emb: 5	0	8	IMMEDIATE LY	-
<b>9326</b>	<b>V24_PG_PC_DATABITS</b>			H05	<b>CR: C4</b>
-	PG: Data bits			BYTE	3/4
OP30, Emb	OP30: 1, Emb: 1	0	1	IMMEDIATE LY	-
<b>9327</b>	<b>V24_PG_PC_PARITY</b>			H05	<b>CR: C4</b>
-	PG: Parity bits			BYTE	3/4
OP30, Emb	OP30: 0, Emb: 0	0	2	IMMEDIATE LY	-
<b>9328</b>	<b>V24_PG_PC_STOPBIT</b>			H05	<b>CR: C4</b>
-	PG: Stop bits			BYTE	3/4
OP30, Emb	OP30: 0, Emb: 0	0	1	IMMEDIATE LY	-
<b>9329</b>	<b>V24_PG_PC_LINE</b>			H05	<b>CR: C4</b>
-	PG: RS-232-C interface (COM1/COM2) (COM1/COM2)			BYTE	3/4
Emb	Emb: 1	1	2	IMMEDIATE LY	-
<b>9400</b>	<b>TOOL_REF_GEO_AXIS1</b>			H05	<b>CR: BO</b>
-	Absolute dimension tool length compensation GEOaxis.1			DOUBLE	3/4
OP30, Emb	OP30: 0, Emb: 0	***	***	IMMEDIATE LY	-
<b>9401</b>	<b>TOOL_REF_GEO_AXIS2</b>			H05	<b>CR: BO</b>
-	Absolute dimension tool length compensation GEOaxis.2			DOUBLE	3/4
OP30, Emb	OP30: 0, Emb: 0	***	***	IMMEDIATE LY	-
<b>9402</b>	<b>TOOL_REF_GEO_AXIS3</b>			H05	<b>CR: BO</b>
-	Absolute dimension tool length compensation GEOaxis.3			DOUBLE	3/4
OP30, Emb	OP30: 0, Emb: 0	***	***	IMMEDIATE LY	-

<b>9410</b>	<b>TM_LOAD_PLACE</b>			H05	<b>CR: BO</b>
-	Number of loading station			INTEGER	3/4
OP30, Emb	OP30: 0, Emb: 0	***	***	Power On	-
<b>9411</b>	<b>TM_NUM_MAG</b>			H05	<b>CR: BO</b>
-	Number of work magazine			INTEGER	3/4
OP30, Emb	OP30: 0, Emb: 0	***	***	Power On	-
<b>9412</b>	<b>TM_DEFAULT_TOOLSIZE</b>			H05	<b>CR: FBW</b>
-	Default for tool size			REAL	3/4
Emb	Emb: 1111	1111	7777	IMMEDIATE LY	SW4.1
<b>9414</b>	<b>TM_KIND_OF_TOOLMANAGEMENT</b>			H01, H02, H05	<b>CR: FBW</b>
-	Type of representation of the tool management 0: Old, 1: New (SW 5.2 and higher)			BYTE	3/4
SM, ST, Emb	SM: 1, ST: 1, Emb: 0	0	1	Power On	SW5
<b>9415</b>	<b>TM_DEFAULT_TOOLPLACESPEC</b>			H05	<b>CR: FBW</b>
-	Default value for location type			BYTE	3/4
Emb	Emb: 1	1	99	IMMEDIATE LY	SW4.2
<b>9416</b>	<b>TM_DEFAULT_TOOLTYPE</b>			H05	<b>CR: FBW</b>
-	Default for location type			REAL	3/4
Emb	Emb: 120	100	900	IMMEDIATE LY	SW4.1
<b>9417</b>	<b>TM_DEFAULT_TOOLSTATE</b>			H05	<b>CR: FBW</b>
-	Default for tool status load			INTEGER	3/4
Emb	Emb: 2	0	255	IMMEDIATE LY	SW4.1
<b>9419</b>	<b>TM_DEFAULT_DELETE_TOOL</b>			H05	<b>CR: FBW</b>
-	Default for tool data auto. Delete			BYTE	3/4
Emb	Emb: 0	0	1	IMMEDIATE LY	SW4.1
<b>9420</b>	<b>MA_ONLY_MKS_DIST_TO_GO</b>			H05	<b>CR: FBW</b>
-	Distance-to-go display in the Work window 0: Work value 1: Machine value			BYTE	3/4
Emb	Emb: 0	0	1	IMMEDIATE LY	SW4.1
<b>9421</b>	<b>MA_AXES_SHOW_GEO_FIRST</b>			H05	<b>CR: K1</b>
-	Actual-value display with leading geo axes			BYTE	3/4
Adv, Emb	Adv: 1, Emb: 1	0	1	IMMEDIATE LY	SW 2
<b>9422</b>	<b>MA_PRESET_MODE</b>			H05	<b>CR: K1</b>
-	Selecting PRESET/base offset in JOG 0 No Preset, no preset actual value memory 1 PRESET 2 Preset actual value memory 3 Preset actual value memory s. Online help			BYTE	3/4
Adv, Emb	Adv: 1, Emb: 1	0	3	IMMEDIATE LY	SW5



<b>9423</b>	<b>MA_MAX_SKP_LEVEL</b>			H05	<b>CR: K1</b>
-	Max. skip levels in NC program			BYTE	3/4
Adv, Emb	Adv: 1, Emb: 1	1	8	Power On	SW5
<b>9424</b>	<b>MA_COORDINATE_SYSTEM</b>			H05	<b>CR: K2</b>
-	Coord. system for actual-value display 0: Work (WCS) 1: SZS (settable zero system)			BYTE	3/4
Adv, Emb	Adv: 0, Emb: 0	0	1	Power On	SW5
<b>9425</b>	<b>MA_SCRATCH_DEFAULT_MODE</b>			H05	<b>CR: K2</b>
-	Tool offset calculation scratching			DOUBLE	3/4
Emb	Emb: 0	0	2236962	IMMEDIATE LY	5.3

<b>9426</b>	<b>MA_AX_DRIVELOAD_FROM_PL1</b>			H01, H02, H05	<b>CR:</b>
-	Machine-axis index, analog spindle power-rating display			BYTE	3/4
Emb	Emb: 0	0	31	Power On	Emb 6.5
<b>9427</b>	<b>MA_AX_DRIVELOAD_FROM_PL2</b>			H01, H02, H05	<b>CR:</b>
-	Machine-axis index, analog spindle power-rating display			BYTE	3/4
Emb	Emb: 0	0	31	Power On	Emb 6.5
<b>9428</b>	<b>MA_SPIND_MAX_POWER</b>			H01, H02, H05	<b>CR: IAM4</b>
-	Maximum value of spindle display			REAL	3/4
Emb	Emb: 100	100	0xFFFF	Power On	Emb 6.5
<b>9429</b>	<b>MA_SPIND_POWER_RANGE</b>			H01, H02, H05	<b>CR: IAM4</b>
-	Maximum value of spindle display			REAL	3/4
Emb	Emb: 100	100	0xFFFF	Power On	Emb 6.5

<b>9430</b>	<b>TM_UNLOAD_AND_DELETE</b>				<b>CR: FBO</b>
-				BOOL	3/4
OP30,	OP30: 0,	***	***	Power On	SW5
<b>9431</b>	<b>TM_LOAD_TOOL_NEW</b>				<b>CR: FBO</b>
-				BOOL	3/4
OP30,	OP30: 1,	***	***	Power On	SW5
<b>9432</b>	<b>TM_TOOL_STATE_DEF_VAL</b>				<b>CR:</b>
-				BYTE	3/4
OP30,	OP30: 2,	***	***	Power On	
<b>9433</b>	<b>TM_ACT_SEARCH_AND_POS</b>				<b>CR:</b>
-				BOOL	3/4
OP30,	OP30: 0,	***	***	Power On	
<b>9434</b>	<b>TM_LOAD_LOC1</b>				<b>CR: FBW</b>
-	0: Autom. determination for one magazine			INTEGER	3/7
OP30,	OP30: 0,	***	***	Power On	SW4.2

<b>9435</b>	<b>TM_LOAD_LOC2</b>				<b>CR: FBW</b>
-	0: Autom. determination for one magazine			INTEGER	3/7
OP30,	OP30: 0,	***	***	Power On	SW4.2
<b>9436</b>	<b>TM_LOAD_LOC3</b>				<b>CR: FBW</b>
-	0: Autom. determination for one magazine			INTEGER	3/7
OP30,	OP30: 0,	***	***	Power On	SW4.2
<b>9437</b>	<b>TM_LOAD_LOC4</b>				<b>CR: FBW</b>
-	0: Autom. determination for one magazine			INTEGER	3/7
OP30,	OP30: 0,	***	***	Power On	SW4.2
<b>9438</b>	<b>TM_LOAD_LOC5</b>				<b>CR: FBW</b>
-	0: Autom. determination for one magazine			INTEGER	3/7
OP30,	OP30: 0,	***	***	Power On	SW4.2
<b>9440</b>	<b>ACTIVATE_SEL_USER_DATA</b>			H05	<b>CR: K2</b>
-	Activating the active wear immediately			BYTE	3/4
Adv, Emb	Adv: 1, Emb: 0	0	1	IMMEDIATE LY	SW4.3
<b>9442</b>	<b>MA_AUXFU_GROUPS</b>			H01, H02, H05	<b>CR:</b>
-	Displayed auxiliary function groups			STRING	3/7
Emb,	Emb: 0	***	***	Power On	SW4.2
<b>9450</b>	<b>WRITE_TOA_FINE_LIMIT</b>			H05	<b>CR: K2</b>
mm	Limit value for wear fine			DOUBLE	3/4
Adv, Emb	Adv: 0, Emb: 0.999	***	***	IMMEDIATE LY	SW4.2
<b>9451</b>	<b>WRITE_ZOA_FINE_LIMIT</b>			H05	<b>CR: K2</b>
mm	Limit value for offset fine			DOUBLE	3/4
Adv, Emb	Adv: 0, Emb: 0.999	***	***	IMMEDIATE LY	SW4.2
<b>9459</b>	<b>PA_ZOA_MODE</b>			H05	<b>CR: K2, IM2</b>
-	Display mode of work offset			BYTE	3/4
Emb	Emb: 1	0	1	IMMEDIATE LY	SW 6.1
<b>9460</b>	<b>PROGRAM_SETTINGS</b>			H05	<b>CR: A2</b>
-	Settings in the Program area			INTEGER	3/4
Adv, Emb	Adv: 0, Emb: 5	***	***	IMMEDIATE LY	SW5.1
<b>9461</b>	<b>CONTOUR_END_TEXT</b>			H05	<b>CR: A2</b>
-	String to be added at the end of the contour			STRING	3/4
Emb	Emb: "	***	***	IMMEDIATE LY	SW5.1
<b>9464</b>	<b>MAX_PROGRAMM_SIZE_CHECK</b>			H05	<b>CR:</b>
-	File size, as of which no check is made			INTEGER	3/4
Emb	Emb: 0	***	***	IMMEDIATE LY	SW6.4

<b>9477</b>	<b>TO_TRACE</b>			H01, H02, H05	<b>CR:</b>
-	For internal testing purposes			REAL	3/4
SM, ST, Emb	SM: 0, ST: 0, Emb: 0	0	0xFFFF	Power On	
<b>9478</b>	<b>TO_OPTION_MASK</b>			H01, H02, H05	<b>CR:</b>
-	For internal purposes			INTEGER	2/2
SM, ST, Emb	SM: 1, ST: 1, Emb: 0	0	0xFFFF	Power On	
<b>9479</b>	<b>TO_MAG_PLACE_DISTANCE</b>			H02, H05	<b>CR:</b>
mm	Distance between individual magazine locations			DOUBLE	3/4
ST, Emb	ST: Emb: 0	0	10000	Power On	SW6.3
<b>9480</b>	<b>MA_SIMULATION_MODE</b>			H05	<b>CR:</b>
-	Simulation mode			BYTE	3/4
Adv	Adv: -1	-1	2	IMMEDIATE LY	SW6.4
<b>9481</b>	<b>MA_STAND_SIMULATION_LIMIT</b>			H05	<b>CR:</b>
-	Limit of the standard simulation in KB			INTEGER	3/4
Adv	Adv: 200	200	2000000	IMMEDIATE LY	SW6.4
<b>9500</b>	<b>NC_PROPERTIES</b>			H05	<b>CR: A2</b>
-	NC properties Bit 0: Digital drives Bit 1: Software startup switch Bits 2...4: Reserved			BYTE	3/4
OP30, Emb	OP30: 255, Emb: 255	0	0xFF	IMMEDIATE LY	SW 2
<b>9509</b>	<b>USER_CLASS_DIRECTORY_CHG</b>			H04, H05	<b>CR: FBT, FBSP, EMB</b>
-	Protection level for network configuration			BYTE	3/4
Emb	Emb: 1	0	7	IMMEDIATE LY	SW6.2
<b>9510</b>	<b>USER_CLASS_DIRECTORY1_P</b>			H04, H05	<b>CR: A2</b>
-	Protection level for network drive 1 prog.			BYTE	3/4
Adv, Emb	Adv: 7, Emb: 1	0	7	IMMEDIATE LY	SW6.1
<b>9511</b>	<b>USER_CLASS_DIRECTORY2_P</b>			H04, H05	<b>CR: A2</b>
-	Protection level for network drive 2 prog.			BYTE	3/4
Adv, Emb	Adv: 7, Emb: 1	0	7	IMMEDIATE LY	SW6.1
<b>9512</b>	<b>USER_CLASS_DIRECTORY3_P</b>			H04, H05	<b>CR: A2</b>
-	Protection level for network drive 3 prog.			BYTE	3/4
Adv, Emb	Adv: 7, Emb: 1	0	7	IMMEDIATE LY	SW6.1
<b>9513</b>	<b>USER_CLASS_DIRECTORY4_P</b>			H04, H05	<b>CR: A2</b>
-	Protection level for network drive 4 prog.			BYTE	3/4
Adv, Emb	Adv: 7, Emb: 1	0	7	IMMEDIATE LY	6.1

<b>9516</b>	<b>USER_CLASS_DIRECTORY1_M</b>			H04, H05	<b>CR: A2</b>
-	Protection level for network drive 1 mach.			BYTE	3/4
Adv, Emb	Adv: 7, Emb: 0	0	7	IMMEDIATE LY	6.1
<b>9517</b>	<b>USER_CLASS_DIRECTORY2_M</b>			H04, H05	<b>CR: A2</b>
-	Protection level for network drive 2 mach.			BYTE	3/4
Adv, Emb	Adv: 7, Emb: 0	0	7	IMMEDIATE LY	6.1
<b>9518</b>	<b>USER_CLASS_DIRECTORY3_M</b>			H04, H05	<b>CR: A2</b>
-	Protection level for network drive 3 mach.			BYTE	3/4
Adv, Emb	Adv: 7, Emb: 0	0	7	IMMEDIATE LY	6.1
<b>9519</b>	<b>USER_CLASS_DIRECTORY4_M</b>			H04, H05	<b>CR: A2</b>
-	Protection level for network drive 4 mach.			BYTE	3/4
Adv, Emb	Adv: 7, Emb: 0	0	7	IMMEDIATE LY	6.1
<b>9600</b>	<b>CTM_SIMULATION_DEF_X</b>			H01, H02, H03, H05	<b>CR: FBMA, FBSP</b>
-	Simulation default value for X			INTEGER	3/4
SM, ST, MT, Emb	SM: 0, ST: 0, MT: 0, Emb: 0	-10000	10000	Power On	SW2.1 (810D), 4.3 (840D)
<b>9601</b>	<b>CTM_SIMULATION_DEF_Y</b>			H01, H02, H03, H05	<b>CR: FBMA, FBSP</b>
-	Simulation default value for Z			INTEGER	3/4
SM, ST, MT, Emb	SM: 0, ST: 0, MT: 0, Emb: 0	-10000	10000	Power On	SW2.1 (810D), 4.3 (840D)
<b>9602</b>	<b>CTM_SIMULATION_DEF_VIS_AREA</b>			H01, H02, H03, H05	<b>CR: FBMA, FBSP</b>
-	Simulation default value for display area			INTEGER	3/4
SM, ST, MT, Emb	SM: 100, ST: 100, MT: 100, Emb: 100	-10000	10000	Power On	SW2.1 (810D), 4.3 (840D)
<b>9603</b>	<b>CTM_SIMULATION_MAX_X</b>			H01, H02, H03, H05	<b>CR: FBMA, FBSP</b>
-	Simulation of maximum display X			INTEGER	3/4
SM, ST, MT, Emb	SM: 0, ST: 0, MT: 0, Emb: 0	-10000	10000	Power On	SW2.1 (810D), 4.3 (840D)

<b>9604</b>	<b>CTM_SIMULATION_MAX_Y</b>			H01, H02, H03, H05	<b>CR:</b> FBMA, FBSP
-	Simulation maximum display Z			INTEGER	3/4
SM, ST, MT, Emb	SM: 0, ST: 0, MT: 0, Emb: 0	-10000	10000	Power On	840D SW 4.3, 810D SW 2.1
<b>9605</b>	<b>CTM_SIMULATION_MAX_VIS_AREA</b>			H01, H02, H03, H05	<b>CR:</b> FBMA, FBSP
-	Simulation of maximum display area			INTEGER	3/4
SM, ST, MT, Emb	SM: 1000, ST: 1000, MT: 1000, Emb: 1000	-10000	10000	Power On	840D SW4.3, 810D SW2.1
<b>9606</b>	<b>CTM_SIMULATION_TIME_NEW_POS</b>			H01, H02, H03, H05	<b>CR:</b> FBMA, FBT
-	Simulation updating rate of actual value			INTEGER	3/4
SM, ST, MT, Emb	SM: 250, ST: 350, MT: 250, Emb: 100	0	4000	Power On	840D SW4.3, 810D SW2.1, ST SW6.1

<b>9610</b>	<b>CTM_POS_COORDINATE_SYSTEM</b>			H03, H05	<b>CR:</b> FBMA
-	Position of coordinate system for turning			BYTE	3/4
MT, Adv, Emb	MT: 2, Adv: 2, Emb: 2	0	7	IMMEDIATE LY	840D SW4.3, 810D SW2.1
<b>9611</b>	<b>CTM_CROSS_AX_DIAMETER_ON</b>			H02, H03, H05	<b>CR:</b> FBMA, FBT
-	Diameter display for active transverse axes			BYTE	3/4
ST, MT, Emb	ST: 1, MT: 1, Emb: 1	0	1	IMMEDIATE LY	840D SW4.3, 810D SW2.1, ST 6.1
<b>9619</b>	<b>CTM_G91_DIAMETER_ON</b>			H02, H03, H05	<b>CR:</b> FBMA, FBT
-	Incremental infeed			BYTE	3/4
ST, MT, Emb	ST: 0, MT: 0, Emb: 1	0	1	IMMEDIATE LY	840D SW4.3, 810D SW2.1, ST SW6.1
<b>9626</b>	<b>CTM_TRACE</b>			H02, H03, H05	<b>CR:</b> FBMA
-	Test flags internal ManualTurn diagnostics			REAL	3/4
ST, MT, Emb	ST: 1, MT: 1, Emb: 0	0	0xFFFF	IMMEDIATE LY	840D SW4.3, 810D SW2.1

<b>9632</b>	<b>CTM_ANGLE_REFERENCE_AXIS</b>			H03, H05	<b>CR:</b> FBMA
-	Angle reference axis 1: 1. Axis 2: 2. Axis			REAL	3/4
MT, Emb	MT: 1, Emb: 1	0	1	IMMEDIA TELY	840D SW4.4, 810D SW2.4

<b>9650</b>	<b>CMM_POS_COORDINATE_SYSTEM</b>			H01, H02, H05	<b>CR:</b> FBSP, FBT
-	Position of coordinate system			BYTE	3/4
SM, ST, Adv, Emb	SM: 0, ST: 34, Adv: 0, Emb: 0	0	47	IMMEDIA TELY	SW4.3, ST SW6.1
<b>9651</b>	<b>CMM_TOOL_MANAGEMENT</b>			H01, H02, H05	<b>CR:</b> FBSP, FBT
-	Tool management strategy			BYTE	3/4
SM, ST, Adv, Emb	SM: 4, ST: 4, Adv: 4, Emb: 4	1	4	Power On	SW6.1, ST SW6.1
<b>9652</b>	<b>CMM_TOOL_LIFE_CONTROL</b>			H01, H02, H05	<b>CR:</b> FBSP, FBT
-	Tool monitoring			BYTE	3/4
SM, ST, Adv, Emb	SM: 1, ST: 1, Adv: 1, Emb: 1	0	1	Power On	SW6.1
<b>9661</b>	<b>CMM_ENABLE_CUSTOMER_M_CODES</b>			H01, H02, H05	<b>CR:</b> FBSP
-	Input of customized M commands			BYTE	3/4
SM, ST, Adv, Emb	SM: 0, ST: 0, Adv: 4, Emb: 0	0	4	Power On	840D SW4.3, 810D SW2.3
<b>9663</b>	<b>CMM_TOOL_DISPLAY_IN_DIAM</b>			H01, H02, H05	<b>CR:</b> FBSP, FBT
-	Display radius/diameter for tool			BYTE	3/4
SM, ST, Adv, Emb	SM: 1, ST: 1, Adv: 1, Emb: 1	0	1	Power On	840D SW4.3, 810D SW2.3, ST SW6.1
<b>9672</b>	<b>CMM_FIXED_TOOL_PLACE</b>			H01, H02, H05	<b>CR:</b> FBSP, FBT
-	Fixed location coding			BYTE	3/4
SM, ST, Adv, Emb	SM: 0, ST: 1, Adv: 0, Emb: 0	0	1	Power On	840D SW4.4, 810D SW2.4, ST SW6.1
<b>9673</b>	<b>CMM_TOOL_LOAD_STATION</b>			H01, H02, H05	<b>CR:</b> FBSP, FBT
-	Number of loading station			BYTE	3/4
SM, ST, Adv, Emb	SM: 1, ST: 1, Adv: 1, Emb: 1	1	2	Power On	840D SW4.4, 810D SW2.4, ST SW6.1
<b>9674</b>	<b>CMM_ENABLE_TOOL_MAGAZINE</b>			H01, H02, H05	<b>CR:</b> FBSP, FBT
-	Display of magazine list			BYTE	3/4
SM, ST, Adv, Emb	SM: 1, ST: 1, Adv: 1, Emb: 1	0	1	Power On	840D SW4.4, 810D SW2.4, ST SW6.1
<b>9676</b>	<b>CMM_DIRECTORY_SOFTKEY_PATH1</b>			H01, H02, H05	<b>CR:</b> FBSP, FBT
-	Path to the drive names in the directory man.			STRING	3/4
SM, ST, Adv, Emb	SM: ", ST: ", Adv: ", Emb: 0	***	***	Power On	840D SW4.4, 810D SW2.4 with SM

<b>9677</b>	<b>CMM_DIRECTORY_SOFTKEY_PATH2</b>		H01, H02, H05	<b>CR:</b> FBSP, FBT
-	Path to the drive names in the directory man.		STRING	3/4
SM, ST, Adv, Emb	SM: ", ST: ", Adv: ", Emb: 0	***	Power On	840D SW4.4, 810D SW2.4 with SM
<b>9678</b>	<b>CMM_DIRECTORY_SOFTKEY_PATH3</b>		H01, H02, H05	<b>CR:</b> FBSP, FBT
-	Path to the drive names in the directory man.		STRING	3/4
SM, ST, Adv, Emb	SM: ", ST: ", Adv: ", Emb: 0	***	Power On	840D SW4.4, 810D SW2.4 with SM
<b>9679</b>	<b>CMM_DIRECTORY_SOFTKEY_PATH4</b>		H01, H02, H05	<b>CR:</b> FBSP, FBT
-	Path to the drive names in the directory man.		STRING	3/4
SM, ST, Adv, Emb	SM: ", ST: ", Adv: ", Emb: 0	***	Power On	840D SW4.4, 810D SW2.4 with SM

<b>9900</b>	<b>MD_TEXT_SWITCH</b>		H05	<b>CR:</b> -
-	Plaintext instead of MD identifier		BOOL	3/4
OP30, Adv, Emb	OP30: 0, Adv: 0, Emb: 0	***	IMMEDIA TELY	SW 2
<b>9950</b>	<b>MD_NC_TEA_FILTER</b>		H05	<b>CR:</b>
-	Display options general machine data		INTEGER	0/0
Emb	Emb: 67108865	***	Power On	
<b>9951</b>	<b>MD_NC_TEA_IDX_LIMIT</b>		H05	<b>CR:</b>
-	Index filter for general machine data		INTEGER	0/0
Emb	Emb: 0	***	Power On	
<b>9952</b>	<b>MD_AX_TEA_FILTER</b>		H05	<b>CR:</b>
-	Display options axis machine data		INTEGER	0/0
Emb	Emb: 67108865	***	Power On	
<b>9952</b>	<b>MD_AX_TEA_FILTER</b>		H05	<b>CR:</b>
-	Display options axis machine data		INTEGER	0/0
Emb	Emb: 67108865	***	Power On	
<b>9953</b>	<b>MD_AX_TEA_IDX_LIMIT</b>		H05	<b>CR:</b>
-	Index filter for axis machine data		INTEGER	0/0
Emb	Emb: 0	***	Power On	
<b>9954</b>	<b>MD_CH_TEA_FILTER</b>		H05	<b>CR:</b>
-	Display options channel machine data		INTEGER	0/0
Emb	Emb: 33554433	***	Power On	
<b>9955</b>	<b>MD_CH_TEA_IDX_LIMIT</b>		H05	<b>CR:</b>
-	Index filter for channel machine data		INTEGER	0/0
Emb	Emb: 0	***	Power On	
<b>9956</b>	<b>MD_DRV_TEA_FILTER</b>		H05	<b>CR:</b>
-	Display options drive machine data		INTEGER	0/0
Emb	Emb: 8388609	***	Power On	
<b>9957</b>	<b>MD_DRV_TEA_IDX_LIMIT</b>		H05	<b>CR:</b>
-	Index filter for drive machine data		INTEGER	0/0
Emb	Emb: 0	***	Power On	

<b>9980</b>	<b>LANGUAGE_SETTINGS</b>			H05	<b>CR:</b>
-	Internal language settings			INTEGER	0/0
Emb	Emb: 513	***	***	Power On	
<b>9990</b>	<b>SW_OPTIONS</b>			H05	<b>CR:</b> FBSP, FBT
-	Enable MMC/HMI SW options			INTEGER	2/2
Adv, Emb	Adv: 0, Emb: 0	***	***	Power On	SW5.3
<b>9991</b>	<b>HMI_HELP_SYSTEMS</b>			H05	<b>CR:</b> FBSP, FBT
-	Enable MMC/HMI help systems Bit 0 = 1 Help for calculator enabled (default)			INTEGER	2/2
Emb	Emb: 1	***	***	Power On	SW6.1
<b>9992</b>	<b>HMI_TESTAUTOMAT_OPTION</b>			H05	<b>CR:</b> FBT, FBSP, EMB
-	Options for HMI automatic test machine			INTEGER	2/2
Emb	Emb: 0	***	***	Power On	SW6.3
<b>9993</b>	<b>HMI_WIZARD_OPTION</b>			H05	<b>CR:</b> FBT, FBSP, EMB
-	Options for the wizard			INTEGER	2/2
Emb	Emb: 0	***	***	Power On	SW6.3
<b>9999</b>	<b>TRACE</b>			H05	<b>CR:</b> -
-	Test flags for internal diagnosis			INTEGER	2/2
OP30, Adv, Emb	OP30: 0, Adv: 0, Emb: 0	0	0xFFFF	Power On	-





# 5

## 5 Creating In-House Alarm Texts

5.1 Alarm, user, message textfiles.....	IM2/5-96
5.1.1 Storage of the text files .....	IM2/5-96
5.2 Number ranges of the text files.....	IM2/5-98
5.3 Syntax of the text file .....	IM2/5-99
5.3.1 File name .....	IM2/5-99
5.3.2 Example of an alpu.txt file.....	IM2/5-99
5.4 Supported languages.....	IM2/5-102
5.5 Bitmaps and icons .....	IM2/5-103
5.5.1 Storage of the bitmaps and icons .....	IM2/5-103
5.5.2 User status display .....	IM2/5-103
5.6 Editing in-house text files.....	IM2/5-105
5.7 Plain text for PLC machine data .....	IM2/5-107

## 5.1 Alarm, user, message textfiles

This chapter describes the basics and the procedure to follow when in-house text files, e.g., for in-house cycle alarm texts or in-house PLC alarm texts and messages, are to be inserted.

With Sinumerik 840D sl, all alarm text files are stored in the CompactFlash card's file system. The file system is subdivided into the directories (subtrees) "siemens", "addon", "oem", and "user". Each directory is structured in the same way.

---

### Note

No changes can be made in "siemens"!

---

New/modified text files can be inserted into "oem" (for the machine manufacturer) or "user" (for end user).

### Access to CompactFlash card (optional)

You can access the CompactFlash card's file system in order to insert/edit text files in the "Startup" operating area via the "HMI" → "Editor" → "Flash card" softkeys and the file functions provided there (see Section: Edit in-house text files).

---

### Note

The CompactFlash card can only be accessed while the NCU is running.

---

### 5.1.1 Storage of the text files

The alarm and message text files are stored in the corresponding subdirectories in the following areas:

"siemens" (/siemens/sinumerik/hmi/Ing/),  
"addon" (/addon/sinumerik/hmi/Ing/),  
"oem" (/oem/sinumerik/hmi/Ing/),  
"user" (/user/sinumerik/hmi/Ing/).

The text files in the "siemens" area represent the original status and cannot be modified.

The text files are sorted according to language and stored in separate subdirectories in each "Ing" directory. The subdirectory names correspond to the language abbreviations in Table 5-5.

For example, you will find German text files under the following path:

/siemens/sinumerik/hmi/Ing/deu

The text files for cycles are stored in a directory named "cycles", which is located parallel to "hmi" in each case:

```
"siemens" (/siemens/sinumerik/cycles/lng/)
"addon"   (/addon/sinumerik/cycles /lng/)
"oem"     (/oem/sinumerik/ cycles /lng/)
"user"    (/user/sinumerik/ cycles /lng/)
```

The language-dependent subdirectories for the installed languages are already set up in the "siemens" directory.

They will have to be created in the "oem" and "user" directories, if required.

### Create directory

You can create the language-dependent directories (deu, eng, etc.; see Table 5-5) in the "oem" and "user" directories.

Press softkeys "Startup" → "HMI" → "Editor" → "File Function" → "New".



#### Important

Directory names must be written in lower case letters, e.g., deu.

An external keyboard is required to write in lower case!

---

### Editable alarm text files

In-house texts can be inserted into the directories (oem/user) using the following files:

In the "hmi" subdirectory (/oem/sinumerik/hmi/lng/deu):

- alpu.txt                    PLC alarm/message texts
- alsj.txt                    Safety Integrated alarm texts
- alc.txt                     Compile cycle alarm texts

In the "cycles" subdirectory (/oem/sinumerik/cycles/lng/deu):

- aluc.txt                    User cycle and tool management texts
- Amend texts for user interface (wizard)
- alzu.txt                    User cycle alarm texts

### Foreground or background language

HMI Embedded sl is supplied with six standard languages. Two additional foreign languages can also be installed.

The installation allows you to choose any combination of two of these languages as the foreground and background language.

The language is changed using the "Change Language" vertical softkey.

## 5.2 Number ranges of the text files

### Alarm numbers

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

Table 5-1 Alarm numbers for cycle, compile-cycle and PLC alarms

Number range	Description	Effect	Delete	Name
000000 - 000320	Safety alarms			alsi.txt
085000 - 089990	User-cycle and tool-management texts, Amend texts for user interface (wizard)			aluc.txt
065000 - 065999	Cycle alarms (user)	Display, interlocking NC Start	Reset	alzu.txt
066000 - 066999		Display, interlocking NC Start, motion stop after execution of predecoded blocks	Reset	
067000 - 067999		Display	Cancel	
068000 - 069000	Reserved			
070000 - 079999	Compile cycle alarms			alc.txt
400000 - 499999	PLC alarms general			alpu.txt
500000 - 599999	PLC alarms for channel			
600000 - 699999	PLC alarms for axis and spindle			
700000 - 709999	PLC alarms for user			
800000 - 899999	PLC alarms for sequence cascades/graphs			
810000 - 810009	System messages in the PLC			

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

**References:** /DA/ Diagnostics Guide

## 5.3 Syntax of the text file

### 5.3.1 File name

Only those file names listed under “Editable alarm text files” in the previous section may be used.

### 5.3.2 Example of an alpu.txt file

```
// CP = 1252
// IDEO = NO
// ANSI = YES
// UNICODE = NO
// VERSION = $VERSION
[SWITCHES]/ALARM

//*****
/* PLC user alarm file */
//*****
/* User entries: */
/* p.e. */
/* 800001 0 0 "Safety door faulty, error analysis required" */
/* max. 62 characters! */
/* Area for Application: 700000-799999,800000-809999,811000-819999
*/
//*****
```

#### Header

The text files begin with several comments, which contain information for the runtime system.

You can enter texts that can be assigned freely after “[SWITCHES]/ALARM”.

```
// CP = XXXX   Number of the code page used to create the file (ANSI table,
               see Table 5-5)
// IDEO =     Use of Asian text:
               Yes: Logographic language
               No: Single-byte language (default: No)

// ANSI =     File format of the text file
               Yes: ANSI code page
               No: OEM code page (default: No)

// UNICODE =  File format of the text file
               Yes: Unicode
               No: Single-byte (default: No)

// VERSION =  Version identifier
```

#### Comment

You can insert comments before every line in the alarm text file. These comments have to begin with “//”.

## Format of text file for PLC alarm texts

The ASCII file for PLC alarm texts has the following structure:

Table 5-2 Structure of the text file for PLC alarm texts

Alarm number	Display	Help ID	Text or alarm number	Text displayed on HMI
510000	1	0	"Channel %K FDDIS all"	Channel 1 FDDIS all
600124	1	0	"Feed disable axis %A"	Feed disable axis 1
600224	0	0	600124	Feed disable axis 2
600324	1	0	600224	Feed disable axis 3
702463	0	0	"Group index:%A Number:%N"	Group index:24 Number:63
// Alarm text file for PLC alarm				

### Alarm number

List of alarm numbers

### Display

This number defines the alarm display type:

- 0: Display in alarm line
- 1: Display in a dialog box

### Help ID

Since help is not displayed with HMI Embedded, a "0" always has to be entered.

### Text or alarm number

Characters " and # must not be used in alarm texts. The % character is reserved for displaying parameters.

If an existing text is to be used, this can be done with a reference to the corresponding alarm. 6-digit alarm number instead of "text".

- The maximum length of the alarm text is 110 characters in total for a 2-line display. If the text is too long, it is truncated and the symbol "/" added to indicate missing text. The entries should be separated from one another by blanks.

Parameter "%K": = 0, e.g., channel number with FC10  
 (As substitute for %K, the 2nd digit of the 6-digit alarm number is inserted in the alarm text). Digit count from left to right.

Parameter "%A": = 24, the parameter is replaced by the signal group no. (e.g., axis no., user area no., sequence cascade no.)  
 (As substitute for %A, the 3rd and 4th digits of the alarm number are inserted in the alarm text).

Parameter "%N": = 63, e.g., signal number  
 (As substitute for %N, the 5th and 6th digits of the alarm number are inserted in the alarm text).

Parameter "%Z": e.g., status number for Higraph  
 (As substitute for %Z, the supplementary information for the alarm number is inserted in the alarm text. This parameter is only relevant in conjunction with HIGRAPH programming 800000...899999).

**Note**

The alarm number is output by the PLC via basic program module FC 10 (this uses PLC system function SFC 52 to send alarms to the HMI). Supplementary information is not supported.

However, the alarm number can also be transferred to the HMI by means of PLC system function SFC 17, SFC 18 (ALARM S, ALARM SQ). The mechanisms are used, for example, by the PDIAG configuring tool.

The alarm number displayed on the HMI is transferred via SFC parameter EV\_ID. The supplementary information for the alarm number is transferred via SFC parameter SD.

**Format of text file for PLC alarm texts****Range of freely assignable texts**

Table 5-3 Freely assignable range, for example, for cycle alarm texts

Alarm number	Display	Help ID	Text or alarm number
065100	1	0	"No D number %1 is programmed"
065101	1	0	600100
065202	0	0	"Axis %2 in channel %1 is still moving"

The alarm number, display and help ID are as described in the previous example.

**Text or alarm number**

The structure of the alarm text/number is the same as that described in the previous example, apart from the parameter assignment:

Parameter "%1": Channel number

Parameter "%2": Block number

**Indexed alarm texts**

Indexed alarms can be used to display additional comments in the alarm line, e.g., an explanation for "action =%"

Table 5-4 Examples:

Alarm no.	Display	Help ID	Text
010203	0	0	"Channel %1 NC Start without reference point (Action=%2<ALNX>)"
016903	1	0	"Channel %1 Action=%2<ALNX> not permitted in current status"
016912	0	0	"Channel %1 Action=%2<ALNX> only possible in reset state"

**References:** /DA/ Diagnostics Guide

## 5.4 Supported languages

To maintain compatibility with text files already existing, the following conversions from OEM to ANSI code pages are supported:

Table 5-5 Languages and ANSI table used

Language	Language codes	ANSI table code page (Windows)	Standard languages
Chinese (simplified)	chs	1252	X
Chinese (traditional)	cht	1252	
Danish	dan	1252	
German	deu	1252	X
English	eng	1252	X
Finnish	fin	1252	
French	fra	1252	X
Italian	ita	1252	X
Japanese	jpn	1252	
Korean	kor	1252	
Dutch	nld	1252	
Polish	plk	1250	
Portuguese	ptb	1252	
Russian	rus	1251	
Swedish	sve	1252	
Spanish	esp	1252	X
Czech	csy	1250	
Turkish	trk	1254	
Hungarian	hun	1250	

OEM code pages are not supported for languages not listed.



## 5.5 Bitmaps and icons

### 5.5.1 Storage of the bitmaps and icons

Bitmaps and icons are stored in the folders below, depending on their resolution and the operator panel being used:

Storage folder	Formats	Resolution	OP (operator panel)
ico640	*.ico, *.png, *.bmp	640 x 480	OP10
ico800	*.ico, *.png, *.bmp	800 x 600	OP12
ico1024	*.ico, *.png, *.bmp	1024 x 768	OP15

The images are stored on the CompactFlash card under various paths, divided into write-protected (standard) and user-specific areas.

#### Standard (write-protected)

The write-protected standard icons are stored in folders (icoxxx) using the following path:

"/siemens/sinumerik/hmi/ico/icoxxx/"

xxx stands for the relevant resolution, "640", "800", "1024" (see table above).

#### User

The user-specific icons are stored in folders (icoxxx) in the following three subdirectories:

- Standard expansion (AddOn projects)      /addon/sinumerik/hmi/ico/icoxxx
- Manufacturer                      /oem/sinumerik/hmi/ico/icoxxx
- User                                      /user/sinumerik/hmi/ico/icoxxx

### In-house configurations

Files with the configured screen definition "Expanding the user interface" (wizard "com files" including icons) are located in folders:

/oem/sinumerik/hmi/proj  
/user/sinumerik/hmi/proj

See submanual: IAM, BE1: Expanding the user interface.

### 5.5.2 User status display

Machine states controlled by the PLC can be displayed in the program status bar via user icons.

The user-icon display has to be activated via display MD 9052  
SHOW\_CHANNEL\_SPANNING\_STATE.

The program path of the currently selected program is then displayed in the bar underneath, together with the program name.

16 display positions are defined in the program status bar.

The user icons have to meet the following requirements:

- Colors: 16-color mode
- Size: OP010/OP010C/OP010S: 16 x 16 pixel (height x width)  
OP012: 20 x 20 pixels  
OP015: 27 x 26 pixels
- File name: 8 characters
- Format: BMP

## Configuring

The cross-channel status display is allocated and configured with user icons in the HEADER.INI file.

HEADER.INI is located on the CompactFlash card in directory  
siemens/sinumerik/hmi/cfg/

Copy the HEADER.INI file to directory oem\sinumerik\hmi\cfg  
Use the editor to open the file and assign the user icons to the desired positions.

Enter the names of the user icons and the signal for controlling symbol selection in the HEADER.INI file, section "UserIcons".

```
[UserIcons]
UI_0= <Ikone_00.bmp>, <position>
      UI_0:           Name of identifier
      Ikone_00.bmp:   Name of user icon
      Position:       Display position (1 to 16)
...
UI_31= <Ikone_31.bmp>, <position>

USER_ICON_BASE = DBx.DBBy
      DBx.DBBy:      Signal for controlling icon selection, defined by
                    the user.
```

User icons are addressed bit-by-bit, i.e., if bit n is set in signal DBx.DBBy, the user icon with identifier UI\_n is displayed.

If the bit is reset by the PLC, the assigned user icon in the program status display is deleted.

If several user icons are assigned to the same position, the user icon with the highest identifier number is displayed. Empty positions do not have to be entered.

## 5.6 Editing in-house text files

There are two ways to create or edit alarm text files.

1. You can copy the default files on the CompactFlash card from the "siemens" area, edit them as required and store them in your own "oem" or "user" areas.
2. You can copy in-house text files, e.g., files that have already been modified, from a USB FlashDrive and insert them in your own "oem" or "user" areas on the CompactFlash card.



### Important

The entire files must always be stored to prevent the contents of files in different subtrees being compared.

---

### Note

If you try to save a copied file to the CompactFlash card by overwriting an existing file of the same name, the message "Internal error" is output. Delete the previous file of the same name and repeat the action.

---

### Copying files, for example, from a USB FlashDrive

- If you want to copy data from a USB FlashDrive, first insert the USB FlashDrive in the USB interface on the front of the OP.
- Press the "Startup" softkey.
- Press the "HMI" softkeys.
- Select the connection, e.g., to the USB FlashDrive using the vertical softkeys ("USB front")
- Select the text file by using the arrow keys to select the directories and opening them with the "Input" key.
- Press the "File Function" softkey.
- Press the "Copy" softkey.
- Press the "Back" softkey and then select the storage path.

### Inserting files, e.g., onto the CompactFlash card

- Press the softkey, for example, "Flash Card".
- Select the appropriate directory (see section: Storage of text files) by using the arrow keys to select the directories and opening them with the "Input" key.
- Press the "File Function" softkey.
- Press the "Insert" softkey.

## Editing files

- To edit the selected file, press the "Input" key to open the file.
- Press the "Close editor" softkey to close the file.

---

### Note

User texts may only be inserted in the "oem" or "user" directories.

---

## 5.7 Plain text for PLC machine data

Application-specific and language-dependent plaintexts can be configured for each index for so-called PLC machine data.

14510[i]	User data (INT)	i = 0 ... 255
14512[i]	User data (HEX)	i = 0 ... 255
14514[i]	User data (FLOAT)	i = 0 ... 31

### Storage

Text files named **NCTEA\_xx.TXT** (xx = language code deu, eng, etc.) can be stored on the CompactFlash card in the following directories:

**user**\lng  
**addon**\lng or  
**oem**\lng

### Syntax

Number plaintext  
 Number: Number of the machine data item  
 Plaintext: Plaintext of the machine data item, maximum 62 characters  
 (The plaintext cannot be enclosed by inverted comas.)

Machine data with different indices:  
 Number[x] plaintext  
 x: Index of the machine data item

---

#### Note

If a machine data item has different indices and you do not specify an index for the plaintext, the same plaintext appears for all the indices of the machine data item.

---

#### Note

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

### Display

The text for the selected machine data item is displayed below the machine data list in the Startup operating area.

■



# 6

## 6 Series startup

6.1 Series startup.....	IM2/6-110
6.1.1 Exporting data.....	IM2/6-110
6.1.2 Importing data.....	IM2/6-111

## 6.1 Series startup

In order to transfer a particular configuration to further controls of the same software version as simply as possible (e.g., those that are operated on the same machine type), you can create what are called series startup files.

1. Series startup with or without compensation data
2. Area—specific archiving:
  - PLC data
  - Drive data

This type of file cannot be modified externally using an ASCII editor.

### 6.1.1 Exporting data

#### Procedure

- Select the "Services" operating area.
- Press the "ETC" key.
- Press the "Series start-up" horizontal softkey.  
A window appears where you can select from the following:
  - NCK data - With compensation data
  - Drive data
  - PLC data

Use the arrow keys to select an area (.arc) and mark it with the "Select" key.

- Start the "Output data" operation by pressing the "Generate Archive" softkey.
- The lower part of the window contains the "Save as..." field and the furnished drives are offered as the storage location.
- Using the arrow keys, select the storage location, e.g. "FlashCard" (CompactFlash card) or "Front" (USB FlashDrive) and press the "OK" softkey.
- The storage directory, e.g.: on the card: card/user/sinumerik/data/archive and the file, e.g., "PLC.ARC" are displayed. Press the "OK" softkey.
- Now, you can assign a different name.
- Press "OK" again to start the data transmission.

---

#### Note

When the drive data is selected, it takes 1 to 2 minutes after pressing the "Create archive" softkey until the drive has saved the data on the CompactFlash Card.

---



---

**Note**

When you save very large files to the CompactFlash card, the message "Please wait - file being flashed" appears. Acknowledge the message with the "Recall" button.

---

### 6.1.2 Importing data

To import the backed up data to another control, proceed as follows:

**Procedure**

In the "Services" area, select the created archive (arc.) and press the "Read in archive" vertical softkey.

The "Read in update archive" window pane opens.

Start importing by pressing the "OK" softkey.

Select "^" to return to the previous window.

After importing an NC series startup archive you must restart HMI-Embedded. In operating area "Installation and Startup" press vertical softkey "NCK Reset" or press keys CRT and Q and then Enter.

**Log**

Press the "Log" softkey to view the transmission report on the screen.





# 7

## 7 Data Backup

7.1 Data backup.....	IM2/7-114
7.1.1 Introduction .....	IM2/7-114
7.1.2 Data backup with HMI Embedded .....	IM2/7-116
7.1.3 Data backup for changed machine data.....	IM2/7-116

## 7.1 Data backup

A data backup should always be made if significant changes have been made.

### 7.1.1 Introduction

#### Execution

You should backup your data

- after startup,
- after changing machine-specific settings,
- after service (e.g., after replacing hardware or software), in order to resume operation quickly
- during startup before altering the memory configuration to make sure that no data are lost during startup.

The modified data can be stored over all furnished drive connections.

#### Required accessories

You will require the following accessories in order to save data:

- USB FlashDrive
- Keyboard (for uppercase/lowercase)

#### Requirements

- To store data to the CompactFlash card you need the license release for the additional 256 MB HMI user memory.
- All configured network drives must be permanently accessible while HMI Embedded is running.

## Structure of the file name

Table 7-1 File name structure for NCK data

_N_	Range	Unit	-	Type
-----	-------	------	---	------

- The area indicates which data are to be backed up or retrieved (general, channel-specific, or axis-specific).
- The unit defines the channel, the axis or the TOA area. No unit has to be specified if the entire area is selected.
- The type determines the data type. During a data backup, the file names are created and output automatically.

### Areas

NC	General NC-specific data
CH	Channel-specific data (unit corresponds to the channel number)
AX	Axis-specific data (unit corresponds to the number of the machine axis)
TO	Tool data
COMPLETE	All data of an area
INITIAL	Data for all areas (_N_INITIAL_INI)

### Types

TEA	Machine data
SEA	Setting data
OPT	Option data
TOA	Tool data
UFR	User input frames: Settable Work offset, rotations, etc.
EEC	Measuring system error compensation
CEC	Sag/angularity compensation
QEC	Quadrant error compensation
PRO	Protection zone
RPA	R parameters
GUD	Global user data
INI	General initialization program (all data in the active file systems)

_N_COMPLETE_TEA	Archiving of all machine data
_N_AX_TEA	Archiving of all axis machine data
_N_CH1_TEA	Archiving of the machine data for channel 1
_N_CH1_GUD	Archiving of the global user data for channel 1
_N_INITIAL_INI	Archiving of all data in the active file system

**References:** /IDsl/ Installation and Startup Manual CNC Part 1 (NCK, PLC, Drive)

### 7.1.2 Data backup with HMI Embedded

For SINUMERIK 840D sl, the data backup of various components is divided as follows:

1. Data backup for NCK
2. Data backup for PLC
3. Drive data

The procedure for this data backup is described in the previous chapter: "Series startup".

Next to backup of "Series startup" data you can also back up individual files in operating area "Services" using the softkeys "Copy" and "Insert". Besides saving them to the CompactFlash card you can also save them to other memory media, for example a USB FlashDrive, via the network connections set up on your machine.

See the procedure in Section: Alarm, user, and message text files.

### 7.1.3 Data backup for changed machine data

#### Saving changed values MD 11210

Using the general machine data

MD: 11210 UPLOAD\_MD\_CHANGES\_ONLY (MD backup only of changed machine data) it can be set for the backup of the machine and setting data

whether all data or only data differing from the default setting should be output.

If the value of a data item which is stored as an array, is changed, the entire MD array is always output (e.g., AXCONF\_MA-CHAX\_NAME\_TAB).

<b>11210</b>	<b>UPLOAD_MD_CHANGES_ONLY</b>	
MD number	MD backup of changed MD only	
Default setting: 0	Minimum input limit: 0	Maximum input limit: 255
Change effective: Immediately	Protection level: 2/4	Unit: –
Data type: BYTE		
Significance:	Selection of differential MD upload:	
	Bit 0 (LSB) Scope of the differential upload with TEA files (area-specific archiving) 0: All data are output 1: Only data, which deviate from the standard, are output (not valid for INITIAL_INI) If a value of a data item, which is stored as an array, is changed, the entire MD array is always output. (e.g., MD 10000: AXCONF_MACHAX_NAME_TAB).	
	Bit 1 Scope of the differential upload with INI files 0: All data are output 1: Only data, which deviate from the standard, are output (e.g., INITIAL_INI)	

	<p>Bit 2 Changes of a field element 0: Complete array is output 1: Only modified elements of an array are output</p> <p>Bit 3 R parameters (only for INITIAL_INI) 0: All R parameters are output 1: Only R parameters not equal to zero are output</p> <p>Bit 4 Frames (only for INITIAL_INI) 0: All frames are output 1: Only frames not equal to zero are output</p> <p>Bit 5 Tool data, cutting edge parameters (only for INITIAL_INI) 0: All tool data are output 1: Only tool data not equal to zero are output</p>
	<p>Bit 6 Battery-backed system variables (\$AC_MARKER [ ]; \$AC_PARAM [ ] only for INITIAL_INI) 0: All system variables are output 1: Only system variables not equal to 0 are output</p> <p>Bit 7 Synchronized action GUD (only for INITIAL_INI) 0: All synchronized action GUD are output 1: Only synchronized action GUD not equal to zero are output</p>
	Effective: Changes to the data become effective when the upload for the next area is started.

**Note**

It might be sensible to back up only altered machine data before a software update if changes have been made to the default machine data settings in the new software version. This applies particularly to machine data, which are assigned SIEMENS protection level 0.

**Note**

MD 11210 UPLOAD\_MD\_CHANGES\_ONLY should be set to "1" or the corresponding bits set to "1". With this setting, the transferred files contain only those data, which deviate from the default.







# 8

## 8 Diagnostics

8.1 Action log .....	IM2/8-120
8.1.1 Log file: "action.log" .....	IM2/8-122
8.1.2 Displaying the log file / setting the action log.....	IM2/8-123
8.1.3 Outputting the log file .....	IM2/8-123
8.2 Software version display .....	IM2/8-124

## 8.1 Action log

The action log function is used to log operating sequences so you can analyze them at a later date.

### Function

The following functions are available:

- Logging of alarms
- Logging of key actuation
- Output of log file via a drive that has been set up, for example, Front (USB FlashDrive), FlashCard (CompactFlash card).

The action log is password-encrypted (protection level less than / equal to 3).

Can be activated/deactivated using the display machine data

MD: 9012 ACTION\_LOG\_MODE.

### Note

The precondition for the Action Log function is that the "Startup" operating area is installed.

9012	ACTION_LOG_MODE		
MD number	Action log settings		
Default value: 255	Min. input value: 0	Max. input value: 0xffff	
Changes effective after POWER ON	Protection level (R/W) 2/2	Unit: -	
Data type: DWORD			
Meaning	This MD can be used to enable/disable the action log and to define a choice of the data to be logged.		
	Bit 0	= 1	Action log ON (default)
		= 0	Action log OFF
	Bit 1	= 1	HMI Advanced: Variable services (write accessing of geometrical data, e.g., tool offset) are logged. Meaning of the variables, see help in operating area Parameters under Variable Views (default).
		= 0	Variable services are not logged.
	Bit 2	= 1	HMI Advanced: PI services (e.g., program selection) are logged, for meaning of variables see help in operating area Parameters under Variable Views (default).
		= 0	PI services are not logged
	Bit 3	= 1	HMI Advanced: Domain services (e.g., load/unload program) are logged (default).
		= 0	Domain services are not logged.
	Bit 4	= 1	Changes to the alarm status are logged (default).
	= 0	Changes to the alarm status are not logged.	

	Bit 5	= 1	Key actuations are logged (default).
		= 0	Key actuations are not logged.
	Bit 6	= 1	Channel status/override is logged (default).
		= 0	Channel status/override is not logged
	Bit 7	= 1	HMI Advanced: Softkey actuations and menu changes are logged. For Siemens internal use only (default) <b>HMI Embedded:</b> Identification number (ID) of the open and closed window is logged. For Siemens internal use only (default).
		= 0	Softkey actuations and menu changes are not logged.

If the action log is enabled, the following loggable files are logged by default.

### Alarm status

All displayed NC/PLC and HMI Embedded alarms and messages are logged.

### Key actuation

All actions on the operator panel front and on the MF2 keyboard are logged.

### Channel status/override

NC/PLC statuses are logged using information about the channel status and the position of the override switch. At least in parts, if they can be recorded by time, these statuses are used to verify the operation of the MCP.

### Window ID

The identification number of a screen is logged when it is opened or closed.

### 8.1.1 Log file: "action.log"

The log file is output in English.

#### File structure of "action.log"

The log file is divided into 4 columns:

- Date
- Time
- NCU name
- Event

#### Example of an "action.log":

Date	Time	Event specification
26.02.1999	07:45:20	HMI booting (conn.no: 0, MM_ACTION_LOG_MODE: ff, entries: 4094)
26.02.1999	07:45:20	Alarm 300701
29.09.1999	18:59:10	Key: 68 KEY_SELECT_MODE (0)
29.09.1999	18:59:07	Open window: 20005
29.09.1999	18:59:07	Close window: 20020

#### Output of the window name

The window name, if known, is displayed in addition to the window ID in the action log.

e.g.:

....

27.02.2002	18:59:07	Open window: (10001) Alarm
------------	----------	----------------------------

For older software versions, you can find an assignment of the window IDs to the window names in the "Documentation \ Window Lists" directory on the software CD.

For each software version, there is an assignment file, which contains the version ID in the file name (e.g., 060507.txt for the software version P06.05.07).

The installed software version appears at the start of the log file.

## 8.1.2 Displaying the log file / setting the action log

### Displaying the log file

The display is password-protected.

The current log file is displayed on the operator panel screen in the "Diagnostics" operating area, "Service Displays" → Action Log → menu in the "Action Log File" window.

If a crash file exists, the softkey "Crash Data" is also displayed, with which the crash file can be displayed.

### Setting the action log

The action log settings are password-encrypted.

The "Action Log Settings" screen is displayed in the "HMI Embedded" → "Action Log" menu in the "Startup" operating area.

Here you can:

- Enable/disable the logging,
- Select the data to be logged,
  - Alarm status change
  - Actuate keys
  - Channel status/override
  - Window IDs

The selected setting must be confirmed by pressing the softkey "OK".

## 8.1.3 Outputting the log file

The log file output is password-encrypted.

There are two possible ways to output the log file:

1. In the "Diagnostics" operating area, select "Service Displays" → "Action Log" → "Save As..." → "FlashCard"/"Front" (logical drive that has been set up).
2. Via the boot menu.  
A ramp-up menu is displayed whenever a crash event occurs (PLC bit DB19, byte0, bit6). The bit is set and reset from the PLC at the user interface (as with key lock, screen darkening, etc.).

HMI Embedded evaluates the rising edge of the signal (i.e., a transition from 0 to 1) and creates the log file when the signal comes. HMI Embedded only reads the data, i.e., the signal must be reset by the PLC/user program.

If the bit is already set when HMI Embedded is booted, HMI Embedded does not respond to the value of the bit (only after it has been reset and set again).

- The log file `_AC_LOGP.TXT` is read out either during the boot process or can be read out in the Diagnosis operating area until the system is next switched off. The file is then deleted.
- Reboot of the control with
- Prompt: "Action log detected! Read out CF-Card? ([Y]/N)"
- Acknowledge with "Yes"; the log file is now saved to the CompactFlash card.

---

**Note**

A latency must be observed between two signals to make sure that the second signal can be processed.

---

## 8.2 Software version display

### Function

The version data of the installed system software are output in the version display.

### Procedure

- Select the "Diagnostics" operating area
- Select the "Service displays" softkey
- Select the "Version" softkey

You can display the version data of the following areas via the horizontal softkeys:

- Version data of the NCU
- Version data of the HMI
- Version data of the cycles (user, manufacturer, standard cycles)
- Definitions
- Compile cycles



# I Index

## I.1 Subject Index

### 2

256 MB user memory ..... IM2/1-9

### A

Access rights ..... IM2/4-33

Accessories ..... IM2/1-9

Action log

  Outputting the log file ..... IM2/8-123

ACTION\_LOG\_MODE

  MD 9012..... IM2/4-76

ACTION\_LOG\_MODE

  MD9012..... IM2/8-120

Actionlog

  Set..... IM2/8-123

ACTIVATE\_SEL\_USER\_DATA

  MD 9440..... IM2/4-88

Activating tool correction

  immediately ..... IM2/4-48

Activating/deactivating calculator .. IM2/4-38

Alarm display ..... IM2/4-41

Alarm number ranges..... IM2/5-98

Alarm text

  Storage path..... IM2/5-96

Alarm text files

  Syntax ..... IM2/5-99

Alarm texts

  Create directory ..... IM2/5-97

  Editable text files ..... IM2/5-97

Alarm texts..... IM2/5-96

ALARM\_ROTATION\_CYCLE

  MD 9056..... IM2/4-78

alc.txt ..... IM2/5-97

alpu.txt ..... IM2/5-97

alsi.txt ..... IM2/5-97

aluc.txt ..... IM2/5-97

alzu.txt ..... IM2/5-97

ANSI table ..... IM2/5-102

Articulated position STAT ..... IM2/4-54

### B

Base offset ..... IM2/4-50

Bitmaps

  Storage path ..... IM2/5-103

Blue screen ..... IM2/3-28

### C

Changed machine data,

  data backup ..... IM2/7-116

Changing protection levels,

  REDEF ..... IM2/4-35

Changing the language..... IM2/4-38

CMM\_DIRECTORY\_SOFTKEY\_PATH1

  MD 9676 ..... IM2/4-92

CMM\_DIRECTORY\_SOFTKEY\_PATH2

  MD 9677 ..... IM2/4-93

CMM\_DIRECTORY\_SOFTKEY\_PATH3

  MD 9678 ..... IM2/4-93

CMM\_DIRECTORY\_SOFTKEY\_PATH4

  MD 9679 ..... IM2/4-93

CMM\_ENABLE\_CUSTOMER\_M\_CODES

  MD 9661 ..... IM2/4-92

CMM\_ENABLE\_TOOL\_MAGAZINE

  MD 9674 ..... IM2/4-92

CMM\_FIXED\_TOOL\_PLACE

  MD 9672 ..... IM2/4-92

CMM\_POS\_COORDINATE\_SYSTEM

  MD 9650 ..... IM2/4-92

CMM\_TOOL\_DISPLAY\_IN\_DIAM

  MD 9663 ..... IM2/4-92

CMM\_TOOL\_LIFE\_CONTROL

  MD 9652 ..... IM2/4-92

CMM\_TOOL\_LOAD\_STATION

  MD 9673 ..... IM2/4-92

CMM\_TOOL\_MANAGEMENT

  MD 9651 ..... IM2/4-92

- Code pages ..... IM2/5-102
- Communication ..... IM2/3-16
- CompactFlash card
- Access ..... IM2/5-96
  - Slot ..... IM2/2-14
- Configuring CAPSLOCK ..... IM2/4-37
- Configuring keyboard response .... IM2/4-37
- Connection structure between RCS Host Embedded und RCS Viewer Embedded ..... IM2/4-67
- CONTOUR\_END\_TEXT
- MD 9461 ..... IM2/4-88
- Coordinate system
- for milling ..... IM2/4-58
- Coordinate systems ..... IM2/4-58
- Copy
- Text files ..... IM2/5-105
- CTM\_ANGLE\_REFERENCE\_AXIS
- MD 9632 ..... IM2/4-91
- CTM\_CROSS\_AX\_DIAMETER\_ON
- MD 9611 ..... IM2/4-91
- CTM\_G91\_DIAMETER\_ON
- MD 9619 ..... IM2/4-91
- CTM\_POS\_COORDINATE\_SYSTEM
- MD 9610 ..... IM2/4-91
- CTM\_SIMULATION\_DEF\_VIS\_AREA
- MD 9602 ..... IM2/4-90
- CTM\_SIMULATION\_DEF\_X
- MD 9600 ..... IM2/4-90
- CTM\_SIMULATION\_DEF\_Y
- MD 9601 ..... IM2/4-90
- CTM\_SIMULATION\_MAX\_VIS\_AREA
- MD 9605 ..... IM2/4-91
- CTM\_SIMULATION\_MAX\_X
- MD 9603 ..... IM2/4-90
- CTM\_SIMULATION\_MAX\_Y
- MD 9604 ..... IM2/4-91
- CTM\_SIMULATION\_TIME\_NEW\_POS
- MD 9606 ..... IM2/4-91
- CTM\_TRACE
- MD 9626 ..... IM2/4-91
- Cycle alarm texts
- Format ..... IM2/5-101
  - Structure ..... IM2/5-101
- D**
- D number ..... IM2/4-48
- DARKTIME\_TO\_PLC
- MD 9015 ..... IM2/4-76
- Data backup
- General ..... IM2/7-114
- Display machine data ..... IM2/4-75
- DISPLAY\_BACKLIGHT
- MD 9025 ..... IM2/4-77
- DISPLAY\_BLACK\_TIME
- MD 9006 ..... IM2/4-76
- DISPLAY\_MODE
- MD 9002 ..... IM2/4-75
- DISPLAY\_RESOLUTION
- MD 9004 ..... IM2/4-75
- DISPLAY\_RESOLUTION\_INCH
- MD 9011 ..... IM2/4-76
- DISPLAY\_TYPE
- MD 9001 ..... IM2/4-75
- Displaying axes ..... IM2/4-54
- Dynamic user memory (DRAM) ..... IM2/4-64
- E**
- Edit
- Text file ..... IM2/5-106
- Editor
- Setting from the HMI user interface ..... IM2/4-39
  - Setting ..... IM2/4-39
- Enter server name ..... IM2/3-23
- Error status screen ..... IM2/3-28
- EXPONENT\_LIMIT
- MD 9030 ..... IM2/4-77
- EXPONENT\_SCIENCE
- MD 9031 ..... IM2/4-77
- F**
- File manager ..... IM2/4-40
- File transfer
- RCS Viewer Embedded ..... IM2/4-74
- Fine correction for tools subject to wear ..... IM2/4-48
- Fine work offset ..... IM2/4-50
- FIRST\_LANGUAGE
- MD 9003 ..... IM2/4-75
- G**
- G code editor ..... IM2/4-56
- Geometry processor ..... IM2/4-40
- H**
- Hardware
- 840D sl ..... IM2/1-8
- HEADER.INI ..... IM2/5-104



Hiding the display of the machine  
 axes ..... IM2/4-53  
 HMI\_HELP\_SYSTEMS  
 MD 9991 ..... IM2/4-94  
 HMI\_MONITOR  
 MD 9032 ..... IM2/4-77  
 HMI\_TESTAUTOMAT\_OPTION  
 MD 9992 ..... IM2/4-94  
 HMI\_WIZARD\_OPTION  
 MD 9993 ..... IM2/4-94

**I**

Icon  
 Storage path ..... IM2/5-103  
 Inch/metric switchover ..... IM2/4-41  
 In-house text files  
 Copy ..... IM2/5-105  
 Edit ..... IM2/5-106  
 Inserting  
 Text file ..... IM2/5-105

**K**

KEYBOARD\_STATE  
 MD 9009 ..... IM2/4-76  
 KEYBOARD\_STATE ..... IM2/4-37  
 KEYBOARD\_TYPE  
 MD 9008 ..... IM2/4-76  
 Keyboard ..... IM2/1-9  
 Keyswitch position ..... IM2/4-35  
 Keyswitch ..... IM2/4-33

**L**

LANGUAGE\_SETTINGS  
 MD 9980 ..... IM2/4-94  
 LAYOUT\_MODE  
 MD 9021 ..... IM2/4-77  
 LCD\_CONTRAST  
 MD 9000 ..... IM2/4-75  
 Licensing  
 All options ..... IM2/3-17  
 Missing options ..... IM2/3-17  
 Overview ..... IM2/3-16  
 Licensing ..... IM2/1-9, IM2/3-16  
 Log  
 Series startup ..... IM2/6-111  
 Log file  
 Action log structure ..... IM2/8-122  
 Display action log ..... IM2/8-123  
 Output, action log ..... IM2/8-123

LOGDRIVE.INI  
 Example ..... IM2/3-26  
 LOGDRIVE.INI ..... IM2/3-25  
 Logical drive ..... IM2/3-18

**M**

MA\_AUXFU\_GROUPS  
 MD 9442 ..... IM2/4-88  
 MA\_AX\_DRIVELOAD\_FROM\_PL1  
 MD 9426 ..... IM2/4-87  
 MA\_AX\_DRIVELOAD\_FROM\_PL2  
 MD 9427 ..... IM2/4-87  
 MA\_AXES\_SHOW\_GEO\_FIRST  
 MD 9421 ..... IM2/4-86  
 MA\_COORDINATE\_SYSTEM  
 MD 9424 ..... IM2/4-87  
 MA\_MAX\_SKP\_LEVEL  
 MD 9423 ..... IM2/4-87  
 MA\_ONLY\_MKS\_DIST\_TO\_GO  
 MD 9420 ..... IM2/4-86  
 MA\_PRESET\_MODE  
 MD 9422 ..... IM2/4-86  
 MA\_SCRATCH\_DEFAULT\_MODE  
 MD 9425 ..... IM2/4-87  
 MA\_SIMULATION\_MODE  
 MD 9480 ..... IM2/4-89  
 MA\_SPIND\_MAX\_POWER  
 MD 9428 ..... IM2/4-87  
 MA\_SPIND\_POWER\_RANGE  
 MD 9429 ..... IM2/4-87  
 MA\_STAND\_SIMULATION\_LIMIT  
 MD 9481 ..... IM2/4-89  
 MA\_STAT\_DISPLAY\_BASE  
 MD 9242 ..... IM2/4-81  
 MA\_TU\_DISPLAY\_BASE  
 MD 9243 ..... IM2/4-81  
 MAX\_PROGRAMM\_SIZE\_CHECK  
 MD 9464 ..... IM2/4-88  
 MD 9052 ..... IM2/5-103  
 MD 9460 PROGRAM\_SETTINGS  
 ..... IM2/4-39, IM2/4-40  
 MD\_AX\_TEA\_FILTER  
 MD 9952 ..... IM2/4-93  
 MD\_AX\_TEA\_IDX\_LIMIT  
 MD 9953 ..... IM2/4-93  
 MD\_CH\_TEA\_FILTER  
 MD 9954 ..... IM2/4-93  
 MD\_CH\_TEA\_IDX\_LIMIT  
 MD 9955 ..... IM2/4-93  
 MD\_DRV\_TEA\_FILTER  
 MD 9956 ..... IM2/4-93  
 MD\_DRV\_TEA\_IDX\_LIMIT  
 MD 9957 ..... IM2/4-93

MD_NC_TEA_FILTER		PLC_ALARM_PICTURE	
MD 9950.....	IM2/4-93	MD 9055 .....	IM2/4-78
MD_NC_TEA_IDX_LIMIT		PLC_HOTKEY	
MD 9951.....	IM2/4-93	MD 9017 .....	IM2/4-76
MD_TEXT_SWITCH		PLC_SYMBOL_FILTER	
MD 9900.....	IM2/4-93	MD 9054 .....	IM2/4-78
Measuring cycles.....	IM2/4-56	PLC_SYMBOL_SORT	
Memory allocation		MD 9053 .....	IM2/4-77
Tool management .....	IM2/4-62	Prerequisites	
User data.....	IM2/4-65	hardware.....	IM2/1-8
Memory expansion of the CF card .	IM2/1-9	software .....	IM2/1-8
Memory media.....	IM2/1-9	Preset.....	IM2/4-51
Message texts .....	IM2/5-96	Presetting actual value memory ....	IM2/4-51
Metric/inch switchover .....	IM2/4-41	PRG_DEFAULT_DIR	
Milling simulation 3D graphic.....	IM2/4-56	MD 9005 .....	IM2/4-75
Milling simulation .....	IM2/4-56	Program simulation.....	IM2/4-55
		PROGRAM_SETTINGS	
		MD 9460 .....	IM2/4-88
<b>N</b>			
NC_PROPERTIES		<b>R</b>	
MD 9500.....	IM2/4-89	RCS Host Embedded	
NETNAMES.INI		Adapting to PC.....	IM2/4-70
file extension .....	IM2/4-43	Configure .....	IM2/4-69
Network		Connection structure .....	IM2/4-67
Configuring.....	IM2/3-21	File Transfer.....	IM2/4-71
Network operation		Sequence.....	IM2/4-68
With HMI-Embedded .....	IM2/3-18	Starting .....	IM2/4-73
Network		RCS Viewer Embedded	
Configure softkey .....	IM2/3-18	Connection structure .....	IM2/4-67
Configuring, example .....	IM2/3-22	File Transfer.....	IM2/4-71
Connection via MD .....	IM2/3-24	install.....	IM2/4-71
Connections via INI file .....	IM2/3-25	Sequence.....	IM2/4-68
Connections .....	IM2/3-18	REDEF .....	IM2/4-35
Save data .....	IM2/3-28	Resources	
USB interface .....	IM2/3-18	Displaying and editing .....	IM2/4-60
NUM_AX_SEL		RI suppression measures .....	IM2/1-10
MD 9027.....	IM2/4-77	Rotary axis position.....	IM2/4-54
<b>P</b>			
PA_ZOA_MODE		<b>S</b>	
MD 9459.....	IM2/4-88	Scratching .....	IM2/4-51
Password		Screen	
Default password .....	IM2/4-34	Screen darkening via PLC .....	IM2/4-37
Permitted characters .....	IM2/4-34	Select tools via a D number	
Resetting .....	IM2/4-34	(flat D no.) .....	IM2/4-48
Setting .....	IM2/4-34	Series startup.....	IM2/6-110
PLC alarm acknowledgement icon	IM2/4-41	Set date.....	IM2/4-38
PLC alarm texts		Set program overview.....	IM2/4-40
Format.....	IM2/5-100	Set time .....	IM2/4-38
Structure.....	IM2/5-100	SHOW_CHANNEL_SPANNING_STATE	
PLC machine data		MD 9052 .....	IM2/4-77
Plaintext.....	IM2/5-107	Simulation	
		milling .....	IM2/4-56

milling .....	IM2/4-56	MD 9435 .....	IM2/4-88
turning .....	IM2/4-56	TM_LOAD_LOC3	
Software		MD 9436 .....	IM2/4-88
HMI Embedded sl.....	IM2/1-9	TM_LOAD_LOC4	
Software version display .....	IM2/8-124	MD 9437 .....	IM2/4-88
Spindles, analog .....	IM2/4-49	TM_LOAD_LOC5	
Standard languages .....	IM2/5-102	MD 9438 .....	IM2/4-88
Start-up of the NCU .....	IM2/2-14	TM_LOAD_PLACE	
STARTUP_LOGO		MD 9410 .....	IM2/4-86
MD 9050.....	IM2/4-77	TM_LOAD_TOOL_NEW	
STAT .....	IM2/4-54	MD 9431 .....	IM2/4-87
Static user memory (SRAM) .....	IM2/4-60	TM_NUM_MAG	
Status (STAT)		MD 9411 .....	IM2/4-86
hide, HMI-Embedded .....	IM2/4-55	TM_TOOL_STATE_DEF_VAL	
show, HMI-Embedded.....	IM2/4-55	MD 9432 .....	IM2/4-87
SW_OPTIONS		TM_UNLOAD_AND_DELETE	
MD 9990.....	IM2/4-94	MD 9430 .....	IM2/4-87
SWITCH_TO_AREA		TO_MAG_PLACE_DISTANCE	
MD 9016.....	IM2/4-76	MD 9479 .....	IM2/4-89
Switching channel		TO_OPTION_MASK	
Foreign language texts.....	IM2/4-45	MD 9478 .....	IM2/4-89
Switching channels		TO_TRACE	
Example .....	IM2/4-43	MD 9477 .....	IM2/4-89
Switching channels.....	IM2/4-42	Tool management	
Switching operator stations .....	IM2/4-42	Memory allocation .....	IM2/4-62
SYS_CLOCK_SYNC_TIME		ShopMill .....	IM2/4-47
MD 9013.....	IM2/4-76	Standard .....	IM2/4-47
<b>T</b>		TOOL_REF_GEO_AXIS1	
TABULATOR_SIZE		MD 9400 .....	IM2/4-85
MD 9007.....	IM2/4-76	TOOL_REF_GEO_AXIS2	
TEACH_MODE		MD 9401 .....	IM2/4-85
MD 9026.....	IM2/4-77	TOOL_REF_GEO_AXIS3	
TECHNOLOGY		MD 9402 .....	IM2/4-85
MD 9020.....	IM2/4-77	TRACE	
Text file name .....	IM2/5-99	MD 9999 .....	IM2/4-94
TM_ACT_SEARCH_AND_POS		TU .....	IM2/4-54
MD 9433.....	IM2/4-87	Turn (TU)	
TM_DEFAULT_DELETE_TOOL		hide, HMI-Embedded .....	IM2/4-55
MD 9419.....	IM2/4-86	show, HMI-Embedded .....	IM2/4-55
TM_DEFAULT_TOOLPLACESPEC		Turning simulation.....	IM2/4-56
MD 9415.....	IM2/4-86	<b>U</b>	
TM_DEFAULT_TOOLSIZE		USB FlashDrive .....	IM2/1-9
MD 9412.....	IM2/4-86	USE_CHANNEL_DISPLAY_DATA	
TM_DEFAULT_TOOLSTATE		MD 9014 .....	IM2/4-76
MD 9417.....	IM2/4-86	User displays	
TM_DEFAULT_TOOLTYPE		expand .....	IM2/4-57
MD 9416.....	IM2/4-86	User status display.....	IM2/5-103
TM_KIND_OF_TOOLMANAGEMENT		User text files	
MD 9414.....	IM2/4-86	Inserting .....	IM2/5-105
TM_LOAD_LOC1		User texts.....	IM2/5-96
MD 9434.....	IM2/4-87	USER_CLASS_APPLICATION	
TM_LOAD_LOC2		MD 9272 .....	IM2/4-83

USER_CLASS_BASE_ZERO_OFF_MA MD 9248.....	IM2/4-81	MD 9227 .....	IM2/4-80
USER_CLASS_BASE_ZERO_OFF_PA MD 9247.....	IM2/4-81	USER_CLASS_SYS_ZERO_OFF MD 9246 .....	IM2/4-81
USER_CLASS_CLEAR_RPA MD 9221.....	IM2/4-80	USER_CLASS_TEACH_IN MD 9219 .....	IM2/4-80
USER_CLASS_DIRECTORY_CHG MD 9509.....	IM2/4-89	USER_CLASS_TM_SKACTPLACE MD 9270 .....	IM2/4-82
USER_CLASS_DIRECTORY1_M MD 9516.....	IM2/4-90	USER_CLASS_TM_SKFINDPLACE MD 9269 .....	IM2/4-82
USER_CLASS_DIRECTORY1_P MD 9510.....	IM2/4-89	USER_CLASS_TM_SKLDTOOLDAT MD 9271 .....	IM2/4-83
USER_CLASS_DIRECTORY2_M MD 9517.....	IM2/4-90	USER_CLASS_TM_SKMGBUFFER MD 9260 .....	IM2/4-82
USER_CLASS_DIRECTORY2_P MD 9511.....	IM2/4-89	USER_CLASS_TM_SKMGFIND MD 9261 .....	IM2/4-82
USER_CLASS_DIRECTORY3_M MD 9518.....	IM2/4-90	USER_CLASS_TM_SKMGLISTPOS MD 9262 .....	IM2/4-82
USER_CLASS_DIRECTORY3_P MD 9512.....	IM2/4-89	USER_CLASS_TM_SKMGLREPR2 MD 9256 .....	IM2/4-81
USER_CLASS_DIRECTORY4_M MD 9519.....	IM2/4-90	USER_CLASS_TM_SKMGLREPR3 MD 9257 .....	IM2/4-81
USER_CLASS_DIRECTORY4_P MD 9513.....	IM2/4-89	USER_CLASS_TM_SKMGNEXT MD 9263 .....	IM2/4-82
USER_CLASS_INCH_METRIC MD 9182.....	IM2/4-78	USER_CLASS_TM_SKNCDELTOOL MD 9259 .....	IM2/4-82
USER_CLASS_OVERSTORE_HIGH MD 9213.....	IM2/4-79	USER_CLASS_TM_SKNCNEWTTOOLE MD 9258 .....	IM2/4-82
USER_CLASS_PRESET MD 9220.....	IM2/4-80	USER_CLASS_TM_SKTLLIST MD 9251 .....	IM2/4-81
USER_CLASS_READ_BD MD 9230.....	IM2/4-80	USER_CLASS_TM_SKTLLREPR1 MD 9265 .....	IM2/4-82
USER_CLASS_READ_CST MD 9225.....	IM2/4-80	USER_CLASS_TM_SKTLLREPR2 MD 9266 .....	IM2/4-82
USER_CLASS_READ_CUS MD 9226.....	IM2/4-80	USER_CLASS_TM_SKTLLREPR3 MD 9267 .....	IM2/4-82
USER_CLASS_READ_DEF MD 9229.....	IM2/4-80	USER_CLASS_TM_SKTLNEWTOOL MD 9264 .....	IM2/4-82
USER_CLASS_READ_GUD_LUD MD 9211.....	IM2/4-79	USER_CLASS_TM_SKTOOLLOAD MD 9252 .....	IM2/4-81
USER_CLASS_READ_IN MD 9224.....	IM2/4-80	USER_CLASS_TM_SKTOOLMOVE MD 9254 .....	IM2/4-81
USER_CLASS_READ_PROGRAM MD 9216.....	IM2/4-79	USER_CLASS_TM_SKTOOLUNLOAD MD 9253 .....	IM2/4-81
USER_CLASS_READ_SYF MD 9228.....	IM2/4-80	USER_CLASS_VERT_MODE_SK MD 9249 .....	IM2/4-81
USER_CLASS_READ_TCARR MD 9180.....	IM2/4-78	USER_CLASS_WRITE_FINE MD 9203 .....	IM2/4-79
USER_CLASS_READ_TOA MD 9200.....	IM2/4-79	USER_CLASS_WRITE_GUD_LUD MD 9212 .....	IM2/4-79
USER_CLASS_SELECT_PROGRAM MD 9218.....	IM2/4-80	USER_CLASS_WRITE_PRG_CONDIT MD 9214 .....	IM2/4-79
USER_CLASS_SET_V24 MD 9223.....	IM2/4-80	USER_CLASS_WRITE_PROGRAM MD 9217 .....	IM2/4-79
USER_CLASS_SHOW_SBL2		USER_CLASS_WRITE_RPA MD 9222 .....	IM2/4-80

USER_CLASS_WRITE_SEA	
MD 9215.....	IM2/4-79
USER_CLASS_WRITE_TCARR	
MD 9181.....	IM2/4-78
USER_CLASS_WRITE_TOA_ADAPT	
MD 9209.....	IM2/4-79
USER_CLASS_WRITE_TOA_GEO	
MD 9201.....	IM2/4-79
USER_CLASS_WRITE_TOA_WEAR	
MD 9202.....	IM2/4-79
USER_CLASS_WRITE_ZOA	
MD 9210.....	IM2/4-79
USER_WRITE_CYCFRAME	
MD 9186.....	IM2/4-78
USER_WRITE_EXTFRAME	
MD 9188.....	IM2/4-78
USER_WRITE_PARTFRAME	
MD 9184.....	IM2/4-78
USER_WRITE_TOOLFRAME	
MD 9183.....	IM2/4-78
USER_WRITE_TRAFRAME	
MD 9187.....	IM2/4-78
USER_WRITE_WPFRAME	
MD 9185.....	IM2/4-78
Utilization display.....	IM2/4-60

**V**

V24_PG_PC_BAUD	
MD 9325.....	IM2/4-85
V24_PG_PC_CONTROLS	
MD 9323.....	IM2/4-85
V24_PG_PC_DATABITS	
MD 9326.....	IM2/4-85
V24_PG_PC_EOF	
MD 9322.....	IM2/4-85
V24_PG_PC_LINE	
MD 9329.....	IM2/4-85
V24_PG_PC_PARITY	
MD 9327.....	IM2/4-85
V24_PG_PC_RTS	
MD 9324.....	IM2/4-85
V24_PG_PC_STOPBIT	
MD 9328.....	IM2/4-85
V24_PG_PC_XOFF	
MD 9321.....	IM2/4-84
V24_PG_PC_XON	
MD 9320.....	IM2/4-84
V24_PRINTER_BAUD	
MD 9315.....	IM2/4-84

V24_PRINTER_CONTROLS	
MD 9313.....	IM2/4-84
V24_PRINTER_DATABITS	
MD 9316.....	IM2/4-84
V24_PRINTER_EOF	
MD 9312.....	IM2/4-84
V24_PRINTER_LINE	
MD 9319.....	IM2/4-84
V24_PRINTER_PARITY	
MD 9317.....	IM2/4-84
V24_PRINTER_RTS	
MD 9314.....	IM2/4-84
V24_PRINTER_STOPBIT	
MD 9318.....	IM2/4-84
V24_PRINTER_XOFF	
MD 9311.....	IM2/4-84
V24_PRINTER_XON	
MD 9310.....	IM2/4-84
V24_USER_BAUD	
MD 9305.....	IM2/4-83
V24_USER_CONTROLS	
MD 9303.....	IM2/4-83
V24_USER_DATABITS	
MD 9306.....	IM2/4-83
V24_USER_EOF	
MD 9302.....	IM2/4-83
V24_USER_LINE	
MD 9309.....	IM2/4-83
V24_USER_PARITY	
MD 9307.....	IM2/4-83
V24_USER_RTS	
MD 9304.....	IM2/4-83
V24_USER_STOPBIT	
MD 9308.....	IM2/4-83
V24_USER_XOFF	
MD 9301.....	IM2/4-83
V24_USER_XON	
MD 9300.....	IM2/4-83

**W**

Work offset.....	IM2/4-50
Workpiece template.....	IM2/4-49
WRITE_TOA_FINE_LIMIT	
MD 9450.....	IM2/4-88
WRITE_ZOA_FINE_LIMIT	
MD 9451.....	IM2/4-88





# SINUMERIK 840D sl/840Di sl

## SINUMERIK 840D/810D

### CNC Part 2 (HMI)

## Commissioning HMI Advanced (IM4)

### Commissioning Manual

#### Valid for

##### *Control*

SINUMERIK 840D sl / 840DE sl  
SINUMERIK 840Di sl / 840DiE sl  
SINUMERIK 840D powerline / 840DE powerline  
SINUMERIK 810D powerline / 810DE powerline

##### *Software*

HMI Advanced

##### *Software Version*

7.2

## Contents

<b>1 General Preparations.....</b>	<b>IM4/M1-5</b>
1.1 Hardware/software prerequisites .....	IM4/1-7
1.2 HMI startup .....	IM4/1-8
1.3 Limits of data management.....	IM4/1-9
1.4 Protection level concept.....	IM4/1-9
<b>2 Setting Functions and Parameters.....</b>	<b>IM4/2-11</b>
2.1 Programming PLC functions .....	IM4/2-13
2.1.1 Activating data transfer from PLC .....	IM4/2-13
2.1.2 Informing the PLC of the current task number of the HMI .....	IM4/2-16
2.1.3 Channel/spindle selection via the PLC.....	IM4/2-17
2.1.4 Block search with multiple channels .....	IM4/2-18
2.1.5 Configurable displays in the header .....	IM4/2-20
2.1.6 Cross-channel status display with icons.....	IM4/2-24
2.2 Parameterization of machine data .....	IM4/2-32
2.2.1 Work offset fine and basic offset.....	IM4/2-32
2.2.2 Machine data for setting actual value, scratching, PRESET .....	IM4/2-33
2.2.3 Immediately activate tool compensation .....	IM4/2-34
2.2.4 Defining position for machine and rotary axis position .....	IM4/2-35
2.2.5 Memory for Cycles in the DRAM.....	IM4/2-35
2.2.6 Redefining protection levels for NC data.....	IM4/2-37
2.2.7 Modifying access protection to softkey basic offset .....	IM4/2-38
2.2.8 Hiding axes .....	IM4/2-38
2.2.9 Representing the spindle utilization .....	IM4/2-40
2.2.10 Inverting the spindle icon of the spindle display .....	IM4/2-43
2.2.11 Keyboard behavior for HMI Advanced .....	IM4/2-44
2.3 Setting up access to external drives or computers.....	IM4/2-45
2.3.1 Connecting drives with display machine data .....	IM4/2-45
2.3.2 Connecting logical drives .....	IM4/2-47
2.4 Parameterizing INI files.....	IM4/2-51
2.4.1 Processing INI files .....	IM4/2-51
2.4.2 Setting the alarm log .....	IM4/2-54
2.4.3 Access authorizations when creating programs.....	IM4/2-56
2.4.4 Configuring the acknowledgement icon for PLC alarms .....	IM4/2-57
2.4.5 Configuring the "Language selection" softkey.....	IM4/2-58
2.4.6 Creating user-specific operating menus.....	IM4/2-59
2.4.7 Expanding the service overview user-specifically .....	IM4/2-66
2.4.8 Workpieces with creating job lists and using templates .....	IM4/2-68
2.4.9 Double-channel display.....	IM4/2-70
2.4.10 Tool management .....	IM4/2-71
2.4.11 Tool selection without tool management.....	IM4/2-72
2.4.12 Setting technology-specific texts.....	IM4/2-74
2.4.13 Configuring the "EXIT" softkey.....	IM4/2-76
2.4.14 Configuring the screen saver .....	IM4/2-77
2.4.15 Executing from the hard disk in M:N .....	IM4/2-77



2.5 Overview of the INI files .....	IM4/2-79
2.5.1 AEDITOR.INI .....	IM4/2-79
2.5.2 COMIC.NSK.....	IM4/2-80
2.5.3 DINO.INI .....	IM4/2-80
2.5.4 DG.INI .....	IM4/2-81
2.5.5 DGOVW.INI .....	IM4/2-82
2.5.6 DH.INI .....	IM4/2-83
2.5.7 DPDH.INI .....	IM4/2-83
2.5.8 DPSIM.INI .....	IM4/2-83
2.5.9 HEADER.INI .....	IM4/2-84
2.5.10 HMIDESK.INI .....	IM4/2-87
2.5.11 IB.INI .....	IM4/2-88
2.5.12 IF.INI .....	IM4/2-89
2.5.13 KEYS.INI.....	IM4/2-90
2.5.14 LOGDRIVE.INI.....	IM4/2-90
2.5.15 MACHINE.INI.....	IM4/2-92
2.5.16 MBDDE.INI .....	IM4/2-95
2.5.17 MMC.INI.....	IM4/2-99
2.5.18 NETNAMES.INI .....	IM4/2-103
2.5.19 OEMFRAME.INI.....	IM4/2-105
2.5.20 PARAM.INI.....	IM4/2-106
2.5.21 PARAMTM.INI .....	IM4/2-106
2.5.22 REGIE.INI .....	IM4/2-107
2.5.23 SEDITOR.INI .....	IM4/2-109
2.5.24 SIMTOGEO.INI.....	IM4/2-115
2.5.25 TASKCONF.INI.....	IM4/2-117
2.6 Simulation Data Matching .....	IM4/2-119
2.6.1 Simulation Data.....	IM4/2-119
2.6.2 Display of the Simulation Data .....	IM4/2-124
2.6.3 Expansion of the Tool Geometry Data of the Simulation .....	IM4/2-125
2.6.4 Data Match of the Simulation .....	IM4/2-128
2.6.5 Memory Configuration for the Simulation.....	IM4/2-132
2.6.6 Accelerating Simulation Startup.....	IM4/2-133
<b>3 Creating User Alarm Texts.....</b>	<b>IM4/3-135</b>
3.1 Creating alarm texts.....	IM4/3-136
3.2 Structure of the user alarms.....	IM4/3-138
3.3 Alarm texts in foreign languages.....	IM4/3-139
3.4 Plain texts for PLC machine data.....	IM4/3-141
3.5 NCU-specific alarm texts (only SINUMERIK powerline) .....	IM4/3-142
<b>4 Diagnosis and Service.....</b>	<b>IM4/4-143</b>
4.1 Installing HMI Software on a PG/PC .....	IM4/4-144
4.1.1 NCU Connection Wizard.....	IM4/4-145
4.1.2 Starting HMI Advanced in a separate desktop.....	IM4/4-145
4.1.3 Configuring the desktop switch application .....	IM4/4-147
4.1.4 Operating the desktop switch application.....	IM4/4-148
4.2 Action log .....	IM4/4-150
4.2.1 Displaying the log file .....	IM4/4-153
4.2.2 Outputting the log file .....	IM4/4-154

4.3 HMI Analyzer .....	IM4/4-155
4.3.1 Operating .....	IM4/4-156
4.3.2 Analyzing data .....	IM4/4-158
4.3.3 Examples of applications .....	IM4/4-161
<b>I Index .....</b>	<b>IM4/I-165</b>
I.1 Subject Index .....	IM4/I-165
I.2 Index of Commands and Identifiers .....	IM4/I-167



# 1

## 1 General Preparations

1.1 Hardware/software prerequisites .....	IM4/1-7
1.2 HMI startup .....	IM4/1-8
1.3 Limits of data management .....	IM4/1-9
1.4 Protection level concept.....	IM4/1-9



## 1.1 Hardware/software prerequisites

### Overview

This manual describes the procedure for performing a system startup of the PCU 50 with the HMI Advanced software.

For further reading on special NCK, HMI, PLC, or drive functions, refer to the applicable descriptions of functions and manuals (see "Documentation").

### Software

Depending on the order details, the HMI Advanced software may be loaded on the PCU 50 at the time of delivery. If the HMI Advanced software is not pre-loaded on the PCU 50 (e.g., PCU 50 is supplied without system software), the HMI software can be installed on the PCU 50 from the CD supplied using the Service menu.

To install the HMI Advanced software on the PCU 50, you need the following:

- PC or programming device with CD drive
- Network connection
- USB memory

This also applies to SW updates (see "SW Installation/SW Update").

### Operating system

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

---

#### Notice

Installing on a PC/PG:

HMI-Advanced can only be run under a non-administrator user if this user has write privileges for the directory mmc2 of HMI Advanced. The installation of HMI Advanced does not provide these privileges.

---

### Documentation

For system startup of the SINUMERIK controller, you need the following documentation:

- Operator Component Manual
- Operator's Guide for HMI Advanced
- Diagnostics Guide
- Description of Functions for the basic machine (Part 1)
- Commissioning Manual for the CNC Part 5 basic software
- Parameter Manual

## 1.2 HMI startup

### Overview

The actions executed during HMI startup are controlled by the contents of initialization files.

Notes on other entries can be found in the OEM documentation

**Reference:** HMI OEM package

For SINUMERIK powerline, the HMI Advanced startup can be modified for M:N configuration in the NETNAMES.INI file. See Section "Overview of INI Files" and

**Reference:** /FB2/ B3 Multiple Operator Panel Fronts and NCUs

### Startup settings

The following functions can be modified for HMI startup in the "Startup" operating area:

- Language selection
- MPI/OPI address setting for SINUMERIK powerline
- Setting the IP addresses for SINUMERIK solution line

### Errors during HMI Startup

The following error can occur when the PCU 50 is started up:

No connection to NCU/CCU: Alarm 120202 is output.

For SINUMERIK powerline, the PCU 50 is linked to the NC and the PLC via a serial bus; for SINUMERIK solution line, the PCU 50 is linked to the NC and the PLC via Ethernet.

This alarm occurs if HMI Advanced is started for the first time and the NC/PLC has not yet finished starting up or communication with these components is faulty. When this alarm occurs, all display values connected with NC/PLC become invalid.

Such faults are normal while the controls are starting up (e.g., after resetting). The alarm disappears automatically as soon as the fault situation is resolved. If the alarm persists, a wide variety of causes may be to blame (e.g., open circuit, NC/PLC not started up, incorrect address/baud rate configuration on one of the bus nodes, etc.).

## 1.3 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 1,000 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 1,000 files, e.g., a maximum of 1,000 archives in the archive directory. The following limit applies to network drives: maximum of 1,000 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 5,000 customized alarms/messages can be created in each language.

## 1.4 Protection level concept

### Overview

A protection level concept for enabling data areas is implemented in SINUMERIK 840D. Protection levels 0 to 7 are available, where 0 represents the highest protection level and 7 the lowest protection level.

Protection levels

- 0 to 3 are locked by means of a password
- 4 to 7 are locked by means of key switch positions

The operator only has access to information corresponding to this particular protection level and the levels below it. Machine data are routinely assigned different protection levels.

Protection level 4 (key switch position 3) is the minimum level required to display machine data. For system startup, the appropriate protection level can generally be enabled by entering the password "SUNRISE".

For information about changing protection levels, refer to

**Reference:** /BAD/, HMI Operator's Guide  
/FB/, A2, Various Interface Signals

Protection levels 0 to 3 require the input of a password. The password for protection 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 reset). This resets all passwords to the defaults for this software version.

### Protection level 0-3

The password remains valid until it is reset with the "Delete Password" softkey.

POWER ON does not reset the password.

### Protection levels 4-7

Protection levels 4 to 7 require a corresponding key 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.

<b>Table      Meaning of keyswitch positions</b>		
<b>Key color</b>	<b>Key switch position</b>	<b>Protection level</b>
All (no key used)	0 = Key removal position	7
Black	0 and 1	6-7
Green	0 to 2	5-7
Red	0 to 3	4-7





# 2

## 2 Setting Functions and Parameters

2.1 Programming PLC functions.....	IM4/2-13
2.1.1 Activating data transfer from PLC.....	IM4/2-13
2.1.2 Informing the PLC of the current task number of the HMI.....	IM4/2-16
2.1.3 Channel/spindle selection via the PLC.....	IM4/2-17
2.1.4 Block search with multiple channels.....	IM4/2-18
2.1.5 Configurable displays in the header.....	IM4/2-20
2.1.6 Cross-channel status display with icons.....	IM4/2-24
2.2 Parameterization of machine data.....	IM4/2-32
2.2.1 Work offset fine and basic offset.....	IM4/2-32
2.2.2 Machine data for setting actual value, scratching, PRESET.....	IM4/2-33
2.2.3 Immediately activate tool compensation.....	IM4/2-34
2.2.4 Defining position for machine and rotary axis position.....	IM4/2-35
2.2.5 Memory for Cycles in the DRAM.....	IM4/2-35
2.2.6 Redefining protection levels for NC data.....	IM4/2-37
2.2.7 Modifying access protection to softkey basic offset.....	IM4/2-38
2.2.8 Hiding axes.....	IM4/2-38
2.2.9 Representing the spindle utilization.....	IM4/2-40
2.2.10 Inverting the spindle icon of the spindle display.....	IM4/2-43
2.2.11 Keyboard behavior for HMI Advanced.....	IM4/2-44
2.3 Setting up access to external drives or computers.....	IM4/2-45
2.3.1 Connecting drives with display machine data.....	IM4/2-45
2.3.2 Connecting logical drives.....	IM4/2-47
2.4 Parameterizing INI files.....	IM4/2-51
2.4.1 Processing INI files.....	IM4/2-51
2.4.2 Setting the alarm log.....	IM4/2-54
2.4.3 Access authorizations when creating programs.....	IM4/2-56
2.4.4 Configuring the acknowledgement icon for PLC alarms.....	IM4/2-57
2.4.5 Configuring the "Language selection" softkey.....	IM4/2-58
2.4.6 Creating user-specific operating menus.....	IM4/2-59
2.4.7 Expanding the service overview user-specifically.....	IM4/2-66
2.4.8 Workpieces with creating job lists and using templates.....	IM4/2-68
2.4.9 Double-channel display.....	IM4/2-70
2.4.10 Tool management.....	IM4/2-71
2.4.11 Tool selection without tool management.....	IM4/2-72
2.4.12 Setting technology-specific texts.....	IM4/2-74
2.4.13 Configuring the "EXIT" softkey.....	IM4/2-76
2.4.14 Configuring the screen saver.....	IM4/2-77
2.4.15 Executing from the hard disk in M:N.....	IM4/2-77

2.5 Overview of the INI files .....	IM4/2-79
2.5.1 AEDITOR.INI .....	IM4/2-79
2.5.2 COMIC.NSK .....	IM4/2-80
2.5.3 DINO.INI .....	IM4/2-80
2.5.4 DG.INI .....	IM4/2-81
2.5.5 DGOVW.INI .....	IM4/2-82
2.5.6 DH.INI .....	IM4/2-83
2.5.7 DPDH.INI .....	IM4/2-83
2.5.8 DPSIM.INI .....	IM4/2-83
2.5.9 HEADER.INI .....	IM4/2-84
2.5.10 HMIDESK.INI .....	IM4/2-87
2.5.11 IB.INI .....	IM4/2-88
2.5.12 IF.INI .....	IM4/2-89
2.5.13 KEYS.INI .....	IM4/2-90
2.5.14 LOGDRIVE.INI .....	IM4/2-90
2.5.15 MACHINE.INI .....	IM4/2-92
2.5.16 MBDDE.INI .....	IM4/2-95
2.5.17 MMC.INI .....	IM4/2-99
2.5.18 NETNAMES.INI .....	IM4/2-103
2.5.19 OEMFRAME.INI .....	IM4/2-105
2.5.20 PARAM.INI .....	IM4/2-106
2.5.21 PARAMTM.INI .....	IM4/2-106
2.5.22 REGIE.INI .....	IM4/2-107
2.5.23 SEDITOR.INI .....	IM4/2-109
2.5.24 SIMTOGEO.INI .....	IM4/2-115
2.5.25 TASKCONF.INI .....	IM4/2-117
2.6 Simulation data matching .....	IM4/2-119
2.6.1 Simulation data .....	IM4/2-119
2.6.2 Display of the simulation data .....	IM4/2-124
2.6.3 Expansion of the tool geometry data of the simulation .....	IM4/2-125
2.6.4 Data match of the simulation .....	IM4/2-128
2.6.5 Memory Configuration for the Simulation .....	IM4/2-132
2.6.6 Accelerating Simulation Startup .....	IM4/2-133

## Overview

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

- Programming the PLC
- Parameterizing machine data (MD)
- Parameterizing INI files

## 2.1 Programming PLC functions

### 2.1.1 Activating data transfer from PLC

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 lists" function in HMI Advanced enables the full range of part program handling operations, such as loading, unloading, selecting or executing, from the hard disk.

**References** relating to job lists:     /IAD/ Installation and Start-Up Guide 840D  
  /BAD/ Operator's Guide HMI Advanced

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 (hard disk) and NCK.

Specify the user control file:

Table 2-1 DB19.DBB16 (PLC → HMI)

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
always 1 (appr. file system)	PLC index for the user control file Value: 1 - 127						

Specify the job list:

Table 2-2 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:

Table 2-3 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:

Table 2-4 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

## Job processing

A PLC job is processed according to the following scheme:

1. The PLC may only initiate a job in the job byte if the acknowledgment byte is 0.
2. 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.
3. Once the operation is finished (OK or Error), the PLC must respond again and delete the job byte.
4. In response, the HMI sets the acknowledgment byte to 0.

This scheme ensures sequential processing of jobs.

## 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, where "xxx" represents a number between 001 and 127. The file contains a maximum of 127 references to job lists.

The user must create this 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

Requirement:

- Existing file PLC\_IN\_OUT\_003.TEA
  - N1 job list A
  - N2 job list B
  - N3 job list C
- Acknowledgment byte DB19.DBB26 is 0.
- Define PLC index for the control file (DB 19, DBB16 = 131)
- Define PLC line offset 2 (DB 19, DBB17 = 2)
- PLC job: Load (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 messages to the PLC**

The HMI outputs the following error values to the PLC via DB19.DBB27:

- 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 response to the AR server can be entered in the MASCHINE.INI configuration file.

The default setting is "yes", i.e., all prompts are responded to with Yes. If "no", all prompts are responded to with No.

```
[PLC_REQUEST]
DefaultAnswer=yes
```

**Multiple operator panel fronts/NCUs M:N (for 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_LEFT
```

This setting for the netmaster NCU/PLC means that data transfer is only activated by the NCU/PLC with the symbolic name NCU\_LEFT.

**Machine manufacturer**

See machine manufacturer's specifications.

---

For a description of DB 19, refer to **/FB/ A2**, Various Interface Signals

## 2.1.2 Informing the PLC of the current task number of the HMI

### Task number

A task describes the position at which an operating area is attached in the HMI area menu, i.e., the highest level.

The following fixed relationship exists between softkey bars of the area menu and tasks:

First softkey bar:

Softkey	1	2	3	4	5	6	7	8
Task number	0	1	2	3	4	5	6	7

Second softkey bar:

Softkey	1	2	3	4	5	6	7	8
Task number	8	9	10	11	12	13	14	15

Third softkey bar:

Softkey	1	2	3	4	5	6	7	8
Task number	16	17	18	19	20	21	22	23

Which application is assigned to which task number is stored/configured in the [TaskConfiguration] section in the REGIE.INI file (see 5.6.14 REGIE.INI).

A fourth softkey bar can be used for OEM applications.

Fourth softkey bar:

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 operator panel and the displayed operating areas/screens is essential.

Example:

The "Move unit" function is selected in a screen form. On the MCP are "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, due to an operating area switchover in the HMI, for example, 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 machine data 9032: HMI\_MONITOR as follows:

DBxx.DBByy

where:

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 takes place 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 numbers and byte area are not checked.

## Updating

The task number in the first byte is updated:

- On 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.

## Value range

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.

## 2.1.3 Channel/spindle selection via the PLC

### Channel selection

Channel selection and display is initiated as follows (display in "Channel name" field):

```
DB 19, DBB 32 = 01000001 [binary] Channel selection
DB 19, DBB 33 = <channel number> 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 no. 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	Meaning
0	No error
1	No function number (DBB32 bit 0-5)
2	Invalid parameter
3	Error when writing HMI-internal variable
10	Channel not present (DBB33)

### Acknowledgment mechanism between HMI ↔ PLC

Via DB 19.DBB32 bits 6, 7, there is an acknowledgment mechanism that ensures that the HMI and PLC agree on 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 "0".
- After displaying the text in the header the HMI writes "0" again in bit 6 and the interface is enabled for the PLC for the next action.

---

#### 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. 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.

---

## 2.1.4 Block search with multiple channels

### Function

For this block search, the NC is operated in program test mode, so that interactions between channel and synchronous actions and between multiple channels are possible within an NCU.

During "Block search in program test mode", all auxiliary functions are output from the NC to the PLC and the part program commands for channel coordination (WAITE, WAITM, WAITMC, Replace axis, Write variables) are carried out by the NC. That is:

- The PLC is updated during this block search
- Machining operations involving interaction among multiple channels are executed properly within the block search.



## Activation

The function is activated by the "Prog. test contour" softkey. The NC 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 NC 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 NC 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.

## Dialog message

A dialog message appears as soon as all the channels involved in the current search have exited program test mode. This 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. You must acknowledge this message.

Any Repos offsets (in the individual channels) can be traversed 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.

## 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 NC

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 by means of the MACHINE.INI file.  
The following entries are possible in the [BlockSearch] section:

### [BlockSearch]

SeruproConfig

- =1 Block search in **currently selected** channel (default)
- =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.

## Deactivating function

The following entries are possible in the [BlockSearch] section:

### [BlockSearch]

SeruproEnabled

- =0 Function inactive
- =1 Function active (default)

Changes take effect immediately, i.e., with the next search.

## 2.1.5 Configurable displays in the header

Using the PLC, 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). See also Alarm Display.

### Text length

The maximum length of the texts to be displayed is 30 characters per line.

### Syntax

The texts can consist of:

- A language-specific part (from alarm/message text file \*.COM)
- Three parameters (%1, %2, %3)

### User PLC-DB

The display must be configured via a user PLC-DB.

- 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
```

where x = data block number, y = start byte in the block.

If this interface is specified, the text length limitation  
Textlength = 33  
must also be activated.

- The name of the PLC alarm text file (e.g., "alpu\_") must 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 user DB interface provides

- A start data byte (job detection by HMI), and
- Secondary bytes, with which
  - a number,
  - 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. Inputs after this "00" entry are not displayed.

Table 2-5: Interface of the user-PLC-DB for displays for program path/program name

Program path line	Start DBBx	Start DBWx + 1	Start DBBx + 3	Start DBBx + 4	Start DBBx +5-17
<b>Data type</b>	Byte	Word	Byte	Byte	String
<b>Meaning</b>	Bit 0=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 from alpu_xx.com to be displayed, 700000 is base address and is always summed internally =0: Delete line	Number from 0 to 255 (corresponds to 1st parameter %1)	Index from -127 ...128 negative number (-1=0xFF, -2=0xFE) is machine axis index, positive number is channel axis index. The index is then converted to an axis name before being displayed. (corresponds to 2. parameter %2)	Character string max. 12 characters (corresponds to 3rd parameter %3)
<b>Program name line</b>	<b>Start DBBx +20</b>	<b>Start DBWx +21</b>	<b>Start DBBx +23</b>	<b>Start DBBx +24</b>	<b>Start DBBx +25-37</b>
<b>Data type</b>	Byte	Word	Byte	Byte	String
<b>Meaning</b>	As above	As above	As above	As above	As above

<b>Start DBBx</b>	HMI detects a job and acknowledges it (all bytes=0).
<b>Start-DBWx +1</b>	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.
<b>Start-DBBx +3</b>	Start DBBx+3 contains a number that is displayed without being converted.
<b>Start-DBBx +4</b>	<p>Start DBBx+4 contains an axis index.</p> <ul style="list-style-type: none"> <li>• Negative value = machine axis index (AXCONF_MACHAX_NAME_TAB[ Index-1], MD 10000),</li> <li>• Positive value = channel axis index (AXCONF_CHANAX_NAME_TAB[ Index-1], MD 20080),</li> </ul> <p>These indices always refer to the channel currently displayed in the HMI. The index is converted to an axis name before being displayed.</p>
<b>Start-DBB +5</b>	Start DBBx +5–17 contains a character string comprising a maximum of 12 characters.

---

**Note**

Bit 0=1 can only be set in Start DBBx for a new job if

- All bytes were reset to zero after the preceding job
  - Parameters for the new job Start DBBx+1 to Start DBBx+5 have been set correctly.
  - 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...).
- 

**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 agree:

- 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 "0".
- After displaying the text in the header, the HMI writes "0" again in Bit 0 and the interface is enabled for the PLC for the next action.

## Response in M:N (SINUMERIK powerline only)

In the M:N function the two text lines must be cleared in passive switching. In active switching, as in area switching, the content of the variables is evaluated and displayed.

## Task

The following texts should be output in the "Program name" field in the header of JOG and MDA modes:

- "Handwheel acting on axis ..." in the first line with the appropriate axis name
- "Slide ... active" in the second line with the appropriate slide number.

## Procedure

1. Identify the PLC-HMI interface in the HEADER.INI file, e.g., DB60.DBB10
2. Enter the name of the PLC user text file in MBDDE.INI,  
e.g.  
[TextFiles]  
UserPLC=alpu
3. Enter texts in the PLC user text file,  
e.g.  
700100            0        0        "Handwheel acting on axis %2"  
700101            0        0        "Slide %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

With the following axis configuration  
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 as follows:

**"Handwheel acting on Y axis"** (first display line)  
**"Slide 1 active"** (second display line)

## 2.1.6 Cross-channel status display with icons

### Function









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

- Channels
- Spindles
- Machine states
- Machine states only

In addition to SINUMERIK icons, manufacturer-defined icons can also be used (configurable using the HEADER.INI file and DB19 DBB32-34).

### SINUMERIK icons

The following SINUMERIK icons are available (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 counterclockwise	(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

**Colors:** TrueColor

**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.

**Activation**

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:

<b>9052</b>	<b>SHOW_CHANNEL_SPANNING_STAT</b>		
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	Unit: -	
Data type: STRING	Applies from version 6.2		
Meaning	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 **four 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 counterclockwise 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 included (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 output deliberately, for example). 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/or user icons.



**Notes:**

- 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 HEADER.INI file is reproduced in Subsection 2.5.9.

**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								

Fig. 2-1: Program Status Line (1)

The SINUMERIK icons are permanently assigned to an output position. A section is created in the HEADER.INI configuration file for each position used.

**Sample view** of the program status line with SINUMERIK icons for 4 channels and 2 spindles:

1	2	3	4	S1	S2										
SKP1	DRY	ROV	SBL1	M01	M17	DRF	PRT								

Fig. 2-2: Program Status Line (2)

These displays always have the predefined channel or spindle number.

**Example:**

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 disable for channel 3 and
                       ; spindle disable for spindles 1 and 2
[Pos3]                ; third display position
```

```
ChanStatFeedStop = 2 ; channel status of channel 2 is displayed
                    ; superimposed with feed disable for 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:



Fig. 2-3: Program Status Line (3)

## Deactivation

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>
```

## User status display

Additionally (or alternatively), PLC-controlled (DB x, DBB y) user icons linked to specific machine states can be output.

Procedure:

- Declaration of icons and positions (1.)
- Control of icon call by PLC (2.)

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:           Identifier
      ; 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
```

Further on in the UserIcons section:

```
DBx.DBBy:   User-defined double word for
             control of the icon selection. If a bit is set for
             a user icon, the associated
             UI_x entry must exist, otherwise nothing is displayed.
```

To display an icon according to the above declaration, the PLC sets the corresponding bit in the double word.

For example:

UI\_0 Display assigned icon → Bit 0

...

UI\_31 Display assigned icon → 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 "Icon\_01.bmp" icon is displayed at the 12<sup>th</sup> position, provided another icon with a higher identifier (>UI\_3) and the same position (12) is not active.

### Possible applications

With **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, see Superimposition

### Superimposition

With **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 in the header

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.

Maschine	Chan1Maschine Chan1MSTT	JOG Inc 100	OEM_Text 001 OEM_Text 004	OEM_Text 002 OEM_Text 005	OEM_Text 003 OEM_Text 006
Kanal RESET			1 // 2 // 2 //	1 // 2 // S2 //	1 //

Fig. 2-4: 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.

```
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	Bit 0=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, MO1, 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.

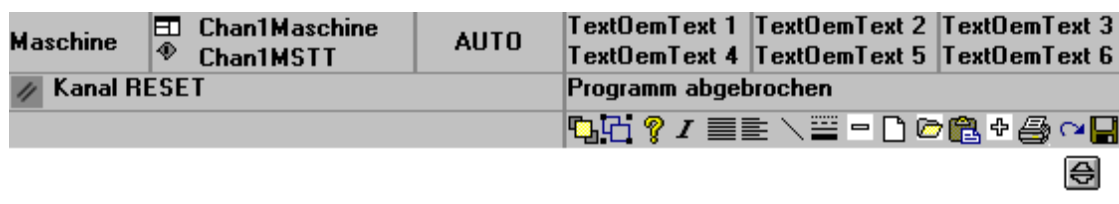


Fig. 2-5: Example for OEM Icons

## 2.2 Parameterization of machine data

### 2.2.1 Work offset fine and basic 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	Amount of change

#### Actual value display: settable zero system

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)
- 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 settable zero system, SZS (take-up position of active tool)

#### Example:

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



---

#### Machine manufacturer

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

---

## 2.2.2 Machine data for setting actual value, scratching, PRESET

### Function

The behavior of the **Scratching**, **Set actual value**, and **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):

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

#### Variant 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.

#### Variant 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:** /BAD/, Operator's Guide 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 des Basisframes.

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.

## 2.2.3 Immediately activate tool compensation

### Function

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.

**Reference:** /FB/ Description of Functions Basic Machine K2 Axes, Coordinate Systems ...



---

### Caution

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

---



## 2.2.4 Defining position for machine and rotary axis position

### 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	Number base for display of moving joint position STAT
MD 9243: MA_TU_DISPLAY_BASE	Number base for display of rotary axis position TU

Possible settings 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'

## 2.2.5 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 "Commissioning" → "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 in the DRAM

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 on startup.

---

### 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.**

---

## 2.2.6 Redefining protection levels for NC data

### Function

The preset protection levels can be modified in the definition file. For machine data, only lower protection levels are permitted as the configured protection level; for setting data, higher levels are also permitted. The file becomes active when the next `_N_INITIAL_INI` is read in. Different protection levels are specified for writing or reading (part 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
                                        ; (APR ... Read access)
REDEF $MA_ENC_SEGMENT_NR APR 2 APW 2
                                        ; (APW ... Write access)
REDEF $SN_JOG_CONT_MODE_LEVELTRIGGRD APR 2 APW 2
M30
```

MD 30100: `$MA_CTRLOUT_SEGMENT_NR` has protection level 2/7, i.e., protection level 2 is required for write access (corresponding to password) and protection level 7 for read access. In order to access the machine data area, key switch position 3 or higher is required.

### Undoing the protection level change

In order to undo protection 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
```

### Compatibility

#### Note

If the access to machine data that was authorized by the command "REDEF <machine data> APR <protection level> APW <protection 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`.

## Protection levels for NC commands

Certain NC commands can be linked to protection 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 switch position 0 to 3 or a password for end users as far as Siemens.

**Reference:** /PGA/, Chapter "Protection levels for NC commands".

## 2.2.7 Modifying access protection to softkey basic offset

### Parameters operating area

The 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/hidden in the "Work offset" window and in the "Active WO + compensations" window.

### Machine operating area

The 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.

## 2.2.8 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:

Bit 0 to 15	Show (=1)/hide (=0) geometry axes (WCS)
Bits 16 to 31	Show (=1)/hide (=0) machine axes (MCS)

**Reference point approach and Safety Integrated (special cases)**

Set machine data 20098: DISPLAY\_AXIS is not used for the displays in the Homing 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.

<b>20098</b>	<b>DISPLAY_AXIS</b>		
MD number	<b>Display or hide axis on the HMI</b>		
Default setting: 0xFFFFFFFF	Minimum input limit: 0	Maximum input limit: 0xFFFFFFFF	
Changes effective after POWER ON	Protection level: 2/7	Unit: -	
Data type: DWORD	Applies from version 4.4		
Meaning:	<p><b>Bits 0 to 15: WCS</b></p> <p>Bit 0= 1 Show geometry axis in actual-value windows 0 Hide geometry axis in actual-value windows</p> <p>Bit 1 1 Show geometry axis in reference point window 0 Hide geometry axis in actual-value windows</p> <p>Bit 2= 1 Show geometry axis in Preset/Basic offset/Scratch/Work offset window 0 Hide geometry axis in Preset/Basic offset/Scratch window</p> <p>Bit 3= 1 Show geometry axis in Handwheel selection window 0 Hide geometry axis in Handwheel selection</p> <p><b>Bits 16 to 31: MCS</b></p> <p>Bit 16= 1 Show machine axis in actual-value windows 0 Hide machine axis in actual-value windows</p> <p>(Bit 17) Not assigned</p> <p>Bit 18= 1 Show machine axis in Basic offset window 0 Hide machine axis in Basic offset window</p> <p>Bit 19= 1 Show machine axis in Handwheel selection window 0 Hide machine axis in Handwheel selection window</p>		

## 2.2.9 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 utilization to 200% using display MD 9429: MA\_SPIND\_POWER\_RANGE.

<b>9428</b>	<b>\$MM_MA_SPIND_MAX_POWER</b>		
MD number	Maximum value of the spindle performance display		
Default setting: 100	Minimum input limit: 100	Max. input value: ****	
Changes effective after: Power ON	Protection level: 3/4	Unit: %	
Data type: WORD	Applies from version 6.4		
Meaning	In this MD, enter the factor by which the supplied spindle utilization will be multiplied.		

<b>9429</b>	<b>\$MM_MA_SPIND_POWER_RANGE</b>		
MD number	Display range for spindle utilization		
Default setting: 100	Minimum input limit: 100	Max. input value: ****	
Changes effective after: Power ON	Protection level: 3/4	Unit: %	
Data type: WORD	Applies from version 6.4		
Meaning	<p>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.</p> <p><b>Value 100:</b> 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.</p> <p><b>Value &gt; 100, e.g., 200:</b> Percentage values 0%, 100%, and 200% are displayed. The color display changes from green to red starting at 100%.</p> <p>If you set the value to 200%, you must enter the normalization factor in display MD \$MM_MA_SPIND_MAX_POWER.</p>		

## 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

1. Open the following window in the "Commissioning" → "Optimization/Test" → "Function Generator" operating area:

The screenshot displays the 'Function generator' configuration window. At the top, it shows 'Start-up' with 'K\_1' and 'JOG Ref' with 'MPF.DIR TEST1.MPF'. Below this, there are 'Channel reset' and 'Program aborted' indicators. The main configuration area is divided into several sections: 'Drive test travel enable' (set to 'Without PLC'), 'Travel range' (with 'Monitoring' set to 'Inactive', 'Upper limit' at '0.000 deg.', and 'Lower limit' at '0.000 deg.'), and 'Mode' (with 'Signal' set to 'Speed setpoint', 'Signal type' set to 'Square-wave', and 'Scaling' set to '100'). A 'Status' section shows 'Inactive' and 'Absolute position: 0.001 deg.'. On the right side, there are vertical buttons for 'Axis +', 'Axis -', 'Direct selection', 'Start', and 'Stop'. At the bottom, a navigation bar contains buttons for 'Measurement', 'Signal parameter', 'Axis-spec. MD', 'Drive MD', 'User views', and 'File functions'.

Fig. 2-6: Function generator

2. In the screen form that appears, select a signal, a signal type and a factor for standardization 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

Open the Trace function main screen by pressing the “Drive/servo” softkey followed by the “Servo trace” softkey in the “Startup” operating area.

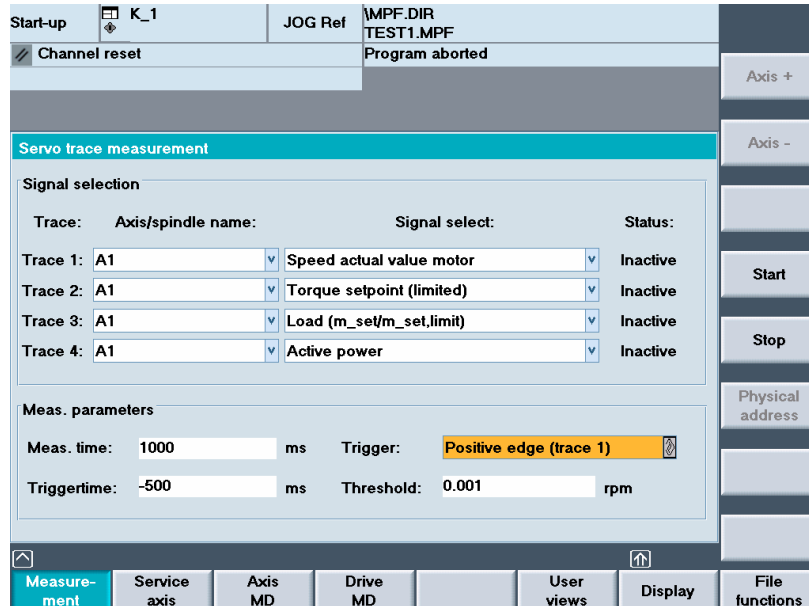


Fig. 2-7: Basic display servo trace

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.

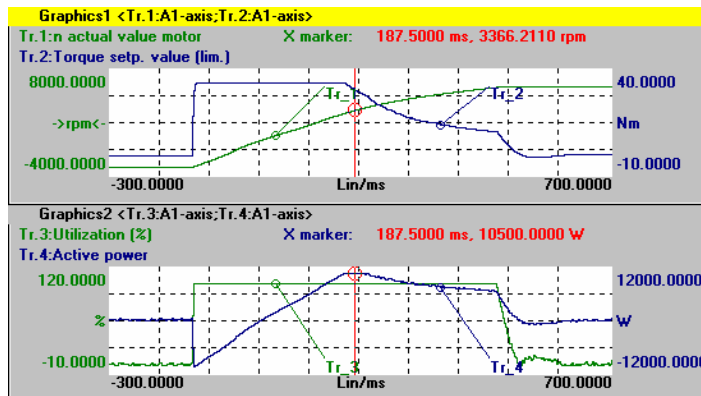


Fig. 2-8: Servo trace graphic

Select the constant power range or constant torque range in order to calculate the normalization factor using these data.

Example: In Fig. 2-8 the measured power is 10.5 kW.



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:

$$\text{MA\_SPIND\_MAX\_POWER} = \text{Maximum\_of\_the\_measured\_power} / \text{rated power\_S1} * 100$$

$$\text{MA\_SPIND\_MAX\_POWER} = 10500\text{W} / 7000\text{W} * 100 = 150$$

Enter the factor "150" in the machine data.

## 2.2.10 Inverting the spindle icon of the spindle display

### Function

The display of the spindle symbol 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).

<b>9033</b>	<b>MA_DISPL_INVERT_DIR_SPIND_M3</b>		
MD number	Spindle rotation direction display		
Default value: 0x0000	Min. input value: 0x0000	Max. input value: 0x7FFFFFFF	
Change effective after: IMMEDIATE	Protection level (R/W) 3/4	Unit: -	
Data type: LONG	Applies from version 6.2		
Meaning	Spindle rotation direction display: Bit[n]=0            M3 displayed as clockwise rotation in the icon Bit[n]=1            M3 displayed as counter-clockwise rotation in the icon n = spindle number – 1 (i.e. bit 0 is spindle S1) Note: With the corresponding setting for MD 9014: USE_CHANNEL_DISPLAY_DATA MD 9033 can be managed channel-specifically in the NC.		

## 2.2.11 Keyboard behavior for HMI Advanced

### CAPSLOCK (PCU 50)

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 standard Windows NTXP operation.

#### Activation

This function is activated by display MD 9009: \$MM\_KEYBOARD\_STATE

0: CAPSLOCK off  
2: CAPSLOCK on

This display MD is only evaluated on during startup. If the MD is changed, the system must be restarted.

#### Operation

If display MD 9009 is set to "CAPSLOCK on" and you want to enter lower case text, proceed as follows:

Press CTRL + SHIFT to switch to lower case text entry. To reset text entry from lower case back to upper case, press CTRL + SHIFT again.

#### 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.

---

## 2.3 Setting up access to external drives or computers

### Application

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.

### 2.3.1 Connecting drives with display machine data

#### Configuring drives/computers

A maximum of four network drives can be furnished 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 paths (e.g. "E:" or "G:\VORBERE\MPF\_DAT" or as network addresses consisting of node identifiers, share name and directory, e.g. "\\R111\

#### Example:

MD9677 CMM\_DIRECTORY\_SOFTKEY\_PATH2=F:\R4711

; Drive + directory

MD9678 CMM\_DIRECTORY\_SOFTKEY\_PATH3=\\R4712\PART1\SHAFT1.DIR

; Computer name, share name with directory

MD9679 CMM\_DIRECTORY\_SOFTKEY\_PATH4=\\R4713\Part2\MPF\_DAT.DIR

; Computer name, share name with directory

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 authorizations for drives/computers

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

- For the **Program** and **Services** areas:  
\$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 during HMI Advanced startup. 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.

## 2.3.2 Connecting logical drives

### 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 "Logical drives" under "Commissioning" → "NC".
- By configuring the LOGDRIVE.INI file (see Subsection 2.5.14).

Configuring a logical drive includes the following specifications:

- Path name with structure:  
"acttcu:/<TCU connector>,<Partition>/<Path name>"
- For USB on a TCU: name, 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, see Subsection 2.3.1 Connecting drives with display machine data).

### Requirements

The "logical drives" softkey is available after access level 2 (Service): For this reason, this function is carried out by a service technician rather than machine operators. A restart of HMI Advanced is necessary to update the assignment of the softkeys.

### "USB local" softkey

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.

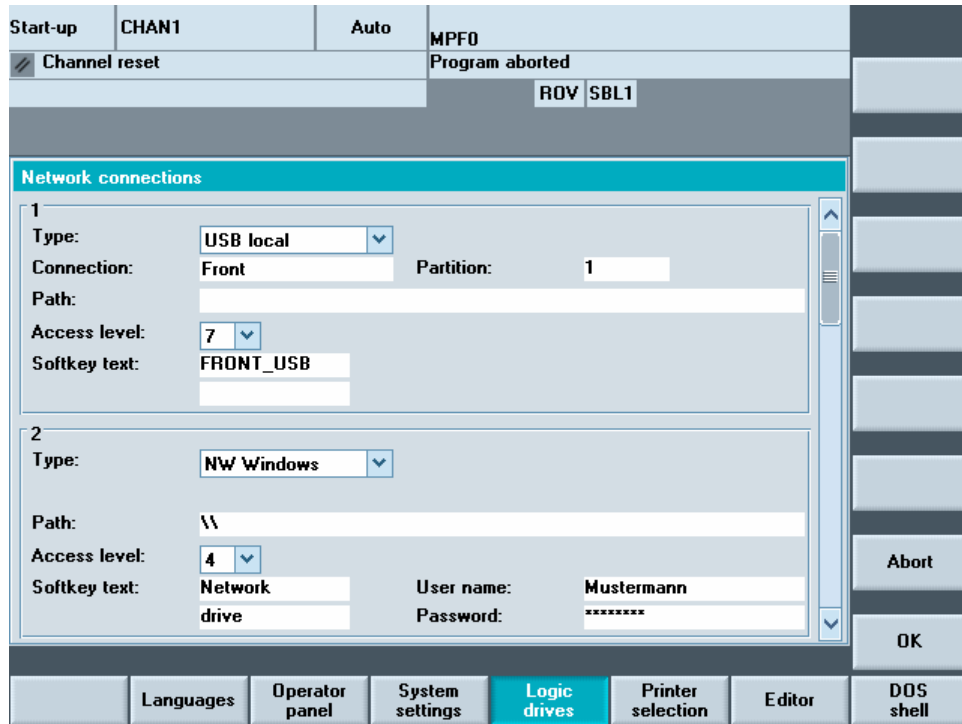


Fig. 2-9: "Logical Drives" setting screen

**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

<b>NW Windows:</b>	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.
<b>Local drive:</b>	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. Two blanks or \n or %n are acceptable line separators.

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 from and to 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.

---

---

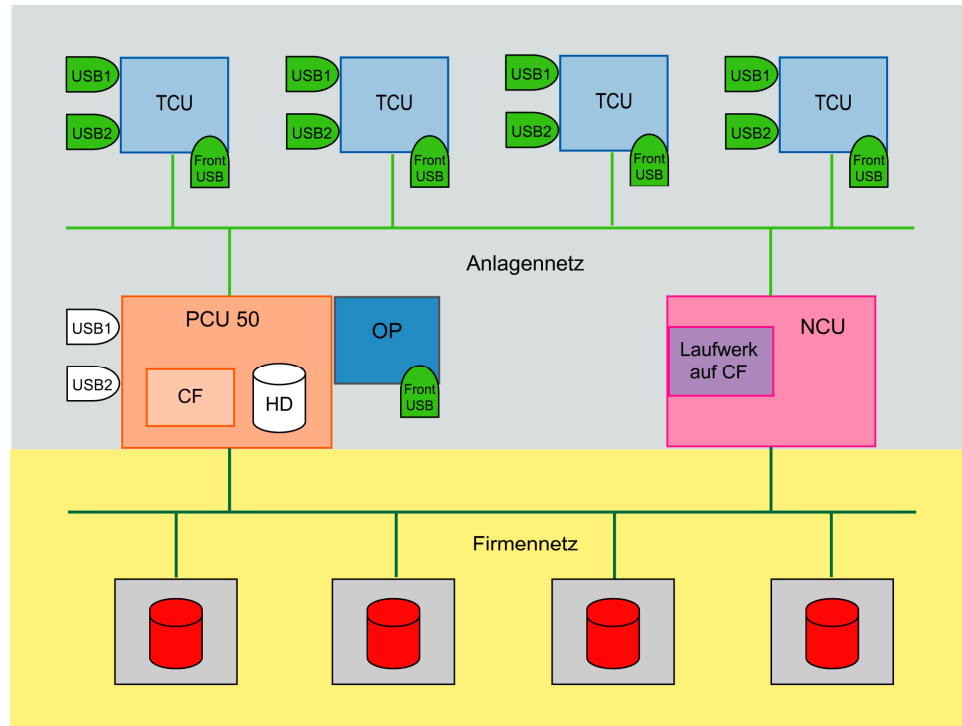
#### Caution

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

---

### "Network drives" option

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



**Legend:**

- Usable without option: all USB interfaces at the TCU
- Usable without option: locally usable
- Locally usable without option; "Remote" only usable with network drives option
- Only usable with network drives option
- Only usable with HMI user memory option, locally usable

Fig. 2-10: Network Drives Option



## 2.4 Parameterizing INI files

### 2.4.1 Processing INI files

#### Notation

[xxx]	; Identification symbol for 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 ; that can be assigned.
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.

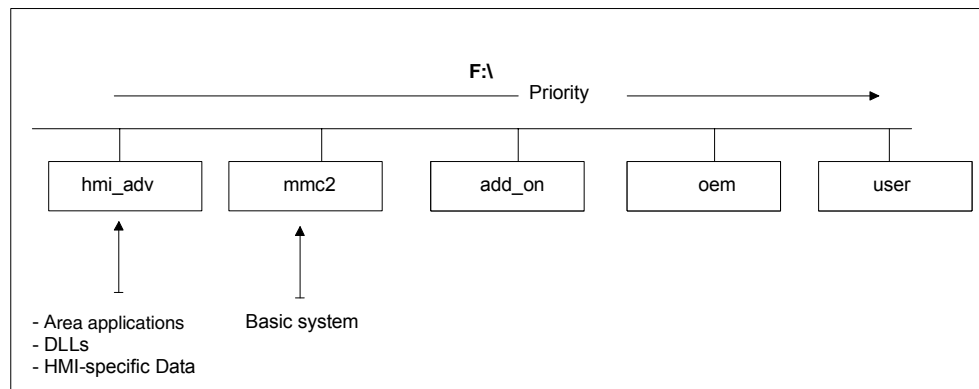


Fig. 2-9: Priority

The directories contain the following contents:

- **hmi\_adv:**  
The directory contains area applications, DLLs and HMI-specific data.
- **mmc2:**  
System directory for basic HMI software.
- **add\_on:**  
Directory for additional Siemens products (e.g., TPM, MDA, DNC, etc.). This can have the same directory structure (e.g. language subdirectory) and directory contents as mmc2. Entries for the 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., language subdirectory) and directory contents as mmc2. Entries for the 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 **user** directory, 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 **user** file.

In principle, only **entry differences** in the ini files as compared with the originals in mmc2 are stored in the parallel directories to mmc2. For example, if an OEM application is added to "oem", a complete "regie.ini" is not created; only the modified sections are stored in "oem".

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

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

---

#### Note

None of the INI files in the "mmc2" and "hmi\_adv" directory may be modified.

---

### Procedure for user-specific entries

1. 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).
2. 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 MBDDE.INI for PLC error message/operating message and scrolling through alarm lines

MBDDE.INI in USER:  
;Section header

#### [TextFiles]

UserPLC = F:\DH\MB.DIR\MYPLC\_

#### [Alarms]

;Scroll through alarm lines  
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 \OEM

---

#### Notice

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

---

### Switching off settings

If there are specific entries in a lower-priority directory that you wish to be disabled in a higher-priority directory, 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 KB. 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.

## 2.4.2 Setting the alarm log

### Application

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 started up. 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 be selected by means of the DiskCare Ini file entry

File: "mbdde.ini",  
Section: [PROTOCOL],  
Entry: "DiskCare".

The following values are possible:

- |             |   |
|-------------|---|
| DiskCare=-1 | The MBDDE server controls the alarm log in the memory. The alarm log is saved to the hard disk if it is displayed in the diagnostics operating area or if the Alarm Cancel key is actuated.   |
| DiskCare=0  | The data is written to the log file immediately.  |
| DiskCare=n  | Changes in the alarm status are written to the log file if no change occurs within n seconds. In addition, the alarm log is saved to the hard disk if it is displayed in the diagnostics operating area or if the Alarm CANCEL key is actuated. |
| DiskCare=-n | n>1 specifies the number of parallel alarm files.   |

The file "mbdde.ini" is evaluated on startup. 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 on startup 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 value: 150  
Minimum value: 18  
Maximum value: 32000

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

## Selection of 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 constructed 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, etc.)
- From Source of alarm
- AckVar Acknowledgment variable

#### RELATION:

- '=' Equal to
- '<' Less than
- '>' Greater than
- '!' Not

#### FEATURE:

- Numbers
- 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 from NCU\_1 are logged

### 2.4.3 Access authorizations when creating programs

#### Application

When the data management server starts up, it reads data scheme SCHEMA.BIN followed by the plain-text identifier from the appropriate language dll.

During startup, 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.

#### Formats and data type

All entries are located in the [ACCESSMASKS] section.

#### Example:

The entries have 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" consists of the data types for

workpiece directories	"wks.dir"
workpieces	"*.wpd"
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 to 7	are permissible
7	Execution authorization is set	0 or 7	are permissible
7	Display authorization for all	0 to 7	are permissible
5	Delete authorization for Protection level 5 and higher	0 to 7	are permissible

## 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.

## 2.4.4 Configuring the acknowledgement icon for PLC alarms

### Application

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 specified with **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.

## 2.4.5 Configuring the "Language selection" softkey

### Application

If there are more than 2 language options on the control, the "Select language" softkey can be configured in the main "Startup" in order to change among the language options via the menu.

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

The values of "SkSelectLngInsteadSkChangeLng" have the following meanings:

- False:** "Change Language" softkey for changing between the foreground and background language, same as before (default setting).  
**True:** "Select language" softkey for selecting among the installed languages. This enables you to switch instantaneously among three or more language online.

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

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

---

### Note

With Select language the first language is not provided as a choice. By listing the first language in the language list again (and therefore also in the corresponding font settings) the first language also appears in the language selection menu.

---



## Parameterization

The names of the soft keys are predefined.

HSx	x 1 - 8 ,	Horizontal soft keys 1 to 8
VSy	y 1 - 8 ,	Vertical softkeys 1 to 8

Not all softkeys need to be assigned.

Reference: /IAM/ BE1, Section "Configuring OP Hotkeys and PLC Keys".

## 2.4.6 Creating user-specific operating menus

### Application

**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 of 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
- Startup and
- OEM applications

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 given in the table below).

A skeleton application can provide its own home 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 the horizontal and vertical softkey rows of the framework 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`.

For additional information refer to Section 2.5 Overview of the INI files.

## Examples

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 skeleton application.

Task0: Horizontal softkey 1

...

Task7: Horizontal softkey 8

**mntmmc** is 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.

**TASKCONF.INI**

```

[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 skeleton application
;   ; is started for the first time
; The softkey index that activates the desired task at program start is defined using the
;   ; start index entry.
; Services with StartIndex 5 for example

StartIndex = 5

; Main screen for the skeleton application
Picture= Skeleton1.bmp

; The first softkey is assigned the Program task (task index 2) as child
(HSK1IsTask=0)
; with the <root> parameter, the softkey texts specified for German (via default)
; and English (UK)

HSK1Task=2
HSK1IsTask=0

; Application-specific parameter: In this case, return to base level for the
; Program application, parameter: <root>
HSK1Command=<root>
HSK1SkText=Program ; No language specified is the default setting
; for all non-configured languages
HSK1SKText _UK=Program ; softkey label in English (_UK)

; The fifth softkey is assigned the Services task (task index 3) as child
(Hsk1IsTask=0)
; with the empty command parameter,
; the softkey texts specified for the default German (_GR) and English (_UK)

HSK5Task=3
HSK5IsTask =0
HSK5Command=
HSK5SkText=SERVICES
HSK5SkText _GR=Services
HSK5SkText _UK=service

; Access authorizations for the "Services" softkey

HSK5AccessLevel= 3

; The 2nd softkey in the ETC bar is assigned the Setup task (task index 5) as
child
; (HSK10IsTask=0) assigned with empty command parameter,
; the softkey texts specified 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, see Subsection 2.1.2

### 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 applications as child tasks

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 terminate
or
; The application is not terminated on return. This is the default setting
HSK<SoftkeyIndex>TerminateTask=0 ; 0 do not terminate
```

### Home 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 "Setup" 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: Keyswitch position 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 Operating area

y Command or state= i

Example:

HSK1Task=Program/<root>

has the same meaning as the full configuration:

HSK1Task=2

HSK1IsTask=0

HSK1Command=<root>

HSK1SkText=Program

x	y	Explanation
Machine	<root>	Go to the Machine home screen ( <b>task 0</b> ) with recall to the highest level for return. The softkey text is <i>Machine</i> .
Parameters	<root>	Go to the Parameters home screen ( <b>task 1</b> ) with recall to the highest level for return. The softkey text is <i>Parameter</i> .
Program	<root>	Go to the Program home screen ( <b>task 2</b> ) with recall to the highest level for return. The softkey text is <i>Program</i> .

x	y	Explanation
Service	<root>	Go to the Service home screen ( <b>task 3</b> ) with recall to the highest level for return. The softkey text is <i>Services</i> .
Diagnostics	<root>	Go to the Diagnostics home screen ( <b>task 4</b> ) with recall to the highest level for return. The softkey text is <i>Diagnosis</i> .
Diagnostics	State=10	Go to the Diagnostics home screen with recall to the highest level for return and selection of the alarm screen. The softkey text is <i>Alarms</i> . <b>Note:</b> The entire horizontal diagnostics softkey bar can be used.
Diagnostics	State=20	Go to the Diagnostics home screen with recall to the highest level for return and selection of the message screen. The softkey text is <i>Messages</i> .
Diagnostics	State=30	Go to the Diagnostics home screen with recall to the highest level for return and selection of the alarm log screen. The softkey text is <i>Alarm log</i> .
Diagnostics	State=40	Go to the Diagnostics service screen with recall to the highest level for return. The softkey text is <i>Service displays</i> .
Diagnostics	State=50	Go to the Diagnostics PLC status screen with recall to the highest level for return. The softkey text is <i>PLC status</i> .
Setup	<root>	Go to the <b>Setup</b> home screen ( <b>task 5</b> ) with recall to the highest level for return. The softkey text is <i>Commissioning</i> .
Setup	State=10	Go to the Machine data screen with recall on exiting the level. The softkey text is <i>Machine data</i> .
Setup	State=40	Go to the PLC overview screen in Setup with recall on exiting the level. The softkey text is <i>PLC</i> .
Setup	State=50	Go to the Drive overview screen in Setup with recall on exiting the level. The softkey text is <i>Drives/Servo</i> .
IBSetup	<root>	Go to the <b>Settings</b> home screen ( <b>task 34</b> ) with recall to the highest level for return. The softkey text is <i>System settings</i> .
Simulation	<root>	Go to <b>Simulation (task 27)</b> . A modal dialog is displayed, which you can use to select the program to be simulated. The simulation is aborted by pressing Cancel. The softkey text is <i>Simulation</i> .

### Other interfaces in the standard applications

The following standard applications can be addressed with these:

- Parameters
- Diagnostics
- Startup

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**(<SoftkeyList>)

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**

Softkeys in the ETC bar cannot be removed.

---

### **DisableVSK**(<SoftkeyList>)

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., DisableVSK(1,3,5-7).

### **DoVSK**=<SoftkeyIndex>

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

### **Recall**(<StatusList>)

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**=<StatusMatrix>

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 Configuring Package /

Please contact your local sales office for the latest electronic documentation.

## 2.4.7 Expanding the service overview user-specifically

### Application

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

If the file is present, the signals configured in it are evaluated and displayed in Diagnostics → "Service Overview" service displays. 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\_<language abbreviation>.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>]
.
.
```

### Meaning 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.



**Item:** (see description of the OPI)

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

**<CH>** for the channel number in which the axis is active

**<AX>** for the 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 axis is active.

**<PLC\_AX>** for a PLC axis DB; is replaced by 30+axis number

e.g.: Item = /Channel/MachineAxis/impulseEnable[u<CH>, <AX>].

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 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.

The numbers 0 to 99 are reserved for Siemens.

Internal bitmaps must be defined in the [BMP] section (see below). The predefined numbers have the following meaning:

0:		do not display bitmap
1:	OK	display bitmap 
2:	Not OK	display bitmap 
3:	Error => alarm is active	display bitmap 
4:	Not applicable	display bitmap 
5:	No drive allocated	display bitmap 
6:	Communication error	display bitmap 

### [BMP]

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

The name must be specified in double inverted commas. Without path names, the bitmap is searched for in mmc2 or addon, oem, user... Otherwise, it is searched for in the stated 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-Nr>= "<any text>".

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). It is stored in the mmc2(user,addon, oem ...)language directory

## 2.4.8 Workpieces with creating job lists and using templates

### Application

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.

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.

## Parameterization

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

```
[Program]
```

```
; 1=finished workpiece via workpiece template,  
; if opening is impossible with MCSP (preset)  
  
; 0=sends 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:  
**Non-existing programs are assigned to the following channels:**
- If no entry could be created in the DAT file:  
**No channel assignment found!**

Additional configuration options for multi-channel sequential programming are described in the Subsection 2.5.17 MMC.INI and 2.5.23 SEDITOR.INI.

## 2.4.9 Double-channel display

### Application

In order to be able to use the double-channel display in the Machine operating area, the following [MULTI\_CHAN\_i] sections are required:

```
[MULTI_CHAN_i]
CHAN_2 = k
"i" denotes the channel number in the current NCU.
The 2nd channel displayed is specified in the CHAN_2 entry CHAN_2 = k
```

---

#### Note

When the channel menu is selected, all entries under [MULTI\_CHAN\_i] are ignored.

---

This default 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.

**Example** for the default configuration of 2-channel display:

An NCU has 6 channels. (1, 2, 3, 6, 7, 8).

The channel combinations 1-6, 2-7, and 3-8 are always to be displayed together:

```
[MULTI_CHAN_1]
CHAN_2 = 6
```

```
[MULTI_CHAN_2]
CHAN_2 = 7
```

```
[MULTI_CHAN_3]
CHAN_2 = 8
```

```
[MULTI_CHAN_6]
CHAN_2 = 1
```

```
[MULTI_CHAN_7]
CHAN_2 = 2
```

```
[MULTI_CHAN_8]
CHAN_2 = 3
```

Note: Each of the paired channels appears once in [MULTI\_CHAN\_i] and once as the 2nd channel in CHAN\_2=i.

## 2.4.10 Tool management

### Application

The Tool Management HMI Advanced already contains the 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>0) 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 identifier
- curToolDuplo                Tool duplo number
- curToolType                 Tool type
- curEdgeNo                    Tool edge number, relative to tool and not to duplo number!
- curDLNo                      Tool edge correction location number, relative to edge
- curMagNo                     Magazine number; "0" if the current tool is not situated on a place or is intended for it.
- curPlaceNo                  Magazine place number; "0" if the current tool is not situated on a place or is intended for it.

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 identifier
- curCabToolDuplo          Tool duplo number
- curCabToolType            Tool type

Current tool in tool catalog:

- |                   |                   |
|-------------------|-------------------|
| - curCatToolIdent | Tool identifier   |
| - curCatToolDuplo | Tool duplo number |
| - curCatToolType  | Tool type         |

Additional information needed for setup can be found in:

**Reference:** /FBW/, Description of Functions Tool Management.

## 2.4.11 Tool selection without tool management

### Application

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 settings in the PARAM.INI file affect when these variables are written to: 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>, where MMCName comes 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 "\_\_xxx\_\_". 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] paragraph 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 edge,  
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
```

## 2.4.12 Setting technology-specific texts

### Application

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

- TECHNOLOGY
- TEXTFILES
- CONTEXT
- TEXTSEARCH

### TECHNOLOGY section

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

### TEXTFILES section

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)

(Also entered via "Settings")

---

#### Notice

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.



Table 2-6

Parameters for the text file	Interpretation
FileName.com	The language-dependent text file is read out
File name _XX.com (XX=language code, e.g. "gr" for German. For a list of language codes, see also /IAD/840D Installation and Start-Up Guide, Section "Alarm and Message Texts")	The language-specific 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

**Example:** ENDUSER=CUS.DIR\IFM\_.COM

First of all, the non-language-specific text file "Ifm.com" is opened in the path "C:\DH\CUS.DIR\" by default or, if the non-language-specific text file does not exist, the relevant language-specific text file (e.g. "IFM\_GR.COM") is opened.

Extract from Short Help txt file "IFM\_GR.COM":

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

## CONTEXT section

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".

(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.

## TEXTSEARCH section

The text search type can be specified in the [TEXTSEARCH] section. The following search methods can be parameterized:

- 1 = instruction texts only
- 2 = descriptive texts only
- 3 = instruction and descriptive texts

(Also entered under "Search").

## 2.4.13 Configuring the "EXIT" softkey

### Application

If you want the sequence control of the HMI software to support an "EXIT" softkey, you should make the following settings:

#### [Miscellaneous]

ExitButton=True  
ExitButtonIndex = *index*

The text for identifying the softkey text must be entered in  
F:\mmc2\language\re\_xx.ini where xx is the language identifier.

ExitButtonAccessLevel=7 ; Dialog box to ask user  
ExitButtonQueryUser = True

### Requesting a RESTART after shutdown (for Windows XP only)

This function enables HMI Advanced (including Windows XP) to be shut down followed by a reboot of the PCU during the setup phase. This enables you to avoid switching the PCU or machine on and off.

The following settings are required:

#### [Miscellaneous]

EnableRebootDialog=True

The entry can be set either directly in the INI file or by means of the user interface via Start-up → HMI → Settings → HMI Exit Mode. If the "EXIT" menu is configured accordingly, the "**Shutdown...**" or "**Restart ...**" prompt appears depending on the access stage.

- If the protection level is lower than the manufacturer level (protection level 2 to 7) or if the above entry is set to "False" in the REGIE.INI file, the prompt can be confirmed with:

<b>OK</b>	HMI is closed and Windows XP is shutdown.
<b>Cancel</b>	HMI is not closed

- If you have the applicable access authorizations (protection level 2 to 7), the following option is offered:

<b>Restart</b>	HMI and Windows XP are closed and then restarted.
<b>Exit</b>	HMI is closed and Windows XP is shutdown.
<b>Cancel</b>	HMI is not closed

## 2.4.14 Configuring the screen saver

### Application

The delay time before the screen saver is activated can be adjusted:

Section: GLOBAL  
Entry : MMCScreenOffTimeInMinutes

---

#### Notice

The HMI screen saver and Windows screen saver must not be used at the same time.

- HMI screen saver: protection for screen and backlight
  - Windows screen saver: protection for screen only (blank screen)
- 

### Displaying server names

You can choose to display the server name with:

Section: GLOBAL  
Entry : ServerVisible

## 2.4.15 Executing from the hard disk in M:N

Valid for: SINUMERIK powerline

### Application

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

- Configuration of channel menu for multiple channel systems
- Configuration of assignments of multiple HMI components and multiple NCU components (M : N configurations).
- If multiple HMI components have been assigned to one NCU, the HMI component that is to provide the part program in the event of "Execution from external source (hard disk)" must be specified.
- Reference to file with symbols for the PLC signals

In order to execute external programs that are located on the hard disk of the HMI in the data management system from the part program using the EXTCALL program command, the following entries, for example, are required in the "Connection part" section 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 for the NCUs from which the HMI
           can be accessed (see note)

; Description of significant net parameters
[param network]
bus=      opi

; HMI descriptions
[param HMI_3]
HMI_typ = 0x40
HMI_.....

; NCU components definition
[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.

---

Details can be found in

**Reference:** /FB/ B3 "Multiple Operator Panel Fronts and NCUs"

## 2.5 Overview of the INI files

---

### Note

This section 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!

---

### 2.5.1 AEDITOR.INI

Information about the ASCII editor can be found in the AEDITOR.INI file.

**[MEMORY]** section defines the size of the editable files.  
 MemFactor=f ; f stands for 8 MB units. Permissible values are  $1 \leq f \leq 31$   
 ; corresponding to 8 MB to 248 MB

Editor access authorizations can be modified in section

**[ACCESSLEVEL]**  
 GlobalReplace=6  
 SettingHiddenCheck=7  
 Enable\_Simulation = 7  
 ; Softkey simulation can be hidden  
 ; in the editor

**[MMC\_Wizard]**  
 AEDITOR=AEDITOR.COM ; see /BE1/

**[MEMORY]**  
 ; default factor for maximum size of file  
 ; valid values are: from 1 to 31  
 ; factor x ( $1 \leq x \leq 31$ ) 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 higher 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 high 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 been
; actuated, response depends on application
; UndoHandling=2 No Undo/Redo function
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
```

## 2.5.2 COMIC.NSK

```
REM ----- TYPICAL COMIC START
REM CHANNEL 1

TOPIC(machineswitch) COMIC_START("COMIC001MachineSwitch
"/Channel/Configuration/mmcCmd[u1],/Channel/Configuration/mmcCmdQuit[u1])

REM CHANNEL 2

TOPIC(machineswitch) COMIC_START("COMIC002MachineSwitch
"/Channel/Configuration/mmcCmd[u2],/Channel/Configuration/mmcCmdQuit[u2])
```

## 2.5.3 DINO.INI

```
[ACCESSLEVEL]
USER=3 ; Read authorization for series setup
; Customer password (default)
PROGRAMMER=4 ; Setup authorization for series setup
; corresponds to keyswitch 3 (default)

[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
    ;   an IBN 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 any
    ;     existing files with same name, if no such file exist destination
    ;     is MMC
    ; else: destination is determined from the file header, i.e. from the
    ;     location where the file was read from
TakeDestinationFromHeader=0

```

## 2.5.4 DG.INI

Softkey access in the Diagnostics operating area can be set individually by entering the protection level. Enter the required protection level in the DG.INI file in section:

### [ACCESSLEVEL]

```

ALARMS=7           ; Alarms

SERVICE=7         ; Service displays

COMMUNIC-LOG=3     ; Communication log
PLC=3              ; PLC status
PLC-CHANGE=2       ; PLC status:
                   ; Softkey change value, softkey reset value
                   ; Softkey commit value
                   ; Locked by manufacturer

REMOTE-DIAGNOSIS=4 ; Remote diagnosis
FILE=7             ; File functions
RUN-LOG=7          ; Softkey action log, locked by user

```

## 2.5.5 DGOVW.INI

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>]
.
.
```

For details on configuring and for an example, see Subsection 2.4.7 Expanding the service overview user-specifically.

## 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 ...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"
```



## 2.5.6 DH.INI

### Example:

```
[ACCESSMASKS]
\wks.dir\*.wpd\*.mpf = 75775
\mpf.dir\*.mpf = 75775
\cus.dir\*.spf = 33773
```

## 2.5.7 DPDH.INI

```
[ACCESSLEVEL]
PROGRAMMER=4
OPERATOR=5
ENABLE_DIALPROG=0
SetStandardWpSave=2
ENABLE_SIMULATION=7
```

## 2.5.8 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 extern model, if program
; size >= EXTERN_SIZE
; only if EXTERN_MODAL<>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 DPNC SIM.EXE
      ;=0: TO_INI.INI and TO_ADDON.INI are loaded by simulation
TOOLS=0
      ;= 1: all cycles are preloaded by DPNC SIM.EXE
      ; if section MAIN, entry CYCLE is set, additionally all cycles from shadow
      directory are preloaded by DPNC SIM.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 in \DP.DIR\SIM.DIR
CYCLEINTERFACE=2
```

## 2.5.9 HEADER.INI

### [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 SBL mode
7=CondStop.bmp           ;NC Stop active
8=CondStop.bmp           ;Read-in enable missing
9=CondStop.bmp           ;Feed enable missing
10=CondWait.bmp          ;Dwell 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 step change
26=CondWait.bmp          ;Waiting for position control
27=CondWait.bmp          ;Waiting for thread cut
28=CondWait.bmp          ;Reserved
29=CondWait.bmp          ;Waiting for punching
```

30=CondWait.bmp	;Waiting for safe operation
31=CondWait.bmp	;No channel ready
32=CondStop.bmp	;Oscillation active
33=CondWait.bmp	;Axis 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 disable
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 started
47=CondWait.bmp	;AXCT: waiting for spindle stop
	;Axis container exchange
48=CondWait.bmp	;Machine data matching (;New config)
49=CondWait.bmp	;Axis replacement: axis in coupling
50=CondWait.bmp	;Axis replacement: Liffast active
51=CondWait.bmp	;Axis replacement: New-Config active
52=CondWait.bmp	;Axis replacement: AXCTSW active
53=CondWait.bmp	;Axis replacement: Waitp active
54=CondWait.bmp	;Axis in another channel
55=CondWait.bmp	;Axis replacement: axis PLC axis
56=CondWait.bmp	;Axis replacement: axis reciprocating axis
57=CondWait.bmp	;Axis replacement: axis jog axis
58=CondWait.bmp	;Axis replacement: command axis
59=CondWait.bmp	;Axis replacement: axis OEM axis
60=CondWait.bmp	;Coupled following axis
61=CondWait.bmp	;Coupled 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 limit stop
66=CondWait.bmp	;"Master-Slave switchover active"
67=CondWait.bmp	;"Axis replacement: 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 of channel 1 is displayed,
Spindles = 1,2 ;superimposed by feed disable for channel 1 and
;spindle disable for spindles 1 and 2

[Pos3]
SpindStat = 1 ;Spindle status for spindle 1
;ChanStatFeedStop = 3 ;Channel status of channel 3 is displayed,
;superimposed with feed disable for channel 3

[Pos5]
ChanStat = 2 ;Channel status for channel 2 is displayed
;SpindStat = 3 ;Spindle status for spindle 3 is displayed

[Pos9]
ChanStatFeedStopSpindleStop = 1 ;Channel status of channel 1 is displayed,
Spindles=1 ;superimposed by feed disable for channel 1 and
;spindle disable of Spindle 1

[Pos11]
ChanStatFeedStop = 2 ;Channel status of channel 2 is displayed,
;superimposed with feed disable for channel 2

[Pos13]
SpindStat = 2 ;Spindle status for spindle 2

[Pos15]
ChanStatFeedStopSpindleStop = 1 ;Channel status of channel 1 is displayed,
Spindles=1 ;superimposed by feed disable for channel 1 and
;spindle disable of Spindle 1

[UserIcons]
;UI_0 = <name.bmp> , <Position> ;Name of icon, position
...
;UI_31 = <name.bmp> , <Position> ;Name of icon, position

;USER_ICON_BASE = DBxx.DBByy
; Number of data block with the
; double word for user icons
; yy start byte of 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 limit if
; user texts are used

[OemIcons]
;Oem_ICON_BASE=DBxx.DBByy

;OI_0= <name.bmp> , <Position> ;Name of icon, position
...
;OI_31 = <name.bmp> , <Position> ;Name of icon, position
[OEMTextInfo]

```

```

;OEM_NUMBER_TEXTFIELD=3 ;A maximum of 3 text fields per line, i.e.,
                        ;a total of 6 text fields in two lines is possible.
                        ;Output length limited to a maximum of 38
                        ;characters per line.

;OEM_TEXT_BASE=DBxx.DBByy

```

### 2.5.10 HMIDESK.INI

Settings are only necessary in this file if HMI Advanced is installed on a programming device or standard PC.

Settings:

#### [Global]

```

;Global configuration data
;Definition of desktops (minimum 2, maximum 4)

```

**NumberOfDesktops = 2**

```

;Control mode for mouse operation: Taskbar

```

**ControlMode = Icon**

**KeyboardControl = False**

```

[Desktop1]

```

#### [Desktop2]

```

;Active desktop with taskbar

```

**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].

```

## 2.5.11 IB.INI

Softkey access in the Commissioning operating area can be set individually by entering the protection level. Enter the desired protection level under the [AccessLevel] section. The operating languages are defined in the [Language] section.

### [ACCESSLEVEL]

RESET =2	;NCK reset	NCK reset
CHG_SCL_SYS=5	;change scaling system	Dimension system inch/metric
MASCHDAT =4	;Machine data	Machine data
DRIVE_CFG=2	;drive configuration	Drive configuration
BOOTFILE=2	;Boot file	Boot file
MOT_CTRL=2	;Motor/Valve selection	Motor/valve selection
VIEW=2	;User views + Display options	User views and display options
NC_ADDR=2	;NCK address	NCK address
SETUP_SWITCH=2	;start-up switch	IBN switch
PLC_STATE=3	;plc state	PLC status
PLC_DATE=3	;plc date	PLC date
PLC-CHANGE=2	;PLC status:	
	;Sk-Change Value	(Change)
	;Sk-Reset Value	(Undo)
	;Sk-Commit Value	(Accept change)
		Locked by manufacturer
DRIVE_SERVO=2	;Drives/Servo	Drives/servo
SET_COL=3	;colours Colors	
SYST_SETTINGS=4	;system settings	System settings
SET_LANG=2	;Language	Language
SET_BTSS=2	;Operator panel	Operator panel front
OPTION=2	;Options	Options
EDIT=1	;Editor	Editor
DOS=1	;DOS shell	DOS Shell
TOOL_MNG=3	tool management	Tool management
FILE=7	;File functions	File functions
PASSWD=3	;Set/Delete password	Change and delete password
CHG_PASSWD=7	;SK "password ..."	SK "password ..."
CHANGE_LANG=7	;change Language	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.

---

## 2.5.12 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   ISO turning or
; SiemensI= DH\CST.DIR\IFIM_GR.COM   ISO milling
; Machine manufacturer = DH\CST.DIR\IFM_GR.COM
; Machine manufacturer = 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_Sensitivity=1
```

### [TEXTSEARCH]

; = 1 Text search for instruction texts only  
; = 2 Text search for descriptive texts only  
; = 3 Text search for instruction and descriptive texts  
; Notation in the section e.g.:

Search type=3

## 2.5.13 KEYS.INI

Entries in this file allow hard keys on the operator panel fronts OP 010 and OP 010C or SINUMERIK keyboards with hotkey block and PLC softkeys to be used.

See: /BE1/ Configuring OP Hotkeys and PLC Keys

## 2.5.14 LOGDRIVE.INI

With the "logdrive.ini" file, the logical drives are connected and the associated access authorizations are 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%local
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 via the operator interface of HMI Advanced under "Commissioning" → "HMI" → "logical drives".

---

## 2.5.15 MACHINE.INI

The [AccessLevel] section in the "maschine.ini" file can be used to make the access authorization function settings. The file includes detailed comments.

**[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), used to sets up tools  
; ; and test and run 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)  
  
; on delivery, all users are authorized for (almost) all work
```

JOG\_PRESET=7 ; Preset or Set actual value at JOG mode  
                   ==>5 Suggestion machine setter  
 JOG\_HAND=7 ; Handwheel in JOG mode  
                   ==>6 Suggestion operator  
 JOG\_ANKRATZ=7 ; Scratch  
                   ==>6 Suggestion operator  
 JOG\_INCREM=7 ; Set increment in JOG mode  
                   ==>6 Suggestion operator  
 JOG\_SYNAKT=7 ; Synchronous actions in JOG mode  
                   ==>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 in JOG mode  
                   ==>7 Suggestion everyone  
 JOG\_INCHMET=7 ; Switch inch/metric in JOG mode  
                   ==>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 ; Handwheel in MDI mode  
                   ==>6 Suggestion operator  
 MDI\_EDITOR=7 ; Edit program in MDI mode  
                   ==>5 Suggestion machine setter  
 MDI\_ED\_DAT=7 ; Editor functions in MDI mode  
                   ==>5 Suggestion machine setter  
 MDI\_ED\_CLR=7 ; Clear buffer in MDI mode  
                   ==>5 Suggestion machine setter  
 MDI\_V\_ISTW=7 ; watch windows of Actual-values,  
                   ==>7 Suggestion everyone  
                   ; Spindles, G functions, Aux. functions,  
                   ; Feedrate, Tools or Axis feedrate in MDI mode  
 MDI\_V\_ZOOM=7 ; Zoom actual values in MDI mode  
                   ==>7 Suggestion everyone  
 MDI\_SYNAKT=7 ; Synchronous actions in MDI mode  
                   ==>5 Suggestion machine setter  
 MDI\_DATEI=4 ; File functions in MDI mode  
 MDI\_INCHMET=7 ; Switch inch/metric in MDI mode  
                   ==>6 Suggestion operator  
  
 AUTO\_OVERST=7 ; Overstore in AUTO mode  
                   ==>5 Suggestion machine setter  
 AUTO\_DRF=7 ; DRF in AUTO mode  
                   ==>6 Suggestion operator  
 AUTO\_PROGBE=7 ; Program control in AUTO mode  
                   ==>6 Suggestion operator  
 AUTO\_SASUCH=7 ; Blocksearch in AUTO mode  
                   ==>6 Suggestion operator  
 AUTO\_HAND=7 ; Handwheel in AUTO mode  
                   ==>6 Suggestion operator

AUTO\_KORREK=7 ; Correct program in AUTO mode  
=>5 Suggestion machine setter  
AUTO\_V\_ISTW=7 ; watch windows of Actual-values,  
=>7 Suggestion everyone  
; Spindles, G functions, Aux. functions,  
; Feedrate, Tools or Axis feedrate 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 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

```

**2.5.16 MBDDE.INI****[Alarms]**

```

; determines the order in which the alarms are inserted in the alarm list
; possible values:
; FIRST: puts newer alarms at the 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 Quits
; 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 diagnostics but causes multiple entries in
; the log list of one alarm event (NEW, NULL, QUIT).
; Using a newer PLC firmware (PLC operating system version: 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: Deactivating the workaround while using 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 chars, 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_  
ZYG=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=  
UserZYG=  
UserCZYK=  
UserSTANDARD_CYCLES=  
UserSHOPMILL_MANUALTURN_CYCLES=  
UserMEASURE_CYCLES=  
UserUSER_CYCLES=  
UserPLC_PMC=
```

```
; NCU-specific texts in M:N environments  
; the entry names are formed 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 annotated examples below correspond to the supplied netnames.bsp file
```

#### [net TextFiles] (for SINUMERIK powerline only)

```
;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  
PLCMMSG=100  
MMC=100

; This section defines different characteristics of the log that  
; specifies how particular error messages are to be registered.

#### [PROTOCOL]

For details on formation of filter expressions for control of message volume,  
refer to Subsection 2.4.2 Setting 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 overwrite old entries  
; (principle: ring buffer)

Records=150

; Mode of alarm log writing  
; if a number less than -1 is specified, this number of log files is  
; kept in parallel, and one of them is actually used  
; if -1 is specified, the log file is not automatically updated  
; if 0 is specified, the log file is immediately updated  
; if a number greater than 0 is specified, the log file is updated after the  
; specified number of seconds elapses 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 multiple file strategy can also be selected by means of the DiskCare entry in the Ini file.

; information about 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 filename extension may also be added (optional).  
; For additional information, see SINUMERIK 840D OEM package  
; HMI User's Manual.

**[Helpcontext]**

File0=help\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=



## 2.5.17 MMC.INI

```

[GLOBAL]
; connectiontype 0 = default: Connection of 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, make 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 for
; parameterization, e.g., for a trace (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] SINUMERIK powerline only**  
ADDRESS0=13,NAME=/NC,SAP=d  
ADDRESS1=13,NAME=/PLC,SAP=1  
ADDRESS4=/SELF,10000103

; Example:  
; WINCOS-->CP-->TCPIP-->CP-->LINCOS-->NC/PLC/DRIVE0-5  
; 1.) change TCPIP address below to your target TCPIP address  
; 2.) do not forget to start your windows CP (e.g., cp\_840di.exe)  
; 3.) Note: no disconnection detection in some cases

**:[NCU840D] only SINUMERIK solution line**  
;ADDRESS0=192.168.214.1,LINE=10,NAME=/NC,SAP=040d,PROFILE=CLT1\_\_CP\_L4\_INT  
;ADDRESS1=192.168.214.1,LINE=10,NAME=/PLC,SAP=0201,PROFILE=CLT1\_\_CP\_L4\_INT  
;ADDRESS2=192.168.214.1,LINE=10,NAME=/CP,SAP=0501,PROFILE=CLT1\_\_CP\_L4\_INT  
;ADDRESS10=192.168.214.1,LINE=10,NAME=/DRIVE0,SAP=0201,SUBNET=004B-00000008:3,  
PROFILE=CLT1\_\_CP\_L4\_INT  
;ADDRESS11=192.168.214.1,LINE=10,NAME=/DRIVE1,SAP=0a00,PROFILE=CLT1\_\_CP\_L4\_INT  
;ADDRESS12=192.168.214.1,LINE=10,NAME=/DRIVE2,SAP=0b00,PROFILE=CLT1\_\_CP\_L4\_INT  
;ADDRESS13=192.168.214.1,LINE=10,NAME=/DRIVE3,SAP=0c00,PROFILE=CLT1\_\_CP\_L4\_INT  
;ADDRESS14=192.168.214.1,LINE=10,NAME=/DRIVE4,SAP=0d00,PROFILE=CLT1\_\_CP\_L4\_INT  
;ADDRESS15=192.168.214.1,LINE=10,NAME=/DRIVE5,SAP=0e00,PROFILE=CLT1\_\_CP\_L4\_INT

The language selection can also be made via the HMI user interface in the operating area "Commissioning" → "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  
LBLEList=German, English, Spanish, French, Italian, Chinese

**[CONTROL] Relevant for HMI-Advanced on PC/programming device only**  
; Resolution: 0=fixed (640x480 default), 1=variable  
Resolution=0  
; BaseScreen specifies the resolution, for which the applications  
; were developed originally (typically: 640x480)  
; BaseScreen - default: 640x480  
BaseScreen=640x480

If a mouse is connected to the controller, the mouse pointer is positioned in the upper right corner of the screen by default. The entry below can be used to switch disable this behavior:

```
;MoveCursor=0: do not move cursor to upper right corner
;MoveCursor=0
```

**[Services]**            **Relevant for HMI-Advanced on PC/programming device only**  
 FloppyDisk= a:  
 PCCard=

---

### Notice

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, which can also be set on a technology-specific basis via the HMI user interface in the “Commissioning” operating area → “Machine Data”:

```
[OPI Settings]
$MM_LCD_CONTRAST=7
$MM_DISPLAY_TYPE = 1
$MM_DISPLAY_MODE = 2
$MM_FIRST_LANGUAGE = 1
$MM_DISPLAY_RESOLUTION=3
$MM_SPIND_DISPLAY_RESOLUTION=3
...
$MM_MD_DRV_TEA_FILTER = 0
$MM_MD_DRV_TEA_IDX_LIMIT = 0
$MM_MD_TEXT_SWITCH=0
$MM_ACTION_LOG_MODE = 0xFE
$MM_PROGRAM_SETTINGS = 0
$MM_SW_OPTIONS = 0
$MM_HMI_TESTAUTOMAT_OPTION = 0
$MM_HMI_WIZARD_OPTION = 0
$MM_TRACE=0
```

The action log can also be set via the HMI user interface in the operating area “Commissioning” → “HMI” → “System Settings”:

```
; ActionLog 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
; 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 a maximum of 10 NC/PLC variables to be monitored.
        ; 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 alarm numbers separated by commas.
        ; Changes in 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: Subsection 2.4.8 Workpieces with creating job lists and using templates.

## 2.5.18 NETNAMES.INI

### M:N assignments (SINUMERIK powerline only)

In the NETNAMES.INI file, you can determine which operating components can operate which NCUs. The following details are specified:

- Names of the operating components
- Names of the NCUs
- Communication connections between the NCUs
- Clear assignments for startup

Details can be found in

**Reference:** /FB/, B3 Multiple Operator Panel Fronts and NCUs

### Procedure

The following sequence is useful when configuring the NETNAMES.INI file for M:N communication:

1. Copy the existing NETNAMES.BSP file created during installation, and modify it for the application.
2. Rename the existing NETNAMES.INI file.
3. Rename the modified NETNAMES.BSP file to NETNAMES.INI.
4. Save in C:\USER\NETNAMES.INI.
5. Define groups (maximum of 24) in "logChanSetList"
6. Define NCs and channels with unique names for the groups in "logChanList".
7. For the channels, specify the NC name in "logNCName"
  - Specify the channel number on the NC in "ChanNum"
  - Specify Reserve softkey yes/no (via the command IsAGap" = True/False)
8. For the HMI, specify the default machine "DEFAULT\_logChanGrp" and the default channel in "DEFAULT\_logChan" for startup.

---

**Note**

The NC writes the channel number of the HMI in DB19, DBB22.

The example below is simplified for 1:1 assignment of HMI : NCU  
such that conn\_2 is omitted and all channels are located on NCU\_1.

---

**Example:**

```
NETNAMES.INI
[own]
owner = MMC_1
; Description of possible connections
[conn MMC_1]
conn_1 = NCU_1
conn_2 = NCU_2
; Description of significant net parameters
[param network]
bus = opi ; the MPI interface must be used on the SINUMERIK 810D

[param MMC_1]
mmc_address = 1
[param NCU_1]
nck_address = 12
plc_address = 12
; core name for the PLC symbol files for representing signals with symbols
PlcSymbolFile= ...

[param NCU_2]
nck_address = 14
plc_address = 14
; core name for the PLC symbol files for representing signals with symbols
PlcSymbolFile= ...
[chan MMC_1]

DEFAULT_logChanGrp = Milling1
DEFAULT_logChan = Channel13
ShowChanMenu = TRUE
logChanSetList = Milling1, Milling2
[Milling1]
logChanList = Channel11, Channel12, Channel13, Channel14, Channel15
[Channel11] ; logical channel
logNCName = NCU_1 ; logical NCU name see above
ChanNum = 1 ; channel number within NCU
[Channel12]
logNCName = NCU_1
ChanNum = 2
[Channel13]
logNCName = NCU_1
ChanNum = 3
[Channel14]
logNCName = NCU_1
ChanNum = 4
[Channel15]
```

```

logNCName = NCU_1
ChanNum = 5
[Milling2]
logChanList = Channel21, Channel22, Channel23, Channel24, Channel25
[Channel21]
logNCName = NCU_2
ChanNum = 1
[Channel22]
logNCName = NCU_2
ChanNum = 2
[Channel23]
logNCName = NCU_2
ChanNum = 3
[Channel24]
logNCName = NCU_2
ChanNum = 4
[Channel25]
logNCName = NCU_2
ChanNum = 5

```

## 2.5.19 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_SYSTEMMENU, WS_MINIMIZEBOX and WS_MAXIMIZEBOX off

WindowStyle_Off = 720896

; other possible task entries aren't used

WindowStyle_On =

x =
y =

Width =
Height =

```

## 2.5.20 PARAM.INI

```
[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 parameters
```

```
[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
```

The tool compensation (turning) is calculated in **Diameter** when the selected axis is equal to the traverse axis and the entry CalDiaVal=1 is set in the **[Bezugma]** section in the Param.ini file.

## 2.5.21 PARAMTM.INI

The full range of parameter settings are presented in the tool management description.

**Reference:** /FBW/ Description of Functions Tool Management



## 2.5.22 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 **temperature monitoring for PCU50**

; TemperatureMonitoring = True

TemperatureMonitoring = False ; Preset from

**Reference:** OEM Package HMI, Chapter 6

### Hibernate (nur Windows XP)

The **HibernateMode** parameter in the [Miscellaneous] section is used for setting hibernation options.

These settings can also be made via the user interface of HMI Advanced. For this, select "Commissioning" → "HMI" → "System Settings" from the operating area.

HibernateMode = ...

The following values can be set:

HibernateMode = Off	Default: off Hibernate saves and Hibernate reboots are not performed.
HibernateMode = Normal	HMI Advance first closes all applications. After the shutdown, the controller is restarted automatically.

```
        ; 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    Reserved for expansions by Siemens  
12 to 24  Available for OEM applications  
25 to 32  Reserved for simulation  
33 to 47  Reserved for Siemens applications (in the ADD_ON directory)
```

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 a task or the EXIT button is assigned to Task 16 or higher in the bar.

```
; 4. bar : allocated 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 a task or the EXIT button is assigned to Task 96 or higher in the bar.

The Task 50 to Task 87 entries can be used by OEMs.

### **[CommandToTask]**

```
        ; only necessary for HMI startup 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.

Further information: see "Activating Dialog Screen from NC Program"

## 2.5.23 SEDITOR.INI

### Application

You can modify the displays in the **multi-channel sequential programming (option)** by means of settings or by making your own additions in this file.

The sections that contain the designation [ \*User\* ] enable the cycles of the machine manufacturer and the user to be identified and displayed. 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 so that, unlike in the standard version of the ini file, only changes and additions have to be stored in the Add-on, oem, and user directories.

See also: Subsection 2.5.17 MMC.INI, [PROGRAM] Section.

### 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**

Shortcut CTRL+I can be used during Editing mode of a step to insert a SEFORM instruction. The text to be inserted can be configured here.

### **[Cache]**

#### **Enabled**

A file can be opened faster by using the cache.

Presently, only the following setting is available:

0: Cache is not 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(...)`. These icons must be located in the directory that was specified in `IconPath`.

#### **Block**

Icon for main sequence `SEFORM(..., 0, ...)`

**SubBlock**

Icon for subsequence SEFORM(..., 1, ...)

**Sync**

Icon for synchronization marks, e.g., WAITM(...)

**Cycle**

Icon for cycle

**GeoProz**

Icon for contour definition

**[SyncMarks\_User]**

Machine manufacturers can enter their expansions for synchronization marks here.

```

;Definition of synchronization marks (search texts)
;sync1 = WAITM>(*n,*c,*c,*c,*c,*c,*c,*c,*c,*c,*c)
;      '#' : separator: text before is searched in one
;           channel, text behind in all others.
;           It isn't possible to synchronize this kind of
;           synchronization 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:

- \* Wildcard; text replaced by '\*' must be the same in all channels.
- n Identification by means of a number, e.g., WAITM(\*n,1,2,3)  
Wait marks can be assigned using this number.
- \*c Channel number, these numbers does not have to be in the same order.  
The program coordination channel does not have to be specified  
(e.g.: WAITM(1,\*c,\*c,\*c).
- # Channel separators: Text before separator is searched for in one channel,  
Text after separator is searched in all other channels. These  
synchronization marks cannot be synchronized with each other.

### **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 affects Channel 1, 2, and 6 only. The entry must be made as follows:

```
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:

**CycleX** = L100(\*  
CycleX = L100(\* „\$85001“

CycleX = L100(\* „\$85001/fixed text“

The following is displayed, e.g.

„L100(2,3,4)“  
Language-dependent text that is  
stored in 85001

Language-dependent text that is  
stored in 85001 + “/fixed text”

**CycleX\_Type**

Definition of whether cycle is to be displayed as a subblock or a main block.

- 1: Subblock
- 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_lco=cm_poly_dr_reaming.bmp  
NCG_4_Level=0  
...
```

[CYC86]

```
;boring  
NCG_1_Label=NC1  
NCG_1_Text=$80666  
NCG_1_lco=cm_poly_dr_bore.bmp  
NCG_1_Level=0
```

```
NCG_2_Label=NC2  
NCG_2_Text=MCALL CYCLE86  
NCG_2_lco=cm_poly_dr_bore.bmp  
NCG_2_Level=0
```

[CYC83]

```
;deep hole drilling  
NCG_1_Label=NC1  
NCG_1_Text=$80794  
NCG_1_lco=cm_poly_dr_deepdrill.bmp  
NCG_1_Level=0
```

```
NCG_2_Label=NC2  
NCG_2_Text=MCALL CYCLE83  
NCG_2_lco=cm_poly_dr_deepdrill.bmp  
NCG_2_Level=0
```

## Explanations:

Two graphics-supported supported cycles are shown here by way of example. 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 found 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 furnished in the INI file for this entry [CYC82].

#### **Section for cycles (e.g.: [CYC82])**

##### **No.**

Number of cycle versions.

##### **NCG\_1\_Label**

"Name" of version. Generally, this is the text that is found in the NC program after the path of the COM file (NC1, NC2, etc.).

##### **NCG\_1\_Text**

Text that is displayed. A fixed text, a language-specific text (\$85111), or a combination of both (\$85111/my additional text) can be specified.

##### **NCG\_1\_Ico**

Associated icon, it must be located in the directory that was specified in IconPath.

##### **NCG\_1\_Level**

Definition of whether cycle is to be displayed as a subblock or a main block.

1: Subblock

0: Main block

---

##### **Note**

See also: /IAM/ BE1 Expand User Interface

---

## **2.5.24 SIMTOGEO.INI**

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 have not effect.

---

##### **[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, insert 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, insert 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, insert 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, insert 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, insert 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, insert length
;holder_angle=         ; TC_DP10, holder angle
;reference_direction=   ; TC_DP11, cutting direction
;clearance_angle=      ; TC_DP24, clearance angle

```

## 2.5.25 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. Instead of APPLICATION, use
; the section name configured in regie.ini. For every section name assigned in regie.ini
; with Task ; = name:= mntmmc cmdLine:= "SectionName", ...
; a standalone section must be configured here.

```

```
[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=

; accesslevel 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=
```

## 2.6 Simulation data matching

This function can be used to simulate machining on 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

### 2.6.1 Simulation data

#### Files used

When HMI Advanced is started up, the following files are initialized for the simulation in data management directory \DP.DIR\SIM.DIR.

UMAC.DEF	User macros
SMAC.DEF	System macros
SMAC_SC.DEF	"
MMAC.DEF	Machine manufacturer macros
UGUD.DEF	Global user data
SGUD.DEF	Global system data
MGUD.DEF	Global machine manufacturer data
GUD4.DEF	NC user data 4
GUD5.DEF	NC user data 5
GUD6.DEF	NC user data 6
GUD7.DEF	NC user data 7
GUD7_SC.DEF	"
GUD8.DEF	NC user data 8
GUD9.DEF	NC user data 9
CH(c)_RPA.INI	R parameters (channel-specific)
CH(c)_GUD.INI, NC_GUD.INI	User data (channel-specific, global)
CH(c)_SEA.INI, NC_SEA.INI	Setting data (channel-specific, global)
AX_SEA.INI	Setting data for axes
CH(c)_UFR.INI, NC_UFR.INI	Work offsets (channel-specific, global)
CH(c)_PRO.INI	Protection zones (channel-specific)
NC_CEC.INI	Sag/angularity
TO_INI.INI	Tool data (complete)
INITIAL.INI	NCK initialization data
DPCUSCYC.COM	Description of the user cycle interfaces

## Startup

The above files, if present, are read in when the simulation application starts up. From the initialization data in the INITIAL.INI file, only the machine data are interpreted by the simulation application during startup.

The files (CH(c)\_xxx.INI, NC\_xxx.INI and AX\_SEA.INI) are read in according to the order of their date stamp, starting with the oldest file.

DPCUSCYC.COM:

The file DPCUSCYC.COM (mapping of user cycle interfaces) is only read if in the DPSIM.INI file in the [PRELOAD] section

CYCLES=0     or  
CYCLEINTERFACE=1 is set.

The following cases are possible:

1. CYCLES=0  
The corresponding cycles are not loaded into the simulation until they are called by the simulation interpreter for the first time.
2. CYCLES=1  
All cycles of the NCK and HMI Advanced are preloaded when the simulation application is started (eliminating cycle reload times during simulation).
3. If a cycle of the same name exists on HMI Advanced and NCK, the NCK cycle is loaded. If a cycle of the same name exists in \CUS.DIR, \CMA.DIR and \CST.DIR, it is found and loaded first in \CUS.DIR, then in \CMA.DIR and finally in \CST.DIR.
4. CYCLES=1 and CYCLEINTERFACE=0  
The cycle interfaces are formed directly by simulation. This means that all cycles (in \CUS.DIR, \CMA.DIR, and \CST.DIR of HMI Advanced and the NCK) have to be loaded in advance (occupying program memory in the simulation NC).

## NC cycles

The NC cycles (CUS.DIR, CMA.DIR and CST.DIR) are copied as an image into the corresponding directories in data management. The location is determined in the DPSIM.INI file by the CYCLE entry (\DP.DIR\SIM.DIR) in the [MAIN] section. Synchronization takes place when the HMI is started at every cycle synchronization. This enables optimized access to the cycles and automatic creation of images of the cycle interfaces.

## Simulation with OEM transformations

For OEM-specific adjustments, e.g. for Compile cycles, the standard configuration, which is normally located in <InstallationPath>\dp\dpsim.tea, can be overwritten. To do this, you must create a dpsim.tea file containing the desired adjustments (e.g. different values for TrafoType) in the <InstallationPath>\OEM directory. The settings in this dpsim.tea are evaluated last and take priority over the default settings in <InstallationPath>\dp\dpsim.tea.

## Displaying the axis names/axis values

The display of axis names and axis values in simulation can be configured in the DPSIM.INI file and applies to all channels.

By default the first 3 geometrical axes are displayed. In addition, any other channel axes can also be displayed. Assignment takes place in the [LINKS] section

```
[LINKS]
ACT_POS_DISPL_1=1
ACT_POS_DISPL_2=2
ACT_POS_DISPL_3=3
ACT_POS_DISPL_4=0
ACT_POS_DISPL_5=0
Value = 0      The associated space in the display is freed
Value = 1      The associated default geometrical axis is displayed
Value = 1 to n  Geometrical axis is displayed
                 (n = 1 to number of generated axes)
```

The indices of the geometrical/channel axes are permitted as values (1 to number of axes), axes that are not present are ignored. A maximum of 5 axis names and values can be displayed simultaneously.

## Formatting

The number of digits (3 to 15) can be specified for formatting the axis values.

```
[LINKS]
ACT_POS_DIGITS=10
```

The number of digits after the decimal point (maximum number of digits) is read from the simulation initialization data via MD 9004 DISPLAY\_RESOLUTION or MD 9011 DISPLAY\_RESOLUTION\_INCH.

## Hiding axes

The geometry/channel axes can be hidden using MD 20098 DISPLAY\_AXIS. Bit 0 (display in the actual-value windows) is evaluated for the simulation.

At a maximum, as many axes can be displayed with their values that are not hidden with ACT\_POS\_DISPL\_<x> with 0, i.e. at a maximum, the first 5 unhidden geometry/channel axes in their sequence.

---

### Note

The more space that is required to display the axes, the less space is available to display the actual block. If necessary, the axis display can be disabled off from the Settings/Options On/Off menu in the "Actual position" field (within the Display options field).

---

## Search strategy

If a program (with no path) is required by the simulation, the following directories are searched:

1. File in current NCK workpiece directory
2. File in current workpiece directory of HMI Advanced
3. File in NCK subroutine/main program directory
4. File in HMI Advanced subroutine/main program directory
5. File in NCK user cycle directory
6. File in HMI user cycle directory
7. File in NCK manufacturer directory
8. File in HMI manufacturer directory
9. File in NCK standard cycle directory
10. File in HMI standard cycle directory

If a program is not present in the simulation, it is loaded from the NC or from the HMI.

All programs (apart from cycles) that are loaded into the simulation are deleted from the simulation again in the event of a workpiece change. If one of these programs is needed again by the simulation, it is reloaded.

## Program selection

If a part program for a workpiece is selected for the simulation, the following actions are executed:

1. All tools are cleared with the file TO\_CLEAR.INI (in HMI\_ADV\DP\SIM) (if present).
2. If WPD\_INI\_MODE = 1 (see item 5) and if a TOA file with the selected program name exists in the workpiece directory, this file is loaded into the active file system for the simulation (otherwise file TO\_INI.INI if it exists in the data management system in \DP.DIR\SIM.DIR).
3. If tool management is active, the file TO\_ADDON.INI (in HMI\_ADV\DP\SIM) is also loaded into the simulation and executed.
4. Workpieces: an initialization file with the workpiece name is loaded into the simulation
5. Workpieces:
  - MD 11280 WPD\_INI\_MODE = 1  
All files (RPA, GUD, SEA, UFR, PRO, CEC, TMA) with the part program names are loaded into the passive file system in the simulation. Program-specific TOA files are loaded directly into the active file system for the simulation, all tools are first cleared and then, if tool management is active, the file TO\_ADDON.INI (in HMI\_ADV\DP\SIM) is loaded into the simulation and executed.
  - MD 11280 WPD\_INI\_MODE = 0 (as before)  
A TOA file with the workpiece name (search sequence NC, then HMI) is loaded into the simulation, otherwise the first TOA file for the workpiece (search sequence NC, then HMI) is loaded into the simulation.

The part program is then loaded into the simulation and selected.

If another part program from the same workpiece is selected for the simulation, it is loaded into the simulation together with the associated data (if MD 11280 WPD\_INI\_MODE = 1) and then selected.



If a part program from another workpiece is selected, the previous workpiece is removed from the simulation.

If a workpiece is selected for the simulation, the part program with the workpiece name is loaded into the simulation together with its data and the workpiece selected. If the part program with the workpiece name does not exist, an error message is output.

## Channel change

Only one channel/program at a time can be simulated in the simulation, i.e. direct synchronization across channels is not possible.

Using the "Channel/spindle" softkey, programs can be assigned to specific channels to be executed one after another.

The preset channel is channel 1 or the first channel in the simulation. Before programs are selected, the simulation is switched to the channel for the current program.

MD 11280 WPD\_INI\_MODE = 1:

The system checks whether a TOA file with the current program name exists. If so,

- the tools are cleared
- the TOA file is loaded into the active file system for the simulation and
- (if tool management is active), the file TO\_ADDON.INI (in HMI\_ADV\DP\SIM) is loaded into the simulation and executed.

All files (RPA, GUD, SEA, UFR, PRO, CEC, TMA) with the part program names are then loaded into the passive file system in the simulation.

The programs in the channel/spindle list are executed one at a time in the specified sequence with each "NC start" of the simulation; with each program change the graphics status is retained as a blank specification for the next program. Hidden programs are skipped.

After the last program the graphic is cleared – after prompting – and the simulation starts again with the first program.

## 2.6.2 Display of the simulation data

The files loaded into the simulation are displayed with softkey **Match data** → **Loaded data**.

### User programs

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.

### 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.

All user cycles loaded into the simulation are displayed.

### 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.

All manufacturer cycles loaded into the simulation are displayed.

### 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.

All user cycles loaded into the simulation are displayed.

## 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.

### 2.6.3 Expansion of the tool geometry data of the simulation

Previously tools could be shown with their cutting edge, shaft and holder, which were calculated internally from data from the NC.

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 definition is produced by means of the **SIMTOGEO.INI** file, 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:

- Activation the "flying edge" function in DPTWP.INI or DPMWP.ini
- Machine data: define memory for OEM parameters
- Create SIMTOGEO.INI file, enter link in file

## Parameterization

The SimToGeo.ini file must contain an entry that establishes a link to the NCK tool data. This is located in the [Settings] section.

Link=TC\_DPCx    where x = 1 to 10 or

Link=TC\_DPCSx    where x = 1 to 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.

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 then appear below the corresponding identifier.

Example: [500.1] ; Roughing tool 1  
tool\_type = 500  
...

**Optional parameters for orienting milling cutters:**

TC\_DPV3= ; 1 = +X, -1 = -X  
TC\_DPV4= ; 1 = +Y, -1 = -Y  
TC\_DPV5= ; 1 = +Z, -1 = -Z

Only **one** of these entries may be used. A free tool orientation cannot be entered.

---

**Note**

- Tools are still displayed in the 3 main planes, XY, YZ or ZX.
  - If a tool cannot be displayed, a polymarker is displayed.
  - With the SimToGeo.ini file the tool\_type can be defined differently from the NC settings. If the corresponding cutting edge parameters (e.g. \$TC\_DPCS1[t,d]) can be set, a link to/evaluation of SIMTOGEO.INI geometrical data is also possible.
  - A complete sample file with entries for each permissible tool type can be found in Subsection 2.5.24 SIMTOGEO.INI.
- 

**Machine data settings**

MD 18206: MM\_NUM\_CCS\_TOA\_PARAM=1  
; Number of OEM data per cutting edge 0 to 10  
MD 18204: MM\_TYPE\_CCS\_TOA\_PARAM[0]=4  
; Data type 4 = Real  
MD 18080: MM\_TOOL\_MANAGEMENT\_MASK='H4'  
; Set aside memory for OEM data

**Startup procedure**

1. Activate the "Flying edge" function in DPTWP.INI or DPMWP.INI, for lathes or milling respectively, [Setup] section:

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

2. Define machine data or memory for OEM parameters:

```
$MN_MM_TOOL_MANAGEMENT_MASK=4 ; For SimToGeo.ini
; Set bit 2=1
```

If MD have already been assigned, the bit must also be 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 value of this variable determines the geometrical block that is used.

For example, the entry for tool 1 cutting edge 1 (T1D1) \$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.

Add expanded geometrical tool data.

This is done in blocks, which are distinguished by float number identifiers, for instance. 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 type500 SL3 L1=100, L2=55, R1
tool_type=500          ;Tool type then $TC_DP1 not required
insert_length=10       ;TC_DP8, insert length
holder_angle=105       ;TC_DP10, holder angle, not rake angle
reference_direction=3   ;TC_DP11, cutting angle
clearance_angle=5      ;TC_DP24, clearance angle
```

### 4. 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
```

## 2.6.4 Data match of the simulation

The data to be evaluated by the simulation must be located in the data management system in \DP.DIR\SIM.DIR. They are read each time the simulation application is started.

---

### 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 cannot be retrieved from the NC.

---

The **Data matching** menu provides the following functions:

### 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 MD 11210:  
UPLOAD\_MD\_CHANGES\_ONLY = 0.

Data are not activated until the simulation is restarted!  
The data are stored in the data management system in \DP.DIR\SIM.DIR.

---

### Note

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 file is copied from the NC.

MATCH_NCU_ENVIRONMENT=1    (default)
; Definition files in the
; data management directory are also matched with the NC.
```

---

## 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 starts up.

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.

## Match setup data

The following data selection can be used to load data from the active file system in 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 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 in \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 (Abort/Repeat dialog box). 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.

## 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.

To support this arrangement in the orientation of graphics tools, the SIM graphics module DPSMASCH.DLL has been expanded to include, among other things, an almost universal evaluation of the TOA parameters.

For example, `$TC_DP3[n,m]=...` can also be evaluated with negative values, with the convention that by default, **positive lengths** generally lead to machining "behind the turning center" and **negative lengths** to machining "in front of the turning center".

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 in SIM
ENDIF
...
```

**Use** (milling and lathe version):

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 lathe 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 lathe 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 for the entry has been 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 have been made directly accessible on 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.

---

### Use (lathe version only):

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

1. **Input monitoring "Unwinding diameter surface area"**  
In the lathe 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 unwinding diameter).
2. **Tool center-point path ("without tool data")** is retained temporarily when a tool data source (e.g., "with tools", etc.) is selected subsequently.  
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.

## 2.6.5 Memory configuration for the simulation

### 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.**

Negative values are ignored.

### DPSIMNCK.INI [DP\_SIMNCK]

The following entries with default values are contained in the [DP\_SIMNCK] section:

```
...
[DP_SIMNCK]
DP_SIMNCK_MM_NUM_GUD_MODULES_OFFSET = 0
    corresponding machine data:
    18118 MM_NUM_GUD_MODULES

DP_SIMNCK_MM_NUM_GUD_NAMES_NCK_OFFSET = 5
    corresponding machine data:
    18120 MM_NUM_GUD_NAMES_NCK

DP_SIMNCK_MM_NUM_GUD_NAMES_CHAN_OFFSET = 30
    corresponding machine data:
    18130 MM_NUM_GUD_NAMES_CHAN

DP_SIMNCK_MM_GUD_VALUES_MEM_OFFSET = 20
    corresponding machine data:
    18150 MM_GUD_VALUES_MEM

DP_SIMNCK_MM_NUM_USER_MACROS_OFFSET = 10
    corresponding machine data:
    18160 MM_NUM_USER_MACROS

DP_SIMNCK_MM_NUM_MAX_FUNC_NAMES_OFFSET = 40
    corresponding machine data:
    18170 MM_NUM_MAX_FUNC_NAMES

DP_SIMNCK_MM_NUM_MAX_FUNC_PARAM_OFFSET = 350
    corresponding machine data:
    18180 MM_NUM_MAX_FUNC_PARAM

DP_SIMNCK_MM_MAX_SIZE_OF_LUD_VALUE_OFFSET = 400
    corresponding machine data:
    18242 MM_MAX_SIZE_OF_LUD_VALUE
...
```

## 2.6.6 Accelerating simulation startup

### Simulation startup

The simulation application is normally started only on request by the operator. This prevents the system from being overloaded by the software components that have to be loaded.

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).

### Loading during HMI startup

To accelerate simulation startup, the loading times can be transferred to the HMI startup. 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 during SimNCK startup

Configuration files for the NC environment (tool data, cycles) are loaded when the SimNCK is started. This means that only the program to be executed and its subroutines need to be loaded into the simulation NCK. This startup behavior is controlled by means of the [PRELOAD] and [MAIN] sections in the DPSIM.INI file. The simulation start time can also be reduced as follows:

**[PRELOAD]**

CYCLES=1

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

TOOLS=1

The files for the tool mapper (TO\_INI.INI and TO\_ADDON.INI) are preloaded when the simulation NCK is started (if applicable).



## Notes

# 3

## 3 Creating User Alarm Texts

3.1 Creating Alarm Texts .....	IM4/3-136
3.2 Structure of the User Alarms .....	IM4/3-138
3.3 Alarm Texts in Foreign Languages.....	IM4/3-139
3.4 Plain Texts for PLC Machine Data.....	IM4/3-141
3.5 NCU-Specific Alarm Texts (only SINUMERIK powerline) .....	IM4/3-142

## 3.1 Creating alarm texts

### Storing the text files

Files containing error texts are stored on the hard disk in directory F:\dh\mb.dir\ . The error text files provided for use are activated in file f:\mmc2\mbdde.ini.

### Standard files

The standard texts in ASCII format are stored on the hard disk of the HMI in the following files:

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

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

If these files are replaced by new ones when the HMI software is upgraded, then the added or modified user-specific alarms will be lost. Users should store their own error texts in user files.

---

#### Note

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

---

### Creating user alarms

The user can replace the stored error texts in the standard files with his own texts. Likewise, the user can add new error texts.

To do this, the additional files must be imported into directory **f:\dh\mb.dir** (MBDDE alarm texts) via the Services operating area. The names of these text files are set in the **f:\mmc2\mbdde.ini** file. An editor is provided in the **Diagnostics\Startup\HMI** area for this purpose.

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

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

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.

### Structure of MBDDE.INI

Extract from **MBDDE.INI** relating to the configuration of alarm text files:

```
...  
[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=f:\dh\mb.dir\myplc_  
UserZyk=  
UserCZyk=  
...
```

### Editor

Any **ASCII editor** (e.g., the DOS editor **edit**) 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 in alarm texts do not take effect until the HMI is started up again. When creating text files, make sure that the correct date and time are set on the PC. Otherwise, the user texts may not appear on the screen.

---

### Help on alarms

Creating user-specific online help texts is described in /HE1/ Installation and Start-Up Guide HMI Online Help.

## 3.2 Structure of the user alarms

### Alarm numbers

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

Number range	Name	Effect	Deletion
60000 - 60999	Cycle alarms (Siemens)	Display, NC start disable	Reset
61000 - 61999		Display, NC start disable, stop motion after executing the pre-decoded blocks	Reset
62000 - 62999		Display	Cancel
63000 - 64999	Reserved		
65000 - 65999	Cycle alarms (user)	Display, NC start disable	Reset
66000 - 66999		Display, NC start disable, stop 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 channel		
600000 - 699999	PLC alarms for axis and spindle		
700000 - 799999	PLC alarms for user		
800000 - 899999	PLC alarms for sequence cascades/graphs		

### Format of text file for cycle alarm texts

The number range indicated in the list is not available with every number (see **Reference:** /FB/ P3 "Basic PLC Program", Lists)

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

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



### 3.3 Alarm texts in foreign languages

#### Language-specific nature of alarm texts

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. The language code is available from the table below.

#### Configuring user alarm texts

If alarm and message texts are created externally, they must be produced with the code pages, ANSI tables, and fonts listed in the tables below.

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 copy it to the "MBDDE alarm texts" directory for the HMI for the target.

#### Supported languages

Table 3-1 Languages and ANSI table/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	RCCB	850	1252
French	FR	850	1252
Italian	IT	850	1252
Dutch	NL	850	1252
Portuguese	PO	850	1252
Swedish	Software	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	JA	932	-
Chinese (Simplified)	CH	936	-
Korean	KO	949	-
Chinese (Traditional)	TW	950	-

## Assignment between ANSI tables and fonts

Table 3-2 ANSI table 1250 (Central Europe)

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

Table 3-3 ANSI table 1251 (Cyrillic)

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

Table 3-4 ANSI table 1252 (West Europe)

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

Table 3-5 ANSI table 1254 (Turkish)

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

During system startup, 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 you want to configure other languages in the control, a "Language selection" softkey can be configured (see "Configuring the 'Language Selection' Softkey"). 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.

## 3.4 Plain texts for PLC machine data

### Overview

Application-specific and language-dependent plaintexts can be configured for each index for the PLC machine data.

#### Example:

14510[i]	User data (INT)	i = 0 to 255
14512[i]	User data (HEX)	i = 0 to 255
14514[i]	User data (FLOAT)	i = 0 to 31

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

Entries in the following format:

<number of entries>		; number of entries
<MD-number>.<index>	<text>	; number of index texts

#### Result:

```
14510.7 My text for PLC-MD 14510.7
14512.0 This is the text for 14512.0
14514.1 My text for float data 14514.1
```

---

#### Note

PLC operational messages are created and handled in the same way as alarm texts.

---

## 3.5 NCU-specific alarm texts (only SINUMERIK powerline)

### Function

If multiple PCUs and NCUs (M:N) are used, NCU-specific alarm texts can be created and managed in the HMI.

This function requires entries to be made in the MBDDE.INI file in the [netTextFiles] section. The file is located in the C:\USER directory.

The entries for the NCUs must match the entries in the [paramNCU\_Bez] section of the current NETNAMES.INI file.

The NETNAMES.INI file in connection with the "Multiple operator panel fronts/NCUs" function is described in

**Reference:** /FB2/ B3 Multiple Operator Panel Fronts and NCUs.

### Search sequence

To output an NCU-specific alarm, the associated text is searched for in the following sequence:

1. NCU-specific texts
2. Non-NCU-specific texts

### Path for alarm texts

Like the other alarm texts, the NCU-specific alarm texts (e.g. NC1\_GR.COM) should be stored in the C:\DH\MB.DIR\ directory.

### Entry in MBDDE.INI

Example:  
[net TextFiles]  
NCU\_1=C:\DH\MB.DIR\NC1\_GR.COM  
NCU\_2=C:\DH\MB.DIR\NC2\_GR.COM  
NCU\_3=C:\DH\MB.DIR\NC3\_GR.COM



# 4

## 4 Diagnosis and Service

4.1 Installing HMI Software on a PG/PC.....	IM4/4-144
4.1.1 NCU Connection Wizard.....	IM4/4-145
4.1.2 Starting HMI Advanced in a Separate Desktop.....	IM4/4-145
4.1.3 Configuring the Desktop Switch Application.....	IM4/4-147
4.1.4 Operating the Desktop Switch Application .....	IM4/4-148
4.2 Action Log .....	IM4/4-150
4.2.1 Displaying the Log File .....	IM4/4-153
4.2.2 Displaying the Log File .....	IM4/4-154
4.3 HMI-Analyzer .....	IM4/4-155
4.3.1 Operating .....	IM4/4-156
4.3.2 Analyzing Data.....	IM4/4-158
4.3.3 Examples of Applications.....	IM4/4-161

## 4.1 Installing HMI Software on a PG/PC

### Requirements

Target system	PG oder PC mit 600 MHz, 256 MB RAM Available hard disk space 500 MB
Operating system	MS Windows XP Professional
MPI driver	Version 6
NC interface (SINUMERIK powerline)	CP5511, CP5611
NC interface (SINUMERIK solution line)	Network connection (Ethernet)

"Small fonts" must be selected under "System Control → Display, Settings".

### Software

- Name: HMI Advanced for PC/PG
- Installation language: German/English
- Distribution medium: CD
- Uninstallable: 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.

---

#### Notice

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.

---

## 4.1.1 NCU Connection Wizard

### Application

The NCU connecting assistant 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 connecting assistants, 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 connecting assistant 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.

---

## 4.1.2 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 taskbar
- Changes to color scheme
- Focus control in connection with OEM software
- All applications of a standard PC run on one desktop.

How to call up HMI Advanced in a separate desktop is described as a remedy in the following section.

### Remedy

Through the dynamic creation of a separate desktop for the HMI Advanced software, the features indicated above can be 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 with the **HMIDesk.ini** file.

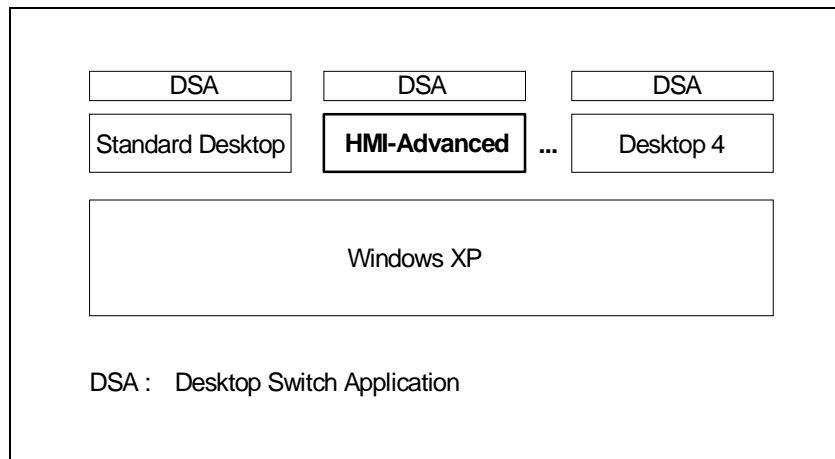


Fig. 4-1: Separate desktops for HMI Advanced and other applications

---

### Notice

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 taskbar.
  2. When a dynamic desktop is closed, the application that is started automatically by inclusion in `hmidesk.ini` is closed with the desktop. 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 process 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.
-



### 4.1.3 Configuring the desktop switch application

#### [Global] section

The application is configured in the **HMIDesk.ini** file.

In [Global], the number of desktops required 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].

<b>ControlMode</b> =	Icon	Icons in the taskbar (see Explorer below)
	Window	Standalone dialog window (see WindowOnTop below)
<b>KeyboardControl</b> =	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.

---

<b>Explorer</b> =	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.
<b>WindowOnTop</b> =	TRUE	If ControlMode = Window appears, the dialog window of the desktop switch application is constantly displayed.
	FALSE	Dialog window can be covered

<b>CreateDesktop =</b>	Immediately Delayed	Start desktop immediately. Start desktop when selected.
<b>SetHMIColors =</b>	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.

## Activation

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.

## Default behavior when creating a desktop

An active desktop is created with the taskbar. The desktop switch window is hidden. The desktop switch is operated via the task bar.

## 4.1.4 Operating the desktop switch application

### Overview

The user interface for the desktop switch application can be configured:

- Keyboard hotkeys
- Dialog window
- Icons in the taskbar

Hotkeys:

- <Alt+1> Switches to desktop 1 (standard desktop of user)
- <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)

## HMI startup in separate desktop

First start a second desktop, then HMI Advanced:



Fig. 4-2: Linking to the desktop switching application

Switching by clicking on the corresponding desktop number:

- 1 Standard desktop
- 2 Desktop with HMI Advanced
- (3, 4) Other optional desktops




---

### Note

If WindowOnTop = False is configured in hmidesk.ini, the window must be brought to the foreground using <Alt + TAB>.

---

## 4.2 Action log

### Application

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.

The following events can be logged:

<b>Variable services</b>	All variables that are written via an input field (e.g., write accesses to geometry data such as tool compensation 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.
<b>PI services</b>	All PI services (e.g. program selection, file deletion, tool deletion) can be logged. A description of the variables can be obtained via the Help function in the Parameters operating area under Variable Views.
<b>Domain services</b>	All domain services (e.g. load/unload program) can be logged.
<b>Alarm status</b>	All NC/PLC and HMI alarms and messages that are displayed can be logged.
<b>Channel status/override</b>	NC/PLC statuses can be logged using information about the channel status and the position of the override switch. 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.
<b>HMI events</b>	Function buttons for screen selection and switching the operating area (Action StateReached).

The following functions are available:

- Logging of alarms
- Output of log file via USB
- The log file output is password-protected (access level: Manufacturer).
- The action log is enabled/disabled via MD 9012: ACTION\_LOG\_MODE or in the operating area "Commissioning" → "HMI" → "System Settings" in the "Action Log Settings" screen.

<b>9012</b>	<b>ACTION_LOG_MODE</b>		
MD number	Action log settings		
Default value: 254	Min. input value: 0	Max. input value: 0xffff	
Changes effective after POWER ON	Protection level (R/W) 2/2	Unit: -	
Data type: DWORD	Applies from version 5.2		
Meaning	This MD can be used to enable/disable the action log and to define a choice of the data to be logged.		
	Bit 0	= 1	Action log ON (default)
		= 0	Action log OFF
	Bit 1	= 1	<b>HMI Advanced:</b> Variable services (writing accesses to geometric data) are logged. For the meaning of the variables, see Help in the Parameter operating area under variable views (standard).
		= 0	Variable services are not logged.
	Bit 2	= 1	<b>HMI Advanced:</b> PI services (e.g. program selection) are logged. For the meaning of the variables, see Help in the Parameter operating area under variable views (standard).
		= 0	PI services are not logged.
	Bit 3	= 1	<b>HMI Advanced:</b> Domain services (e.g. load/unload program) are logged (standard).
		= 0	Domain services are not logged.
	Bit 4	= 1	Changes to the alarm status are logged (default).
		= 0	Changes in the alarm status are not logged.
	Bit 5	= 1	Key actuations are logged: only ShopMill, ShopTurn, ManualTurn (standard).
		= 0	Key actuations are not logged.
	Bit 6	= 1	Channel status/override is logged (default).
		= 0	Channel status/override is not logged.
	Bit 7	= 1	<b>HMI Advanced:</b> Softkey actuations and menu changes are logged (standard) Only for use by Siemens.
		= 0	Softkey actuations and menu changes are not logged.

## Setting the action log

Via the operating area "Commissioning" → "HMI" → "System Settings," select which actions of the control should be logged in the "Action log settings" (also see Subsection 2.5.17 MMC.INI):

- Logging on/off
- Logged data
- Action log settings:
  - Write interval file:
    - 1: (default) The log is only saved to hard disk upon request, e.g. trigger.
    - > 0: The longest waiting time in seconds before a log file is saved on hard disk.
  - File size of log file: 40 KB (default).
  - Write trigger via alarms: 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.
  - The path of the log file is permanently set and cannot be changed

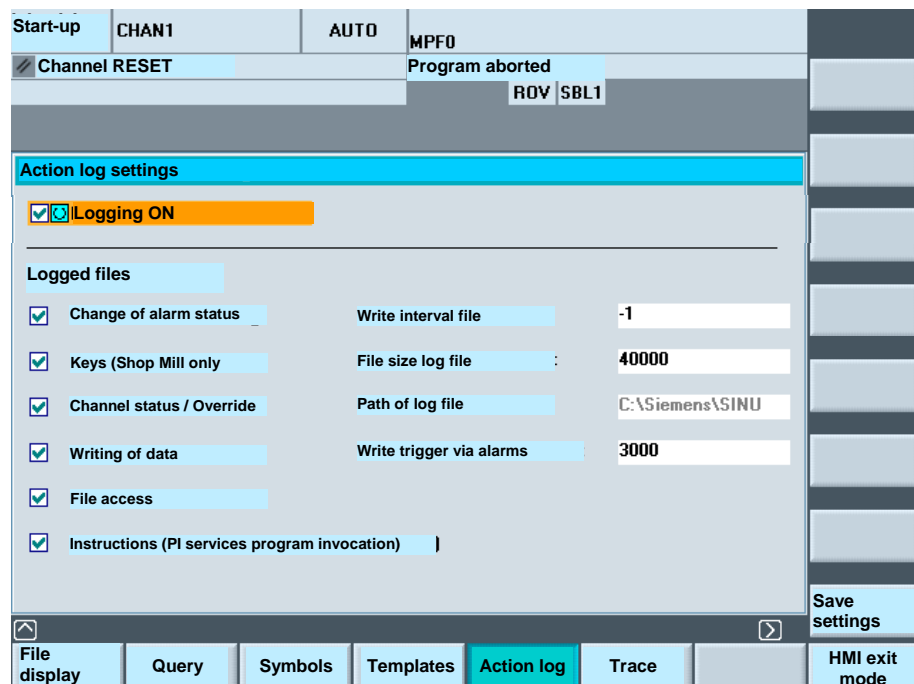


Fig. 4-3: Action log settings

The "Keys (only ShopMill)" selection is without function in HMI Advanced (also see MD 9012, Bit 5).

## 4.2.1 Displaying the log file

### Overview

The log file output is password-protected (access level: Manufacturer). On the operator panel, the current log file is displayed on the screen in the operating area "Diagnostics" → "Service displays" → "Action log".

The log is output in English with the following columns:

- Date
- Time
- Name of NCU
- Event

### Example of a log file:

Date	Time	NCU	Event
01.09.05	06:22:48	local	
=====			
==			
01.09.05	06:22:48	local	ACTIONLOG: HMI Booting (OutputSelect:0xff CommitLatency:-1 FileSize:40000)
01.09.05	06:22:48	local	ACTIONLOG: Monitored Variable:"/Channel/ProgramInfo/msg" Mode:0x1
01.09.05	06:22:48	local	ACTIONLOG: Monitored Variable:"/plc/datablock/byte[c19,0]" Mode:0x81
01.09.05	06:22:48	MachineSwitch	ACTIONLOG: variable "/plc/Datablock/Byte[c19,0]" Mode:0x81
01.09.05	06:22:48	MachineSwitch	ACTIONLOG: variable "/Channel/ProgramInfo/msg" = ""
		:	
		:	
01.09.05	06:24:52	NCU840D	application:RE;state:-992;softkey:-1;etc:0
01.09.05	06:24:52	NCU840D	application:RE;state:-990;softkey:-1;etc:0
01.09.05	06:25:05	NCU840D	application:RE;state:-991;softkey:-1;etc:0
01.09.05	06:37:58	local	ACTIONLOG: HMI Terminates

## 4.2.2 Outputting the log file

### Application

The log file can be read out from the startup menu. A startup menu is displayed whenever a crash occurs (PLC Bit DB19, Byte 0, Bit 6). The PLC must set and reset the bit at the user interface (same as, e.g., key lock, screen brightness control, etc.).

The HMI evaluates the rising edge of the signal (i.e., a transition from 0 to 1) and creates the log file when the signal arrives. The HMI only read-accesses the data, i.e., the signal must be reset from the PLC/user program. If the bit is already set when the HMI starts up, the HMI does not respond to the value of the bit (until it is reset and set again).

### Reading out the file

To read out the log files, proceed as follows:

1. In the Service menu choose the "A\_\_Actionlog" menu (press the "A" button). The "A\_\_Actionlog" menu is only available if a log file exists.
2. The current log files (AC\_LOGA.TXT and AC\_LOGB.TXT) are copied to the part program directory. The following message appears: "The action log (crash) is filed in the part program directory for output (files AC\_LOG?.TXT).
3. Press any key to return to the main menu.
4. Once the controller is restarted, you can read out the log files from the "Services" operating area in the "Data Out" -> Via USB menu.

### Reading out a file after a system crash

If events cause the control to crash, crash files are created from the log files in the data management system.

If you are unable to select the Service menu, you should proceed as follows:

1. Switch on the control.
2. During startup, a log file is scanned for crash files. The current crash files (AC\_LOGP1.TXT to AC\_LOGPn.TXT) are copied to the part program directory.
3. After the controller is restarted, you can read out the log files from the "Services" operating area in the "Data Out" → via USB menu.



## 4.3 HMI Analyzer

### Application

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.

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 = MMC2
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.

## 4.3.1 Operating

### 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).

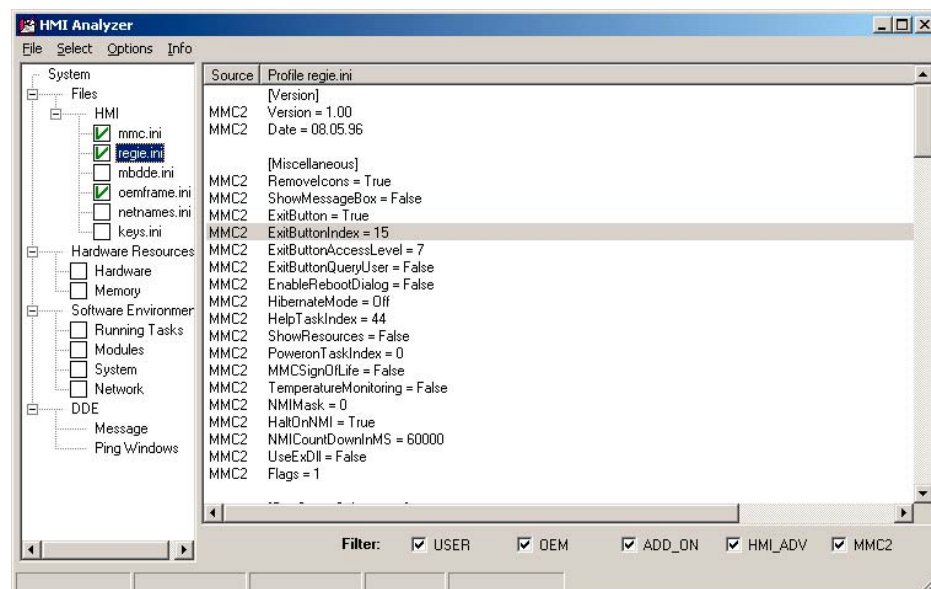


Fig. 4-4: Example of .ini files

### 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 via the key "Add →" 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 "← 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.

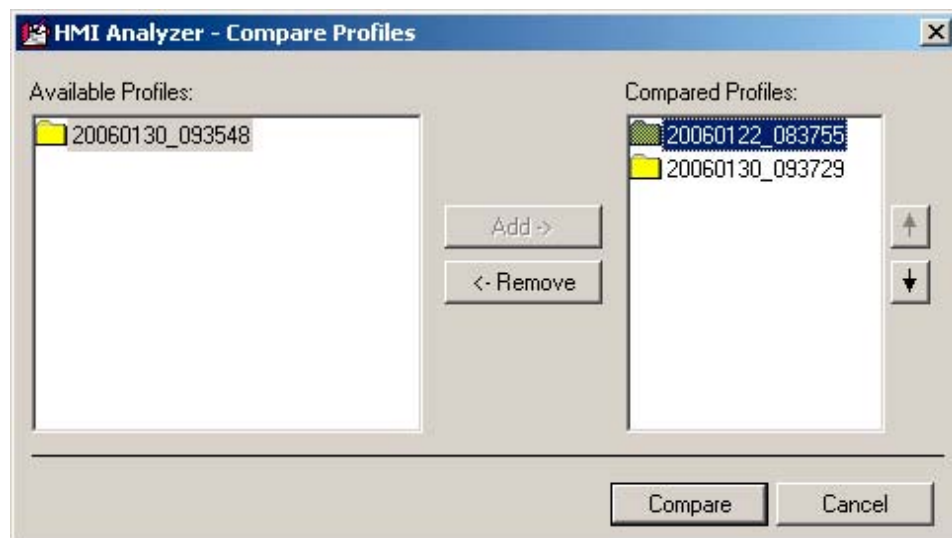


Fig. 4-5: Comparing profiles

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".

### 4.3.2 Analyzing data

#### 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 running, '-' Process 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 reanalyzed.

**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.

---

#### 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. 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".

---

## System → DDE

**Message** The DDE message records DDE notifications. Depending on the settings, requests, pokes, executes and hotlinks (advise) are logged. With the option "Only Active Hotlinks" only the currently active hotlinks can be displayed. Finished hotlinks disappear from the screen.

Only DDE notifications that occur after the start of the HMI Analyzer are recorded. The elapsed time can be determined from the "Time" column. This column displays the time that has elapsed since the start of windows in milliseconds.

**Ping Windows** With "Ping Windows", you can check whether a window does not reply to DDE broadcasts. When a hotlink is started, the system sends a notification to all windows which must be immediately answered by all windows.

If a window answers very late or not at all, the operation of HMI Advanced can become slower or even stop completely.

Single	The view is updated once as a static display. The answers from all the windows in the system are listed.
Cyclic	The view is cyclically updated. Notifications are sent cyclically to all windows and must be answered. The update rate can be specified in milliseconds [ms] to the left next to the "Cyclic" selection.
Reset	Deletes the display.
Cycle time	The waiting time between the individual run-throughs during cyclic analysis
Answer Time	The time within which a window must answer so that it is not listed when the "Show All" option is selected.
Show All	Shows the log of the ping attempts with the answer time of all the windows. When option is deselected, only those windows are displayed which do not answer within the specified time ("Answer Time")
Append	The log of each individual run-through depends on the previous one. When the option is deselected, the log window is deleted before each run-through

---

### Note

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).

---

### 4.3.3 Examples of applications

#### (1) Tied-in OEM application doesn't start

**Problem:**

An individual OEM application was tied-in to 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>

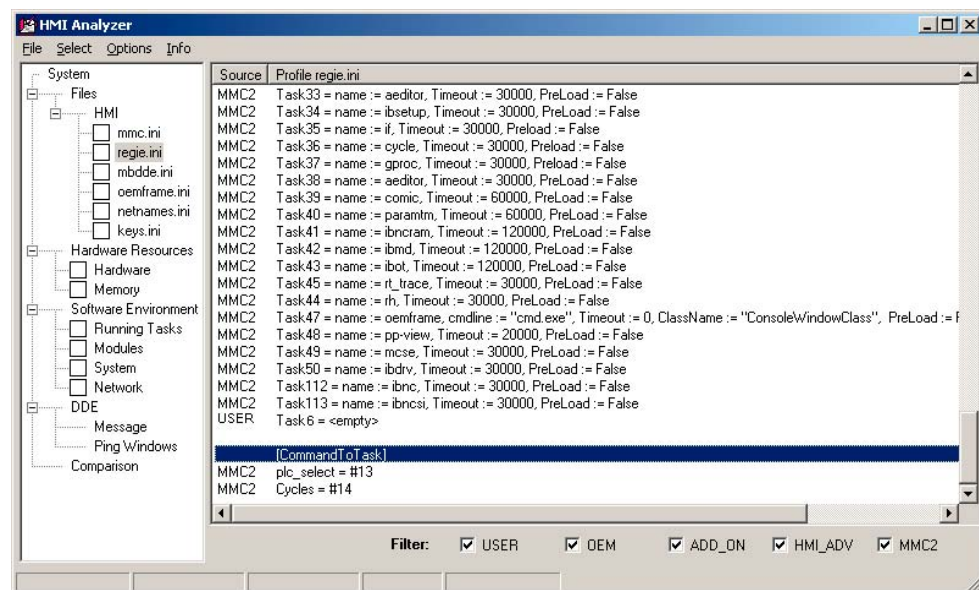


Fig. 4-6: Check the start of the OEM application

**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
```

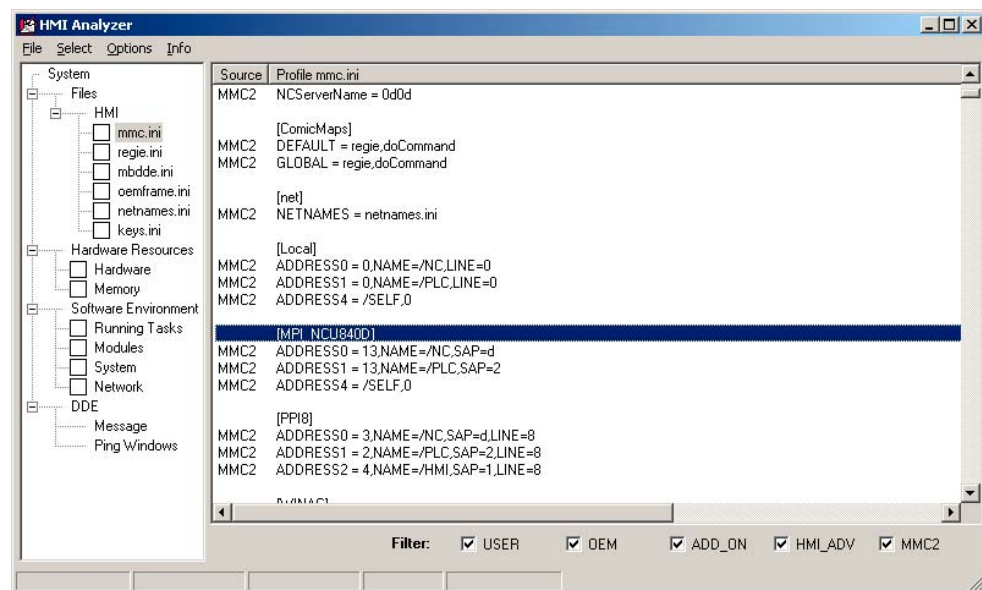


Fig. 4-7: Checking the connection to controller

### HMI Analyzer:

Under Files → HMI, the mmc.ini file 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 lethargic 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 lethargic or blocked area switcher.

**Regulation:**

"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.

---

**Notes**

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 carried out. 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 slowed down extremely by this.

**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 controller 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 started up, 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 controller 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 for OPC DataAccess programmers**

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.

---



# I Index

## I.1 Subject Index

### \$

\$MM\_MD\_DRV\_TEA  
\_FILTER = 18351 ..... IM4/M2-35

### A

Access authorizations  
  Creating programs ..... IM4/2-56  
  Drives ..... IM4/2-46  
Access protection for basic offset.. IM4/2-38  
Action log..... IM4/4-150, IM4/4-153  
Activating data transfer..... IM4/2-14  
Actual value  
  Displaying..... IM4/2-32  
  Setting ..... IM4/2-33  
Adding to service overview..... IM4/2-66  
Alarm texts  
  Languages ..... IM4/3-139  
  NCU-specific ..... IM4/3-142  
  Storage..... IM4/3-136  
ANSI tables/fonts..... IM4/3-139  
Axis names/axis values ..... IM4/2-121

### B

Block search ..... IM4/2-18  
Browse drives ..... IM4/2-47

### C

Configuring displays in the  
header ..... IM4/2-20  
Configuring OEM icons ..... IM4/2-31  
Configuring OEM texts .. IM4/2-30, IM4/2-86  
Configuring screen saver ..... IM4/2-77  
Creating operator menu ..... IM4/2-59

### Cycles

  Simulation ..... IM4/2-120  
Cycles in the DRAM..... IM4/2-35

### D

Data management ..... IM4/1-9  
Data matching in simulation..... IM4/2-128  
Definition files..... IM4/2-37  
Desktop switch..... IM4/4-145  
Displaying server names ..... IM4/2-77  
Displaying spindle utilization..... IM4/2-40  
Double channel display..... IM4/2-70  
Drives  
  Connection via MD ..... IM4/2-45

### E

Error messages to the PLC ..... IM4/2-15  
EXIT softkey..... IM4/2-76

### F

Filter for alarms ..... IM4/2-55  
Flying edge ..... IM4/2-125

### H

Hibernate ..... IM4/2-107  
Hiding axes ..... IM4/2-38  
HMI functions ..... IM4/2-13  
HMI Analyzer  
  configuring ..... IM4/4-155, IM4/4-156,  
  IM4/4-158, IM4/4-161

### I

INI files (overview) ..... IM4/2-79

Input monitoring..... IM4/2-131  
 Inverting spindle icon..... IM4/2-43

**J**

Job list ..... IM4/2-68

**K**

Keyboard behavior ..... IM4/2-44

**L**

Language  
   Installing ..... IM4/3-140  
   Selecting..... IM4/3-140  
 Language support ..... IM4/3-139  
 Log file ..... IM4/4-153

**M**

Match cycles..... IM4/2-129  
 Match setup data..... IM4/2-129  
 Match tools ..... IM4/2-128  
 Multi-channel sequential  
 programming ..... IM4/2-103, IM4/2-109

**N**

NC language commands ..... IM4/2-38  
 NCU Connection Wizard ..... IM4/4-145

**O**

Operational messages (PLC) ..... IM4/3-141  
 Optimization..... IM4/2-131

**P**

Parameterization of INI files ..... IM4/2-51  
 Plain text identifier ..... IM4/3-141  
 PLC  
   Acknowledging alarms ..... IM4/2-57  
   Channel selection..... IM4/2-17  
   Machine data..... IM4/3-141  
   Programming..... IM4/2-13  
   Spindle selection ..... IM4/2-17  
 Protection levels  
   Concept..... IM4/1-9  
   NC commands ..... IM4/2-38

**R**

Rotating coordinate system ..... IM4/2-130

**S**

Saving in the Flash File System  
 (FFS)..... IM4/2-36  
 Scratching ..... IM4/2-33  
 Search configuration ..... IM4/2-19  
 Select language softkey..... IM4/2-58  
 Setting alarm log ..... IM4/2-54  
 Setting technology ..... IM4/2-74  
 Simulation  
   Channel change ..... IM4/2-123  
   Coordinate system ..... IM4/2-130  
   Data ..... IM4/2-119  
   Match cycles..... IM4/2-129  
   Match MDs..... IM4/2-128  
   Match setup data ..... IM4/2-129  
   Match tools ..... IM4/2-128  
   Mirroring tools ..... IM4/2-130  
   NC cycles..... IM4/2-120  
   Program selection..... IM4/2-122  
   Search strategy..... IM4/2-122  
   Startup ..... IM4/2-120  
 Simulation data display ..... IM4/2-124  
 Startup  
   Affecting..... IM4/1-8  
   HMI Advanced ..... IM4/1-8  
   Simulation ..... IM4/2-120, IM4/2-133  
 Status display with icons..... IM4/2-24  
 System limits..... IM4/1-9

**T**

Templates ..... IM4/2-68  
 Tool center-point path ..... IM4/2-131  
 Tool management (WZV) ..... IM4/2-71  
 Tool offset ..... IM4/2-34  
 Tool selection with tool  
 management..... IM4/2-72

**U**

User control file ..... IM4/2-14

**W**

Work offset..... IM4/2-32



CMM\_DIRECTORY\_SOFTKEY  
\_PATH1 ..... IM4/2-45  
MD 9677  
CMM\_DIRECTORY\_SOFTKEY  
\_PATH2 ..... IM4/2-45  
MD 9678  
CMM\_DIRECTORY\_SOFTKEY\_PATH3..  
IM4/2-45  
MD 9679  
CMM\_DIRECTORY\_SOFTKEY\_PATH4..  
IM4/2-45  
MMC.INI ..... IM4/2-99

**N**

NETNAMES.INI ..... IM4/2-77, IM4/2-103

**O**

OEMFRAME.INI ..... IM4/2-105

**P**

PARAM.INI ..... IM4/2-106  
PARAMTM.INI ..... IM4/2-106  
PRESET ..... IM4/2-33

**R**

REGIE.INI  
Hibernate ..... IM4/2-107  
HMI startup ..... IM4/2-133  
Requesting a RESTART ..... IM4/2-76  
Softkey EXIT ..... IM4/2-76

**S**

SEDITOR.INI ..... IM4/2-109  
SimNCK ..... IM4/2-133  
SIMTOGEO.INI ..... IM4/2-115, IM4/2-125  
STAT (moving joint position) ..... IM4/2-35

**T**

TASKCONF.INI ..... IM4/2-117  
TU (rotary axis position) ..... IM4/2-35



# SINUMERIK 840D sl/840D/840Di/810D

## Commissioning CNC Part 2 (HMI)

### Creating Foreign Language Texts with Windows 2000 / XP (TX2)

#### Valid for

##### *Controller*

SINUMERIK 840D sl/840DE sl  
SINUMERIK 840D powerline/840DE powerline  
SINUMERIK 840Di powerline/840DiE powerline  
SINUMERIK 810D powerline/810DE powerline

##### *Software*

HMI Advanced  
HMI-Embedded

##### *Software version*

7.1  
7.1





<b>1 Introduction</b> .....	<b>TX2/1-5</b>
1.1 General .....	TX2/1-6
1.2 Prerequisites .....	TX2/1-6
1.3 Language families .....	TX2/1-7
1.4 Tools used .....	TX2/1-8
1.5 Definition of terms .....	TX2/1-9
<b>2 Editing Files with Word 2000/2003</b> .....	<b>TX2/2-13</b>
2.1 Overview .....	TX2/2-14
2.2 Create a new text file .....	TX2/2-14
2.3 Open an existing file .....	TX2/2-15
2.4 Save text files .....	TX2/2-17
2.5 Codes for SINUMERIK HMI .....	TX2/2-19
<b>3 Input of Text</b> .....	<b>TX2/3-21</b>
3.1 Special characters of different language families .....	TX2/3-22
3.2 Entering characters with the character map .....	TX2/3-23
3.3 Compound input .....	TX2/3-25
3.4 Using a different keyboard layout .....	TX2/3-26
3.4.1 Concept of the keyboard language property .....	TX2/3-26
3.4.2 Setting up a keyboard layout under Windows 2000 .....	TX2/3-27
3.4.3 Setting up a keyboard layout under Windows XP .....	TX2/3-28
3.4.4 Use .....	TX2/3-30
3.5 Numerical text input .....	TX2/3-32
3.6 Language properties in Word files .....	TX2/3-33
3.7 Multi-language Word files .....	TX2/3-36
3.8 Special considerations when working with Greek and Cyrillic texts .....	TX2/3-39
3.9 Special considerations when working with East Asian texts .....	TX2/3-40
<b>4 Problems and Remedies</b> .....	<b>TX2/4-45</b>
4.1 Problems and remedies .....	TX2/4-46

<b>5 References</b> .....	<b>TX2/5-49</b>
5.1 References for text creation.....	TX2/5-50
<b>I Index</b> .....	<b>TX2/I-51</b>
I.1 Index.....	TX2/I-51

# 1

## 1 Introduction

1.1 General .....	TX2/1-6
1.2 Prerequisites .....	TX2/1-6
1.3 Language families .....	TX2/1-7
1.4 Tools used .....	TX2/1-8
1.5 Definition of terms .....	TX2/1-9

## 1.1 General

Since the 1990s, Microsoft have put considerable effort into satisfying global demands for information processing and software development.

In addition to supporting as many languages as possible, one of the aims was to be able to support all languages with the same uniform operating system, as opposed to the language-specific versions of operating systems which were used before.

As part of these efforts, many of the system components and applications were gradually converted to the cross-language platform Unicode.

Although in many cases the user would have been completely unaware of this conversion process, in terms of the subjects discussed in this guide fundamentally different procedures need to be adopted depending on which version of the operating system is used.

The procedure recommended for Windows 2000/XP is much simpler than the one recommended for Windows 95/NT4. Therefore it makes sense to change over to the new systems. The guide for older systems is no longer being updated by the documentation department.

## 1.2 Prerequisites

This document describes the procedure for creating a text with the following preconditions:

- The text is to be created on a Microsoft Windows operating system (Windows 2000 or Windows XP).
- The text is to be created in a language not identical to the language of the operating system (= default language).
- The present description uses German as the default language for all examples provided herein. However, the described procedure will work just as well with a different language as the default Windows language.
- The text is to be created as a so-called text file, i.e. without formatting such as font types, font sizes, underscores etc. This means that, apart from the text characters, the file will only contain ASCII control characters like Tab, CR and LF (tab indent, carriage return and line feed).
- The language of the text requires a different Windows code page to that of the default language, i.e. the target language does not belong to the same language family as the default language.
- The text file is to be coded using this code page, not as a Unicode file.
- The means used to create the text are to be as simple as possible, i.e. especially without the need to install any additional software.
- For the purposes of this guide, it is assumed that you have installed Microsoft Word 2000 (or a later version) and are familiar with using it.

## Text types for HMI Embedded and HMI Advanced

The following text types can be changed in the individual systems:

Table 1-1

Text types	Systems
Alarm texts Adding texts for the user interface Application texts for all operating areas incl. softkey texts	HMI Embedded
Alarm texts Adding texts for the user interface Texts for the Help function in the editor Machine data Softkey texts for all operating areas	HMI Advanced

## 1.3 Language families

A language family is a group of languages for which Microsoft operating systems use the same code page. Microsoft uses the term "language group"

Table 1-2 Language families

Language family	Windows-code page	Languages
Central European	1250	Albanian, Croat, Polish, Romanian, Serbian (Latin), Slovakian, Slovenian, Czech, Hungarian
Cyrillic	1251	Bulgarian, Macedonian, Russian, Serbian (Cyrillic), Ukrainian, White Russian
Western	1252	German, English, Finnish, French, Indonesian, Icelandic, Italian, Dutch, Norwegian, Portuguese, Swedish, Spanish
Greek	1253	Greek
Turkish	1254	Turkish
Baltic	1257	Estonian, Lettish, Lithuanian
Japanese	932	Japanese
Chinese	936	Simplified Chinese
Korean	949	Korean
Chinese	950	Standard Chinese (Traditional Chinese)

## Operating system and target language from the same language family

If the language of the operating system (e.g. German) belongs to the same language family as the target language (e.g. Portuguese), then some difficulties could possibly occur with the input of certain characters. However, there will be no problems displaying these characters with a simple text editor.

Remedy:

- Use the Windows "Character Map"  
(see Section 3.2: Entering characters with the character map) or
- Switch over the keyboard  
(see Section 3.4: Using a different keyboard layout) or
- Use the ALT+digit combination to enter characters  
(see Section 3.5: Numerical text input)

There should not be any further problems in this case. Use a simple text editor (e.g. Notepad) and save the texts in the usual way.  
Section 2 does not apply.

## Operating system and target language from different language families

If the language of the operating system (e.g. German) does not belong to the same language family as the target language (e.g. Hungarian), difficulties will occur with both inputting and displaying the characters in a simple text editor (Notepad).

## 1.4 Tools used

### Standard version

This document describes the use of Microsoft Word 2000/2003 and other tools which may not necessarily be installed on your PC.

You should have installed the following components:

- Microsoft Word 2000/2003 (required)
- Windows 2000/XP: Language Support for East Asian characters (necessary for these languages)
- Windows 2000/XP: Character Map (recommended)

### Checking the installation of the Language Support for East Asian languages

Installation of the Language Support for East Asian languages is performed differently for Windows 2000 and Windows XP.

Please refer to Section 3.9 for details: Special requirements for East Asian languages

## 1.5 Definition of terms

### Code page

A list of characters which forms the character set for one or more languages. A code which is unique within this code page is assigned to each of the characters in the list.

All of the code pages used under Windows share the ASCII range (codes 0 to 127).

A distinction is made between single-byte code pages and multiple-byte code pages.

In single-byte code pages (all European languages), each character is represented with a single byte. Accordingly, single-byte code pages comprise a maximum of 256 characters.

Multiple-byte code pages (Asian languages) contain both characters which can be stored as a single byte as well as characters which are stored with two (or more) bytes. The shared ASCII range is contained as a range of single-byte characters.

There is a 1:1 assignment between Windows code page and language family. Section 1.3 (Language family) contains a list of the language families in Table 1-1.

### Diacritical characters

(diacritical = distinguishing)

In most cases, a relatively small supplementary character, which is attached to a letter to give the letter a specific stress, a specific pronunciation or even a new meaning.

Section 3.1 (Special characters in different language families) contains a list of diacritical characters.

If the combination of basic letters and diacritical characters belongs in a language-specific alphabet, this combination is contained in the appropriate Windows character sets (code page, Unicode) as a separate character.

## Use of diacritical characters

A diacritical character is used

- to designate the combined character (e.g. ð = G cedilla, Ö = O tilde),
- for the combined input if there is no separate key that possesses the required combined character (see Section 3.4: Using a different keyboard layout),
- in Vietnamese and Thai for normal coding in text files,
- for output (only in typographically compound texts): Diacritical characters and basic letter are output separately, whereby any combinations are possible, e.g. for ancient languages and phonetics.

In all the other cases, the combined letter is always considered as a compound unit.

In some cases, however, the combined letter is incorrectly referred to as the diacritical character.

## Input Method Editor (IME)

An aid for entering the CJK characters (CJK: Abbreviation for Chinese, Japanese and Korean).

## Language family

A language family is a group of languages which uses the same code page. This requires the use of the same basic alphabet (e.g. Latin or Cyrillic), but it does not mean that the alphabet of all languages of this group is the same.

There is a 1:1 assignment between Windows code page and language family. Section 1.3 (Language family) contains a list of the language families in Table 1-1.

## Font files (TrueType fonts)

Windows 2000 and Windows XP do not use font files stored separated by the code page, but instead so-called "Big Fonts"; these are fonts that contain the character set for several code pages, saving memory capacity for the common characters. With Word 2000/2003, the selection of characters from this total character set is performed via the Unicode coding, not via the code page coding. This makes it impossible to mix up characters with the same code in different code pages.

## Default language

The term "default language" is used in this guide for the language of the operating system, i.e. German for a German Windows environment and English for an English Windows environment.



## Default keyboard setting

### Windows 2000:

Under "Control Panel" → "Keyboard", one of the settings in the "Input" tab is defined as the default keyboard setting. This setting is indicated with a check mark on the left.

### Windows XP:

Under "Control Panel" → "Regional and Language Options", in the tab "Languages", click under the heading "Text Services and Input languages" on the button "Details..." to reach the dialog "Text Services and Input languages". Here you can see the default keyboard setting in the "Settings" tab under the heading "Default Language and Region Support".

### Windows 2000 and Windows XP:

The default keyboard setting consists of a language property (left) and a keyboard layout (right).

This language property of the default keyboard setting need not absolutely match the default language.

## Text file

The term "Text file" is used to describe a file which does not contain any formatting (font type, font size, underscore, bold, italics etc.) or embedded objects (pictures, tables, graphics, footnotes etc.).

This means that, apart from the text characters, the file will only contain ASCII control characters like Tab, CR and LF (tab indent, carriage return and line feed).

The text characters are taken from a particular code page.

Occasionally, text files are also incorrectly referred to as ASCII files.

This is only correct if a text file only contains ASCII characters (codes 0 to 127) and is therefore code page invariant.

## Unicode

A 16-bit character set for which code pages are no longer required.

All character codes are unique, without needing to specify the code page.

Unicode programs like Word 2000/2003 do not offer code page-dependent interpreting of characters and thus also cannot display characters differently by assigning a font or language property.





# 2

## 2 Editing Files with Word 2000/2003

2.1 Overview .....	TX2/2-14
2.2 Create a new text file .....	TX2/2-14
2.3 Open an existing file .....	TX2/2-15
2.4 Save text files.....	TX2/2-17
2.5 Codes for SINUMERIK HMI.....	TX2/2-19

## 2.1 Overview

This section describes how text files can be edited using Microsoft Word 2000/2003.

Word 2000/2003 does not edit text files directly. Instead, it converts them to Word files upon opening. During saving Word can then convert the edited text back into a text file.	We recommend saving an additional version of the text file as a Word file and using this version as the basis for future editing. It is then no longer necessary to open and convert the file.
--	--

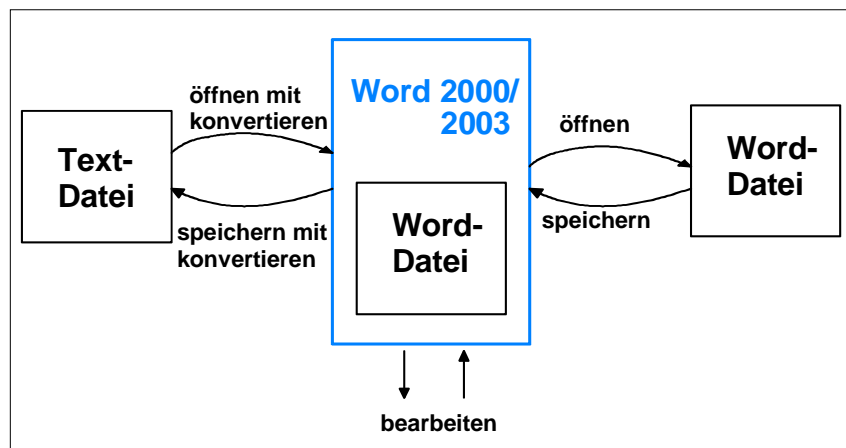


Fig. 2-1

## 2.2 Create a new text file

Create a new text file in the usual way, using one of the following methods:

- In Windows Explorer (Windows 2000) click on the command "File" → "New" → "Text File".
- In Windows Explorer (Windows XP) click on the command "File" → "New" → "Text Document".
- Launch Notepad without a file. A new empty file is then opened.  
Windows 2000: Then select the command "File" → "Save".  
Windows XP: Then select the command "File" → "Save As" → "Encoding: ANSI"
- Make a copy of an existing text file you wish to use as a template for the new text file.

Now open the text file with Word 2000/2003 in the way described in the next section.

## 2.3 Open an existing file

1. In Word 2000/2003 select "Open..." from the "File" menu.
2. In the "Open" dialog box, choose the setting "Encoded Text Files (\*.txt)" (Word 2000) or "Text Files (\*.txt)" (Word 2003) in the "Files of type" line.  
If your file does not end in ".txt" then you will need to rename it first so that it does end in ".txt".

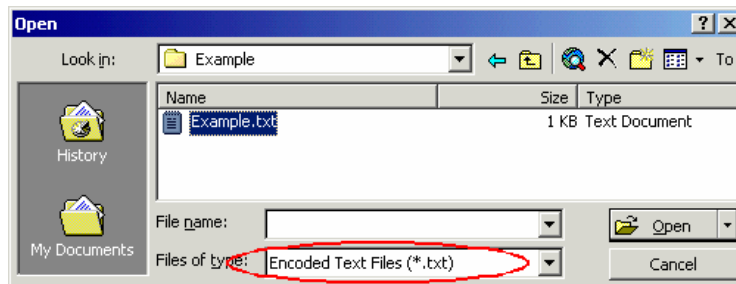


Fig. 2-2 Word 2000

3. Choose the file you would like to open and click on "Open".

Instead of steps 1-3, you can also use one of the standard shortcuts:

- In Word you can use the list of the recently used files.
- Drag and drop the text file from Explorer to the Word icon on your desktop.
- If Word has already been launched, drag and drop the text file from Explorer to the title bar of the Word window.

4. Note what happens next.

Continue with "Convert File" (see point 5.) if this dialog box opens.

Continue with "File Conversion" (see point 6.) if this dialog box opens.

If however the file is opened straight away with neither the "Convert File" nor the "File Conversion" dialog boxes opening up, then Word has probably opened the file incorrectly under the assumption that the text has been written using the default language.

Be careful with special characters. The special characters will appear incorrectly if the code page of the text file is not the same as the code page of the default language.

In this case, go to "Extras" → "Options..." and open the Options dialog for Word. Click on "General" and select the option "Confirm conversion at Open".

When this option is active the additional dialog "Convert File" (see point 5.) will appear every time, but when the option is inactive it is skipped.

Afterwards close the file and start again from point 1.

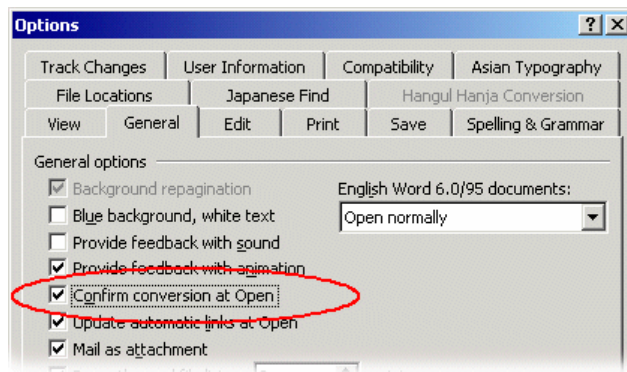


Fig. 2-3

5. The "Convert File" dialog box will appear. Choose "Encoded Text File" again and click on "OK".

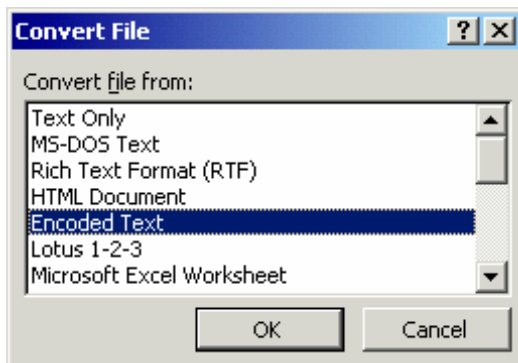


Fig. 2-4

6. A dialog box appears (Word 2000): "File Conversion - Example.txt". If "Text Only" is selected, activate "Other encoding".

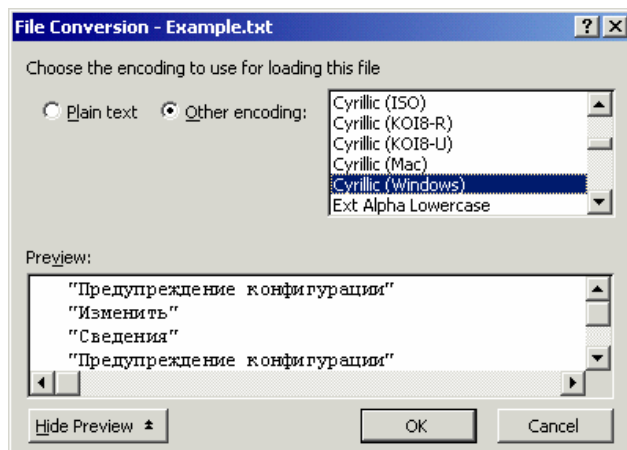


Fig. 2-5

- (Word 2003): "File Conversion - Example.txt". If "Windows (default)" is selected, activate "Other Encoding".

Select the correct encoding (Word will make a suggestion which will not necessarily be correct).  
Section 2.5 (Codes for SINUMERIK HMI) lists which encoding is suitable in Table 2-1.  
Check the preview area while doing this.

- Click on "OK".  
The file is opened and converted to a Word file at the same time.  
The existing text is formatted using a default font.

## 2.4 Save text files

Save the entered text as a "Text File".

- In Word 2000/2003, select "Save As..." from the "File" menu.

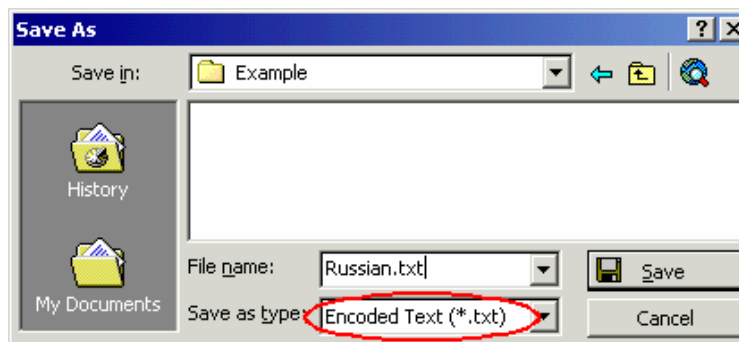


Fig. 2-6

- In the "Files of type" line, choose the setting "Encoded Text File (\*.txt)" (Word 2000) or "Text Only (\*.txt)" (Word 2003).
- Enter the name of the file in the "File name" box and click on "Save".
- A warning dialog appears (Word 2000):

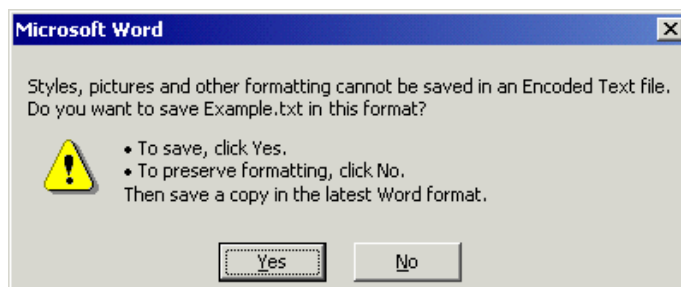


Fig. 2-7

This warning dialog does not appear in Word 2003, or is included in the dialog below (see Fig. 2-9).

This warning tells you that saving the file as a text file may result in formatting information and graphics objects being lost.

This message will even appear if there are no special formatting or graphics objects in the text.  
Click "Yes" to confirm.

5. The "File Conversion" dialog window will then appear.

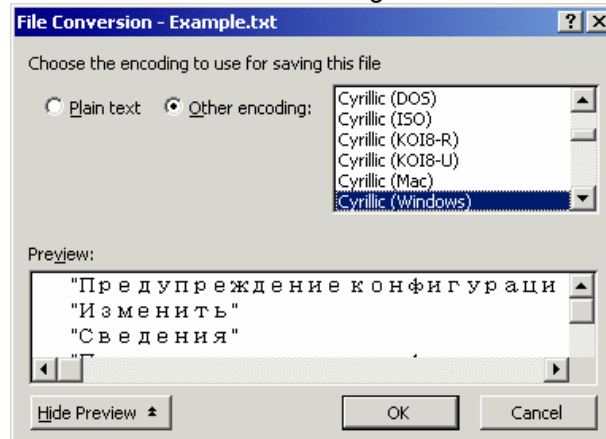


Fig. 2-8

Select the encoding which is suitable for your target language and click on "OK".

Section 2.5 "Codes for SINUMERIK HMI", Table 2-1 lists the suitable encoding options.

Your file is now saved as a text file and encoded using the selected code page in the process.

If you have selected a completely unsuitable code then a warning is displayed in the dialog box.

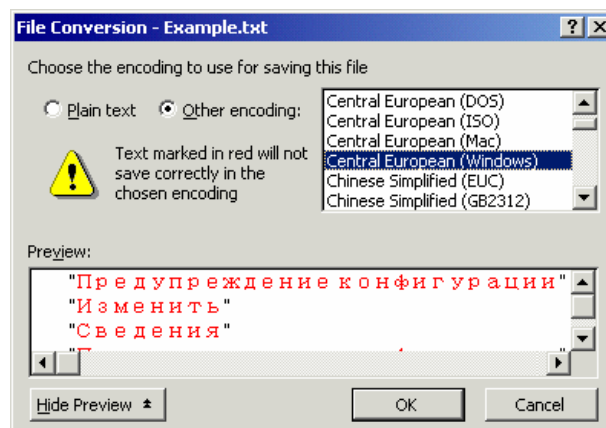


Fig. 2-9

This warning will also appear if you have selected the correct code but your file contains characters which are not permitted, such as Latin mutated vowels in a Cyrillic text.

If you choose to ignore the warning then all characters which cannot be encoded are replaced with a question mark.



## 2.5 Codes for SINUMERIK HMI

When you open or save a file, code page names are used rather than code page numbers, see Section:

Section 2.3: Open an existing file, or.

Section 2.4 Save text files

Depending on the installation of your system, many other codes may be listed, including those for Unix (ISO), Apple (Mac), OEM (DOS) and mainframe (EBCDIC) platforms.

Please use the Windows (ANSI) codes for SINUMERIK HMI.

Use the following codes to open and save files (shown in alphabetical sequence in accordance with the Word listing):

Table 2-1 Codes

<b>Name</b>	<b>Code page</b>
Traditional Chinese (Big5)	950
Simplified Chinese (GB2312)	936
Japanese (Shift-JIS)	932
Korean	949
Cyrillic (Windows)	1251
Central European (Windows)	1250
Turkish (Windows)	1254
Western European (Windows)	1252





# 3

## 3 Input of Text

3.1 Special characters of different language families .....	TX2/3-22
3.2 Entering characters with the character map .....	TX2/3-23
3.3 Compound input.....	TX2/3-25
3.4 Using a different keyboard layout .....	TX2/3-26
3.4.1 Concept of the keyboard language property .....	TX2/3-26
3.4.2 Setting up a keyboard layout under Windows 2000 .....	TX2/3-27
3.4.3 Setting up a keyboard layout under Windows XP .....	TX2/3-28
3.4.4 Use.....	TX2/3-30
3.5 Numerical text input.....	TX2/3-32
3.6 Language properties in Word files .....	TX2/3-33
3.7 Multi-language Word files .....	TX2/3-36
3.8 Special considerations when working with Greek and Cyrillic texts .....	TX2/3-39
3.9 Special considerations when working with East Asian texts ....	TX2/3-40

## 3.1 Special characters of different language families

### Latin language family

With the Latin-based language families (Baltic, Central European, Turkish, Western), you can enter most of the characters, namely the basic Latin alphabet, directly with your (e.g. Western) keyboard.

Most of the special characters, such as Ā ā â ã ç đ é è Í ï Ľ Ń ñ ö ó ő ŧ Ũ ů Ū are made up of the Latin letters A-Z a-z and diacritical characters such as acute ´,

breve ˘, cedilla ¸, circumflex (caret) ^, point ˙, colon (dieresis, trema) ¨, double-acute ˝, grave accent ` , háček (caron) ˇ, macron ¯, ogonek (nasal hook, crooked hook) ˛, squiggle °, slash /, hyphen – or tilde ~.

Other characters of the alphabet are special letters (Ð þ, Æ œ, Æ æ, Œ œ, Œ œ) or ligatures (Æ æ, Œ œ, Œ œ). These are supplemented by language-specific abbreviations and punctuation marks, e.g. for the endings of the ordinal numbers in Spanish (ª and º) as well as the exclamation and interrogation marks put upside down (¡ and ¿).

The input methods described in the following will apply without restrictions to the Latin-based language families.

### Greek/Cyrillic language family

Certain special considerations need to be taken into account when inputting texts using Greek or Cyrillic fonts.

Please refer to Section 3.8: Special considerations when working with Greek and Cyrillic texts.

### East Asian languages

Certain special considerations also need to be taken into account when creating texts in Japanese, Chinese or Korean.

Please refer to Section 3.9: Special requirements for East Asian texts.

### Disruptive Word options

Deactivate the following options in Word to prevent unwanted characters from reaching the text:

In the dialog "Extras" → "AutoCorrect Options" go to the "AutoFormat as you type" tab under

"Replace as you type":

- "Straight quotes" with "Smart quotes"
- English ordinals (1st) with superscript
- Fractions (1/2) with fraction character (½)
- Special characters (--) with symbols (—)

"Apply as you type":

- Automatic bulleted lists
- Automatic numbered lists

If you do leave any of the options active then you can always use Ctrl+Z to undo any automatic corrections. As a general rule this will give the same results as if the option was deactivated.

## 3.2 Entering characters with the character map

### Note

The character map integrated in Word (menu commands "Insert" → "Symbol...") is not suitable for creating code page-encoded text files, as it cannot be limited to the relevant code page.

Use the "Character Map" system program as described below.

### Selecting the character map

1. On the start menu, choose "Programs" → "Accessories" → "System Tools" → "Character Map" to start the "Character Map" tool.
2. From "Font", choose the same font type as you have selected for the text file.

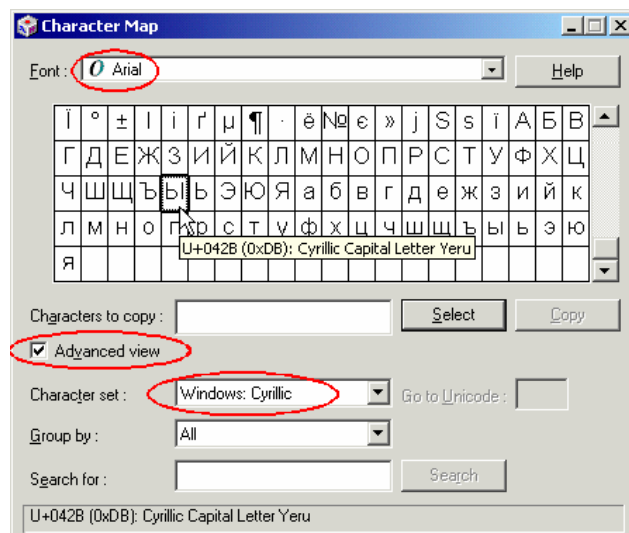


Fig. 3-1

3. Activate the box "Advanced view".

4. Under "Character set" select the code page for your text file in order to restrict the number of available characters to this code page.  
The code pages are identified as follows:

Table 3-1 Encoding of text files

Name	Code page
Windows: Chinese (Taiwan)	950
Windows: Chinese (PROC)	936
Windows: Japanese	932
Windows: Korean	949
Windows: Cyrillic	1251
Windows: Eastern European	1250
Windows: Turkish	1254
Windows: Western	1252

### Entering characters

1. Choose a character you want to enter.  
The information bar at the bottom will display the Unicode code (U+..., hexadecimal), the code page code (in brackets) and the clear text description of the character.  
Once you have selected the appropriate keyboard layout (see Section 3.4: Use a different keyboard layout), the key combination for the numerical input is displayed in the status line at the bottom right (see Section 3.5: Numerical text input).



Fig. 3-2

2. Click on "Select".  
The character is copied into the "Characters to copy" field.
3. Repeat steps 1 and 2 for further characters.
4. After you have selected all the characters, click on "Copy".  
The characters are copied from the "Characters to copy" field to the clipboard.
5. Go back to Word 2000/2003 and press Ctrl-V.  
The characters are pasted from the Clipboard into the text.

### 3.3 Compound input

With many of the compound special characters it is possible to enter the diacritical character first and then the letter. The diacritical character will not appear at first and is then combined with the letter to form a special character.

#### Example

With a German keyboard layout, press for example the keys ´ (acute) and "a" one after the other to obtain the character á (a-acute).

The keys with the diacritical characters, which are thus dealt with in a special way, are marked with a special color in the descriptions of the keyboard layouts.

#### Keyboard assignments

The table below shows some keyboard layouts for the supported diacritical characters:

Table 3-2

Keyboard assignments	Acute	Grave accent	Circumflex	Háček	Breve	Ogonek	Point	Colon	Cedilla	Double-acute	Squiggle	Tilde
	´	¸	ˆ	ˇ	˘	¸	·	:	¸	˝	°	˜
German	X	X	X									
French		X	X					X				X
Polish	X		X	X	X	X	X	X	X	X	X	
Spanish	X	X	X					X				X
Czech	X		X	X	X	X	X	X	X	X	X	
Hungarian	X		X	X	X	X		X	X	X	X	
U.S International	X	X	X					X				X

Obtain a graphical representation of your keyboard layout to learn which key combinations produce the diacritical characters.

#### Keyboard layout in picture form

Images of the keyboard layouts can be obtained from Microsoft via the Internet at the following address:

<http://www.microsoft.com/globaldev/reference/keyboards.msp>

If you press the space bar after the diacritical character, you will see the character itself.

---

**Note**

It can be useful to switch the keyboard layout to "U.S. International" which corresponds mainly to the American keyboard, but additionally contains 5 diacritical characters and provides many special characters.

---

## 3.4 Using a different keyboard layout

### Necessity for changing the keyboard layout

When is a change in the keyboard layout necessary?

- If you have to enter a lot of text.
- Your translator, who is used to a certain language-specific keyboard, wants to use your PC.

However, if you (as a German) only need to make smaller changes to existing files, then the German keyboard layout may be enough provided there are only a few special characters which cannot be represented with the German keyboard. You can enter these characters using the Character Map.

### Example

In Hungarian for example, you would have to enter these characters as follows:

- őŰűÚ (O double-acute, U double acute) via the character map
- öŬűÚ directly
- áÁéÉíÍóÓúÚ compound input

### 3.4.1 Concept of the keyboard language property

Windows manages pairs for the keyboard consisting of language ("Language and Region Support") and keyboard layout.

The set keyboard language determines which language property is assigned to the entered characters.

The associated keyboard layout will determine which key produces which character.

Several such settings can be set up and switched between. One of the settings is defined as the default setting.

The keyboard driver assigns the entered key characters a "language property".

This defines that this piece of text has been created in this language.

Windows uses the term "Language and Region Support" as this can be used not only to switch the language properties, but also to select settings such as currency, date format etc.



### 3.4.2 Setting up a keyboard layout under Windows 2000

Proceed as follows to select a new keyboard layout:

1. From the Start menu, select "Settings" → "Control Panel" and click on "Keyboard".
2. Select the "Input Locales" tab.

If a symbol like **DE** is displayed in the right-hand task bar then you can also right-click on this symbol and select the command "Properties" to speed up steps 1 and 2.

You will see here at least one assignment "Input language" (left-hand column) and "Keyboard layout" (right-hand column) already set up. One entry is defined by default: Your default language (the language of your operating system) and your default keyboard layout. This entry is selected and indicated with a check mark on the left.

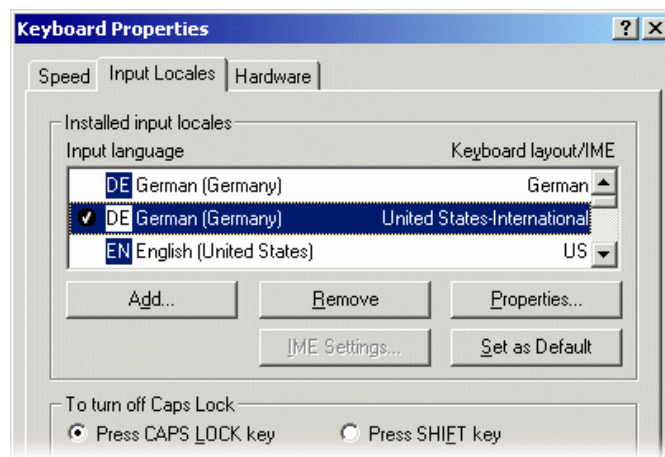


Fig. 3-3

3. Click on "Add".  
The dialog box "Add Language and Region Support" is opened.

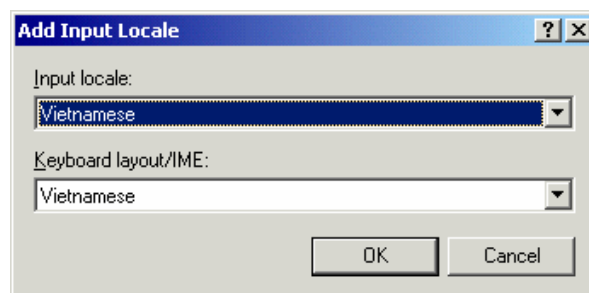


Fig. 3-4

4. In the dialog "Add Language and Region Support", select a language under "Language and Region Support" and the associated keyboard layout under "Keyboard Layout".  
The language ("Language and Region Support") may also be the same as the default language or as some other language for which you have already set up a keyboard layout. However, in this case you will not be able to make such a good distinction between the two keyboard layouts, as Windows usually only displays the language ("Language and Region Support").
5. Click on "OK". The dialog box "Add Language and Region Support" is closed. You then return to the "Keyboard Properties" window.
6. Use "Hot keys for input locales" to select the way how you want to switch the keyboard layout. If you select "none", you can only switch using the mouse.



Fig. 3-5

7. Activate the option "Enable indicator on task bar". This provides you with the possibility of switching over quickly using the mouse.
8. Click on "OK" to quit the dialog.

### 3.4.3 Setting up a keyboard layout under Windows XP

Proceed as follows to select a new keyboard layout:

1. From the Start menu, select "Settings" → "Control Panel" and click on "Regional and Language Options".
2. Click on the "Languages" tab and click on the button "Details..." under the heading "Text Services and Input Languages".

If a symbol like **DE** is displayed in the right-hand task bar then you can also right-click on this symbol and select the command "Properties..." to speed up steps 1 and 2.

3. The dialog "Text Services and Input Languages" is opened. Select the "Settings" tab.

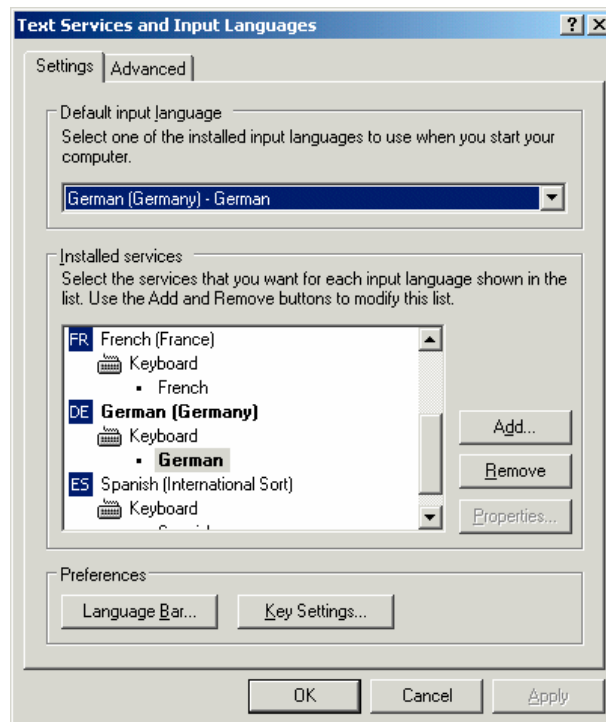


Fig. 3-6

4. Click on "Add".  
The dialog box "Add Language and Region Support" is opened.

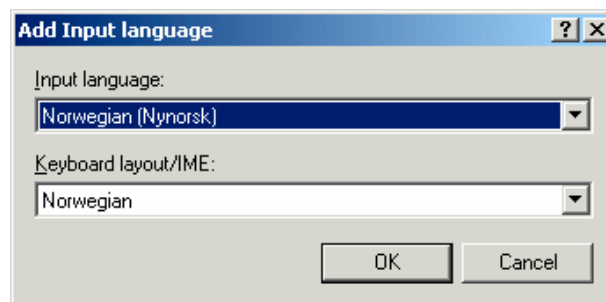


Fig. 3-7

5. In the dialog "Add Language and Region Support", select a language under "Language and Region Support" and the associated keyboard layout under "Keyboard Layout".  
The language ("Language and Region Support") may also be the same as the default language or as some other language for which you have already set up a keyboard layout. However, in this case you will not be able to make such a good distinction between the two keyboard layouts, as Windows usually only displays the language ("Language and Region Support").

6. Click on "OK". The dialog box "Add Language and Region Support" is closed. You then return to the "Text Services and Input Languages" window.
7. Use "Keyboard..." to select the way how you want to switch the keyboard layout. If you select "none", you can only switch using the mouse.
8. Under the "Language and Region Support" bar you can select whether and how the current keyboard layout is displayed in the task bar or in a special button bar.
9. Click on "OK" to quit the dialog.

### 3.4.4 Use

You can now create text with a "keyboard layout" suitable for the target language (for the input of your texts in a foreign language) or with your previous default keyboard layout (for the remaining operation). You can switch between the two layouts as required.

Now, a small blue field with a two-letter language abbreviation should appear in the task bar for the language you have set:

If you position the mouse pointer on this field (without clicking on it), the setting is displayed .

If the selected keyboard layout is the default layout for the selected language (e.g. "Turkish" – "Turkish Q"), then only the language ("Turkish") is displayed, otherwise the complete settings with language and keyboard layout will be displayed ("Turkish – Turkish F").



Fig. 3-8

If you click with the left mouse button on this field, the languages you have set will be offered in a small menu.

The current setting is marked by an arrow (Windows XP does this with a check).



Fig. 3-9

If you click with the right mouse button on this field, an abbreviation of the dialog box "Keyboard Properties" is displayed above the "Properties" menu option.

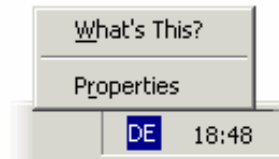


Fig. 3-10

Further settings are offered under Windows XP. Among other things you can also arrange this display as an Input language bar independently of the task bar, for example at the top edge of the screen.

### Keyboard layout specific to input window

The Windows operating system will memorize the current keyboard layout for each input window.

- The keyboard layout is switched automatically if you switch to another window.  
A new window is always started with the default setting. After starting Word, you may have to reselect the keyboard layout.
- Enable the window in which you want to enter texts before you select the keyboard settings for this window.

### Menu operation

All menus in the active program will also use the changed keyboard layout. You will possibly no longer find certain key combinations with Alt and the letter key or Ctrl with a letter key or even initiate a wrong command by mistake. You should therefore use the mouse or, in menus, the arrow keys.

---

#### Additional notes

If you do not want to switch back to the other keyboard layout quite as frequently, you can also declare the new combination with the special keyboard layout as the default target language by selecting Start menu "Settings" → "Control Panel" → "Regional and Language Options" → tab "Languages", dialog box "Keyboard Properties" (Windows 2000) or "Text Services and Input Languages" (Windows XP):

#### Windows 2000:

Select the new entry and click on the button "Set as default".  
The default setting is indicated with a check mark on the left.

#### Windows XP:

Select the new setting under "Default Language and Region Support".

Please note that all new windows, the Start menu and the desktop will now also use the new keyboard layout.

---

## Keyboard layout in picture form

You can obtain the keyboard layout in picture form.  
For appropriate Internet addresses, see Section 5.1. References for text creation

## Checking the keyboard layout

Compare the keyboard layouts and note a key with two different assignments. If you are not sure, press this key when entering the text to test which layout is active.

Example: If the German keyboard layout is active, pressing key "1" (letter group) will produce a "1", and if the Czech keyboard layout is active, a "+".

---

### Note

Check whether the automatic option for switching over the keyboard has been activated in Word. Inadvertent switching of the keyboard layout by Word can cause confusion.

Section 3.7 (Multi-language Word files) describes situations in which the automatic keyboard switchover function is useful and how to activate and deactivate it.

---

## 3.5 Numerical text input

You can use numerical inputs with Word 2000/ 2003 only if you have selected a suitable keyboard layout (see Section 3.4: Using a different keyboard layout).

Press the ALT key, and while holding down the key, enter zero and then the three-digit decimal coding of the desired special character on the numerical key group. Only then release the ALT key.

Example (Turkish):

ALT+0222 results in Ş (S with cedilla).

To produce this character, you will only need the character map of the appropriate code page (see Section 1.3: Language family).

You will find overview tables of the code pages e.g. at Microsoft under <http://www.microsoft.com/globaldev/reference/WinCP.mspx>

From these tables, combine the headers for the column and row (hexadecimal 00 to FF) and convert this number to form a decimal number (0 to 255).

The 4-digit hexadecimal numbers contained in the individual table fields are the corresponding Unicode codes which can be used for identification of a character, but they cannot be used for numerical input of that character.

Section 3.2 (Input via the Character Map) also describes where the key combination is displayed in the Character Map.

## 3.6 Language properties in Word files

### Language property

In Word, a property (attribute) is managed separately for each character similarly to other text properties (bold, underscore).

In the language properties, Word will remember the language to which the character, the character sequence, the word or the whole sentence belongs.

The language property is independent of the font. For example, a Cyrillic character can possess the language property "German".

The language property is managed internally as a combination of main language and sublanguage, e.g. "French (Canada)".

Microsoft sometimes uses the designation "Region scheme" for the language property, whereby this term also covers properties, such as currency, date format, decimal separators etc.

### Effects of the language property

This property is used (evaluated) for the spell checker and for the automatic keyboard switchover (see Section 3.7: Multi-language Word files).

If you do not wish to use either the spell checker or the automatic keyboard switchover function, there is generally no need to worry about the language properties. In this case do not forget to deactivate the automatic keyboard switchover function.

### Saving the language property

Microsoft Word saves this property in Word files (\*.doc). If a document is saved as a text file (\*.txt) then this property is lost along with the other text properties.

If you wish to use the language property, it therefore makes sense to use a Word file to edit the texts and to keep it for subsequent changes. Whenever the editing reaches a stage where you want to stop you can then save the document as a text file as described in Section 2.4: Saving text files.

## Displaying the language property

In Word 2000, the current language property is displayed in the status bar at the bottom edge of the window.



Fig. 3-11

## Setting the language property

Proceed as follows to set the language property for a text:

1. Highlight the part of the text for which you wish to set the language property, or press CTRL+A to select the entire text.
2. Then select the command "Extras" → "Language" →  
with Word 2000: "Set Language..."  
with Word 2003: "Set Language..."
3. In the "Language" dialog box you can choose the correct language. Click on "OK" when you are done.

## Quick setting of the language property

In Microsoft Word 2000/2003 you can insert a combined display and selection box into a symbol bar. This both displays the current language property and enables you to set the language as well.

Proceed as follows to insert this field into a symbol bar:

1. In the "Extras" menu select the command "Customize..."

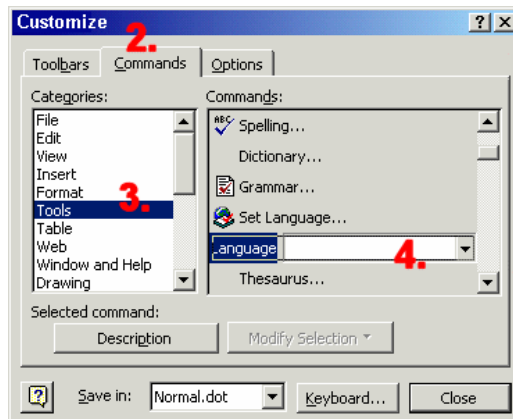


Fig. 3-12 Word 2000

2. In the "Customize" dialog box select the "Commands" tab.
3. Select "Tools" from the list of categories on the left.
4. Then select "Language" from the list of commands on the right (see Fig.).



5. Drag and drop this entry with the left mouse button into a symbol bar.

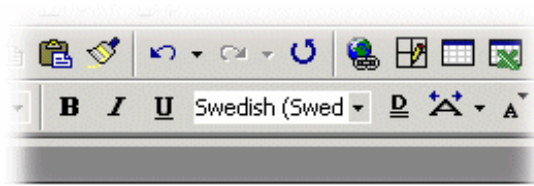


Fig. 3-13

6. Close the dialog box.

With this selection box you can then change the selected language for any part of the text.

### Automatic setting of the language property

Word 2000/2003 automatically sets the language property in the following situations:

- When opening a file of type "Encoded Text File" (see Section 2.3: Opening an existing text file).
- New text entered with the keyboard is automatically assigned the selected keyboard language.
- If the automatic recognition option is selected then Word will try to assign the entered words to a language once they are complete and then automatically sets the language property.

---

#### Note

The "Detect language automatically" can often deliver confusing results, particularly in technical texts.  
This option is usually pre-activated as a default setting.

**Recommendation:** Deactivate the "Detect language automatically" option.

---

Proceed as follows to deactivate the automatic language recognition option:

1. Make sure that no part of the text has been highlighted and select the command "Tools" → "Language" →  
with Word 2000: "Set Language..."  
with Word 2003: "Set Language..."
2. In the "Language" dialog box deactivate the option "Detect language automatically" and click on "OK".

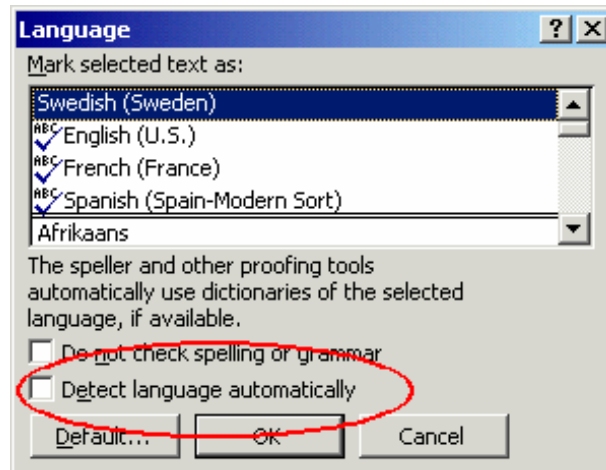


Fig. 3-14

### 3.7 Multi-language Word files

In certain cases it may be useful to deliberately set different language properties for different parts of the text in a Word file.

#### Boundary conditions for multi-language Word files

You want to enter certain parts of the text with the German keyboard layout and then also edit these parts later on with the German keyboard layout, whereas other parts of the text are to be created and edited with the keyboard layout of the target language.

#### Examples

This can be necessary in the following cases:

- With Greek and Cyrillic texts, the Latin letters and other ASCII characters are missing in the appropriate keyboard layouts.
- With some keyboard layouts, e.g. Czech, no access to the digits is granted on the standard keyboard level.
- The arrangement of the Latin letters deviates substantially from the arrangement to which you are accustomed to (e.g. Turkish-F keyboard layout).

- You as a German want to prepare one part of the text, and a second part is to be created by your translator.

### Table with two language properties

To create a multi-column text in which for example the first column possesses the language property "German" and the second column for example "Russian", you can proceed as follows:

- Switch to the German keyboard and enter the first column of the first line. Then switch to the Russian keyboard and enter the second column:  
[Text 103][Mode 28] "йцукенгшщзхъфывапроджэячсмить"
- This does not need to be a Word table with borders and fields. You can also use a simple line and separate the different parts with spaces.
- Select the whole line, copy this to the clipboard by using Ctrl-C and paste this line several times by pressing Ctrl-V. This will give you two columns with German on the left and Russian on the right.

Table 3-3

[Text 103][Mode 28]	"йцукенгшщзхъфывапроджэячсмить "
[Text 103][Mode 28]	"йцукенгшщзхъфывапроджэячсмить "
[Text 103][Mode 28]	"йцукенгшщзхъфывапроджэячсмить "
[Text 103][Mode 28]	"йцукенгшщзхъфывапроджэячсмить "
[Text 103][Mode 28]	"йцукенгшщзхъфывапроджэячсмить "
[Text 103][Mode 28]	"йцукенгшщзхъфывапроджэячсмить "

German Russian

### Automatic keyboard switching

In order to automatically have the correct keyboard layout when working on different parts of the text in Word, go to "Extras" → "Options" and activate the option "Any-Keyboard switching" in the "Edit" tab.

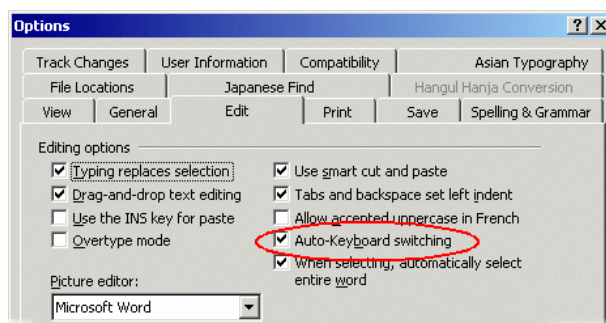


Fig. 3-15

---

### Note

Accidental use of the automatic keyboard switching function can be confusing. Only use this option when the above boundary conditions are met.

---

### Active keyboard layout always visible

To make sure you know what is going on, you can adjust the task bar to always display the currently active keyboard layout.

1. Right-click on the task bar and select "Properties".
2. Activate the option  
in Windows 2000: "Always on Top"  
in Windows XP: "Key the taskbar on top of other windows"
3. Deactivate the option  
in Windows 2000: "Auto hide"  
in Windows XP: "Auto-hide the taskbar"

With Windows XP you are also free to position the Language and Region Support bar anywhere you want on the screen. To do this, click on the button for switching over the keyboard (e.g. **DE**) in the task bar and select "Restore Language and Region Support Bar".

Another option is to activate the switchover of the keyboard in Word:

1. In the "Extras" menu select the command "Customize..."

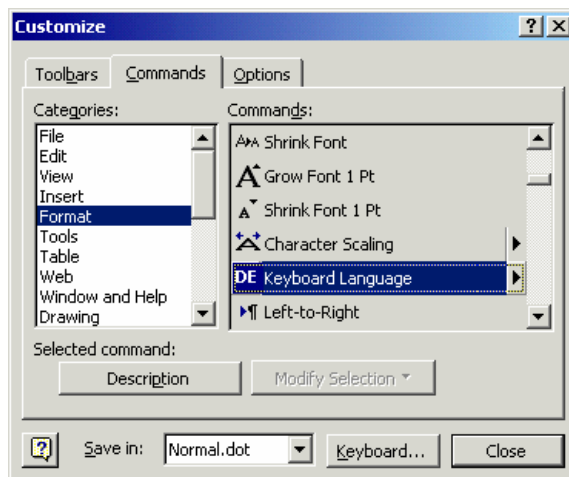


Fig. 3-16

2. In the "Customize" dialog box select the "Commands" tab.
3. Select "Format" from the list of categories on the left.
4. Then select "DE Keyboard Language" from the list of commands on the right (see Fig.).
6. Drag and drop this entry with the left mouse button into a symbol bar.

## 3.8 Special considerations when working with Greek and Cyrillic texts

---

### Note

If you create texts in the Greek or Cyrillic language for the first time, then you **MUST** read this section!

---

From the view of the language sciences, the Greek and Cyrillic alphabets are considered as separate and independent alphabets, and not for example as an extension of the Latin alphabet.

This means that Greek and Cyrillic letters which in their appearance are identical to Latin letters, are not the same characters and may therefore also not be coded identically.

In some cases this becomes evident by the fact that although certain letters look the same in upper case, the corresponding lower case letters look different.

Table 3-4

Font	Name	Meaning
Latin	P, p	Latin letter P
Greek	Ρ, ρ	Greek letter RHO
Cyrillic	Р, р	Cyrillic letter ER
Latin	H, h	Latin letter H
Greek	Η, η	Greek letter ETA
Cyrillic	Н, н	Cyrillic letter EN

### Example

Incorrect coding could have the following consequences:

Supposed you write the Russian word for "NO" ("HET") using Latin letters instead of "HET" using Cyrillic letters. Initially you will see no difference. However, the search for "нет" in lower case letters with the search option "Ignore uppercase/lowercase letters" will not find the incorrectly coded spelling.

### Keyboard layout for Greek/Cyrillic font

For this reason, the keyboard layouts for the Greek and Cyrillic fonts have been designed in such a way that the Latin letters are not available at all, not even via AltGr key combinations. These keyboard layouts have normally only two levels (standard and SHIFT).

## Keyboard layout for the Russian font

In the Russian keyboard layout, there are also some ASCII special characters which are missing, such as # \$ [ ] { } ' ` | ~ & .  
For these languages it therefore makes sense to use Word files with two languages as described in Section 3.7 Multi-language Word files.

---

### Note

Switch the keyboard to Latin only if this is absolutely necessary! When doing so, do not switch over for individual letters, but only for complete words and sentences.

Never use words with mixed codings!

---

### Note

Find out whether and to what extent certain German, English or international abbreviations (e.g. DIN/ISO) may or must be written using Latin fonts, or whether a conversion (ДИН/ИСО) is more appropriate according to the conventions of the country in question.

The key assignment for combinations, such as Ctrl-C, Ctrl-V, Ctrl-Z, Alt-A, etc. normally follow the US keyboard layout.

---

## 3.9 Special considerations when working with East Asian texts

### Requirement

To author or correct texts in Japanese, Chinese or Korean, you will need to have installed the corresponding expansion package for the operating system.

### Checking the system installation for Windows 2000

To check your installation, proceed as follows:

- Open the Control Panel ("Start" → "Settings" → "Control Panel").
- In the "Control Panel", select "Regional Options".
- Click on the "General" tab in the "Regional Options" dialog box.
- You can see the installed language families in the lower part of the window under "Language settings for the system".
- Check whether the language family you want to use has been activated.

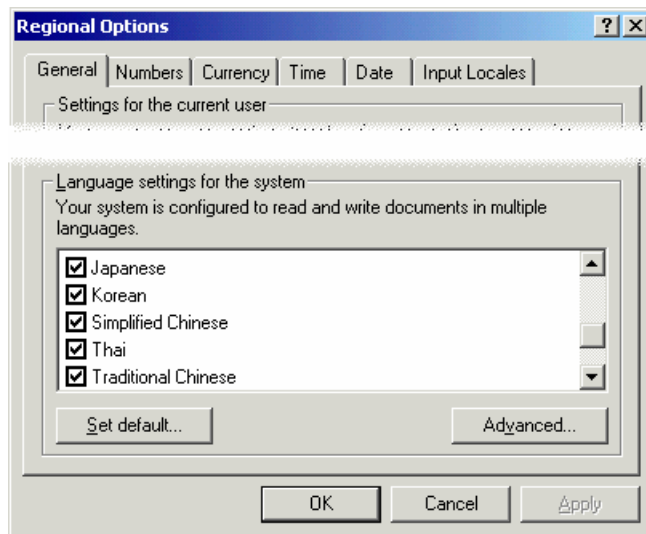


Fig. 3-17

You may require the Microsoft installation CD to install any missing components. You will also require Administrator Rights to perform the installation.

### Checking the system installation for Windows XP

In the Start menu select → Control Panel → "Regional and Language Options". Check the "Languages" tab to see whether the option "Install files for East Asian languages" has been selected.

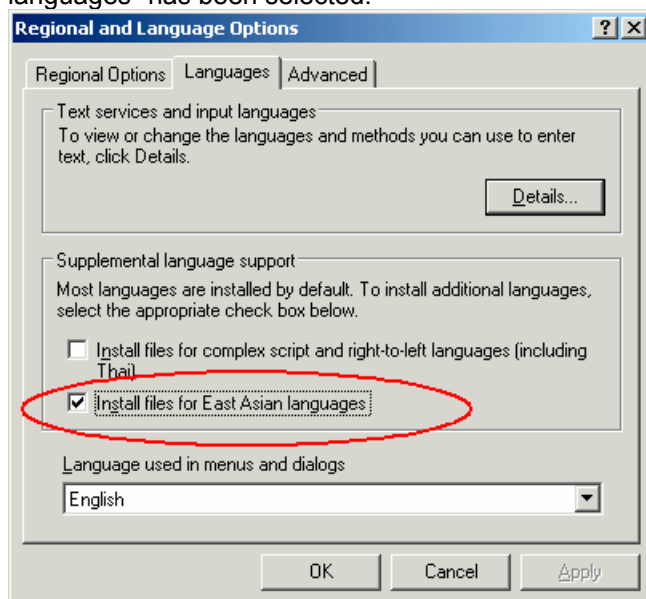


Fig. 3-18

You will not usually need the installation CD to reinstall the files for East Asian languages in Windows XP, as the files are already present on the hard drive in compressed form.

You will however require Administrator Rights to perform the installation.

## Input

An Input Method Editor (IME) is used for inputting. This is a tool which is used to put together the pictograms on a European keyboard and can be activated analogously to a keyboard layout.

You will normally need a translator who speaks the target language as his/her mother tongue to operate the IME system.

From Windows 2000 it is no longer necessary to have the Windows operating system in the target language.

## Full-width European characters

In the code pages for the East Asian languages, two versions of many of the European characters (Latin letters, numbers and the characters # \$ % & etc.) are included:

1. Normal (narrow, half-width) version.  
This is the set of ASCII characters which is compatible with European code pages.
2. Double-width (full-width) version.  
These versions of the characters are twice as wide and are not compatible with the ASCII characters.

If you are using characters with a functional significance it is important to use the ASCII versions.

### Correct example (SINUMERIK alarm text, Japanese):

014195 0 0 "DコードとG49が同時に指令されています (Ch%1 %2)"

Fig. 3-19

The circled characters are:

014195 0 0	Identifier and attributes
Space character	Syntactical separator
Quotes	Text delimiters
%1, %2	Parameter variables

Only the ASCII versions must be used for these characters.

---

### Note

In the remaining part of the text it is quite acceptable to use full-width characters for better readability (as in the example here for G49).

---



---

**Notice**

There is even a full-width version of the space character (ideographic space). You can only recognize this character from its width by selecting it.

---

**Incorrect example:**

014195 0 0 "DコードとG 4 9が同時に指令されています (Ch%1 %2)"

Fig. 3-20

The circled characters are full-width variants which have been used incorrectly.

---

**Note**

When placing a translation order, tell the translator that only the narrow, ASCII-compatible characters must be used for characters with a functional significance.

---

**Simplified/traditional Chinese**

---

**Note**

Please make a careful distinction between simplified Chinese and traditional (or standard) Chinese. The former is used in the PR of China, whereas the latter is used in Taiwan. These two variants are coded differently and therefore have to be treated differently. You should therefore also make sure that this is clearly specified in your translation order.

---

**Creating texts**

When you are creating texts in Japanese, Chinese or Korean, it definitely makes sense to initially create a Word file. There should be no problems if you are using Word 2000/2003 to exchange documents with your external translator. Afterwards you can then convert this Word file into a text file in the way described in Section 2.4 (Saving text files).

■



# 4

## 4 Problems and Remedies

4.1 Problems and remedies.....TX2/4-46

## 4.1 Problems and remedies

### Keyboard layout switches unintentionally

**Problem:**

The keyboard occasionally switches to a different layout while inputting text without this being explicitly requested.

**Remedy:**

Check the option for automatic switchover of the keyboard as described in Section 3.7: Multi-language Word files.  
Deactivate this option.

### Language properties are changed inadvertently

**Problem:**

After creating a text with the keyboard in a certain layout (e.g. Polish), you notice that parts of the text have German or English as the language.

**Remedy:**

Deactivate the automatic language recognition option as described in Section 3.6: Language properties in Word files

### Hotkeys no longer work

**Problem:**

Certain commands (Ctrl+C, Alt+T etc.) no longer work.

**Remedy:**

As a result of assigning a different keyboard layout, the keyboard shortcuts have changed for certain commands. Use the mouse while you are working with the other keyboard layout.

### Individual texts not working

**Problem:**

In some cases the finished text files do not work in the SINUMERIK Runtime system.  
Individual texts or lines of text are obviously not being read correctly.

**Remedy:**

Check the formal text criteria:

Are all of the characters which have a functional significance correctly encoded?

- If inverted commas are a requirement, check whether Word has converted the straight inverted commas (ASCII characters) that were entered into typographic quotation marks. Some disruptive Word options are listed in Section 3.1: Special characters of various language families.
- If the texts in question are East Asian texts, check the coding of the non-Asian characters, see Section 3.9: Special requirements for East Asian texts.  
If full-width characters have been used for functional characters, replace them with the corresponding ASCII variants.

## Yen character (Japanese) or Won character (Korean) in the text

### **Problem:**

The translator has obviously used a Yen character ¥ or a Won character ₩ wherever you would expect a backslash “\”.

### **Remedy:**

This is not necessarily a mistake.

Instead, this has historic reasons. For many years the Yen character has been used in Japan and the Won character in Korea in place of the backslash character.

This representation was kept once the PC users in those countries had got used to using this representation in path names (where you would otherwise expect the backslash character). However, internally this is still the backslash character.

In the fonts "MS Gothic" (Japanese) and "Batang" (Korean) the backslash symbol is represented as a currency character.

However, it is also possible that the Yen character (Unicode U+00A5) or Won character (Unicode U+20A9) was actually entered instead of the backslash character (Unicode U+005C).

To check this, reformat the text in Word using a different font, e.g. "Courier New".

If you see the backslash character in its normal representation then everything is in order.

However, if you can still see the currency symbol or a symbol which is not a valid symbol then you should correct the character and replace it with a backslash.

Afterwards revert back to the normal font (MS Gothic or Batang).





# 5

## 5 References

5.1 References for text creation.....TX2/5-50

## 5.1 References for text creation

For further information about creating texts, please refer to:

Book: *Developing International Software, Second Edition*  
Microsoft Press, October 2002, ISBN 0-7356-1583-7

Online version:

[http://www.microsoft.com/globaldev/getwr/dis\\_v2/default.aspx](http://www.microsoft.com/globaldev/getwr/dis_v2/default.aspx)

1. edition (*Developing International Software For Windows 95 and Windows NT*):

[http://www.microsoft.com/globaldev/dis\\_v1/disv1.asp](http://www.microsoft.com/globaldev/dis_v1/disv1.asp)

Overview tables of the Microsoft code pages

<http://www.microsoft.com/globaldev/reference/WinCP.aspx>

Pictures of the keyboard layouts in Microsoft can be found at the following address:

<http://www.microsoft.com/globaldev/reference/keyboards.aspx>

For printing:

<http://www.microsoft.com/globaldev/reference/win2k/setup/localsupport.aspx>

<http://www.microsoft.com/globaldev/DrIntl/faqs/winxp.aspx>

Microsoft Global Software Development

Detailed explanations on various aspects of international software

<http://www.microsoft.com/globaldev>

Unicode

Tables, definitions, standards and tools

<http://www.unicode.org>





# I Index

## I.1 Index

### A

Automatic keyboard switching..... TX2/3-37

### C

Character map

Code page ..... TX2/3-24

Entering ..... TX2/3-23

CJK characters ..... TX2/1-10

Code page

Name..... TX2/2-19

Number..... TX2/2-19

Code page ..... TX2/1-9

Cyrillic language family ..... TX2/3-39

### D

Default keyboard setting..... TX2/1-11

Diacritical characters

Overview ..... TX2/3-22

### E

East Asian texts ..... TX2/3-40

### F

File conversion ..... TX2/2-16

### G

Greek/Cyrillic language family ..... TX2/3-39

### I

IME (Input Method Editor)..... TX2/1-10

Input Method Editor (IME) ..... TX2/1-10

Installation überprüfen ..... TX2/1-8

### K

Keyboard layout always visible... TX2/3-38

Keyboard layout specific

to input window ..... TX2/3-31

### L

Language family

Overview..... TX2/1-7

Language family..... TX2/1-10

### M

Multi-language Word files ..... TX2/3-36

### Q

Quick setting of

the language property ..... TX2/3-34

### S

Setting the language property..... TX2/3-34

Special characters of different

language families ..... TX2/3-22

System installation

for Windows 2000..... TX2/3-40

for Windows XP ..... TX2/3-41

**T**

Table with two  
language properties ..... TX2/3-37  
Text file  
  Convert..... TX2/2-16  
  Create new..... TX2/2-14  
  Open..... TX2/2-15  
Text input  
  Numerical ..... TX2/3-32  
TrueType fonts ..... TX2/1-10

**U**

Unicode ..... TX2/1-11  
Use of diacritical characters ..... TX2/1-10

**W**

Windows-code page ..... TX2/1-7



To:  
Siemens AG

A&D MC MS  
Postfach 3180

91050 ERLANGEN, GERMANY

Phone: +49 (0) 180 5050 - 222 [Hotline]  
Fax: +49 (0) 9131 98 - 63315 [Documentation]  
E-mail: <mailto:motioncontrol.docu@siemens.com>

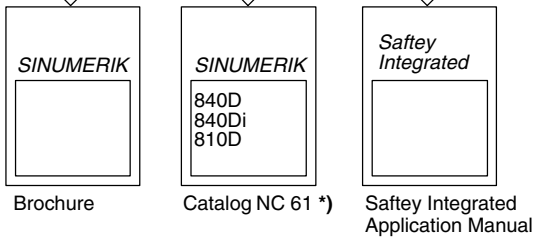
<p>From Name:  Address of your company/department Address: _____ Zip code: _____ City: _____ Telephone: _____ / _____ Fax: _____ / _____</p>	<p><b>Suggestions</b> <b>Corrections</b></p> <p>For publication/manual: SINUMERIK 840D sl/840Di sl SINUMERIK 840D/840Di/810D Commissioning CNC Part 2 (HMI)  Manufacturer/Service Documentation</p> <p>Commissioning Manual Order No.: 6FC5397-3AP10-1BA0 Edition: 01/2006</p> <p>Should you come across any printing errors when reading this publication, please notify us on this sheet. Suggestions for improvements are also welcome.</p>
--	--

**Suggestions and/or corrections**

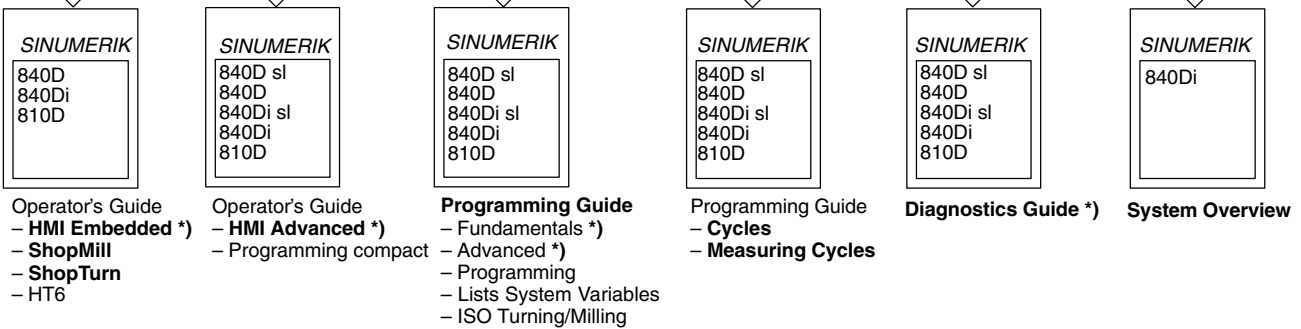


# Overview of SINUMERIK 840D/840Di/810D Documentation (01/2006)

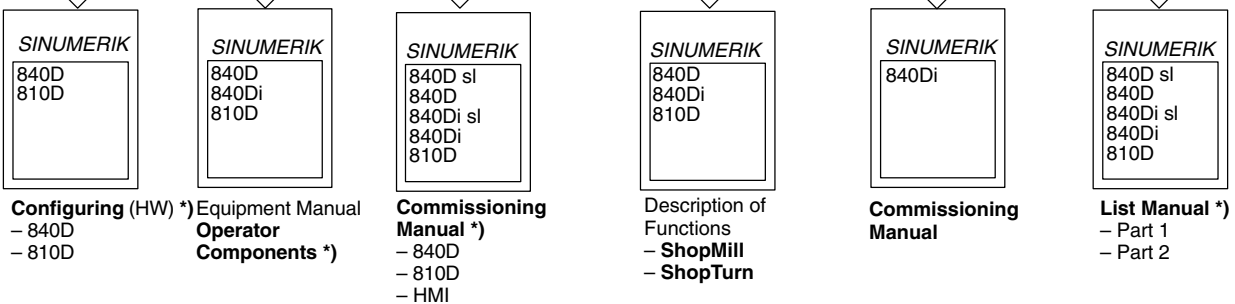
## General Documentation



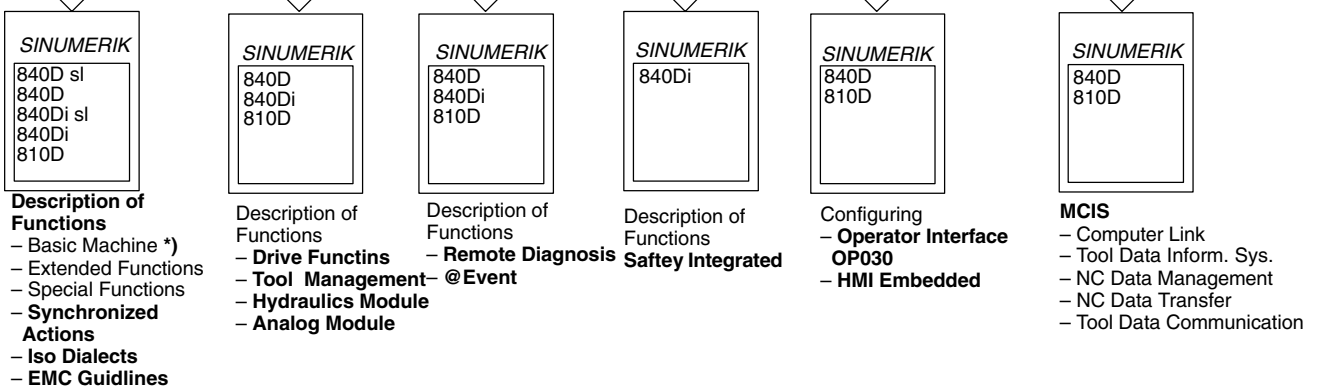
## User Documentation



## Manufacturer/Service Documentation



## Manufacturer/Service Documentation



## Electronic Documentation

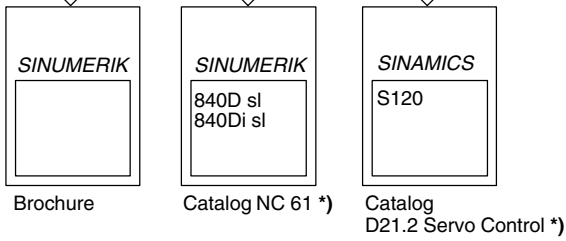


DOCONCD \*)  
DOCONWEB

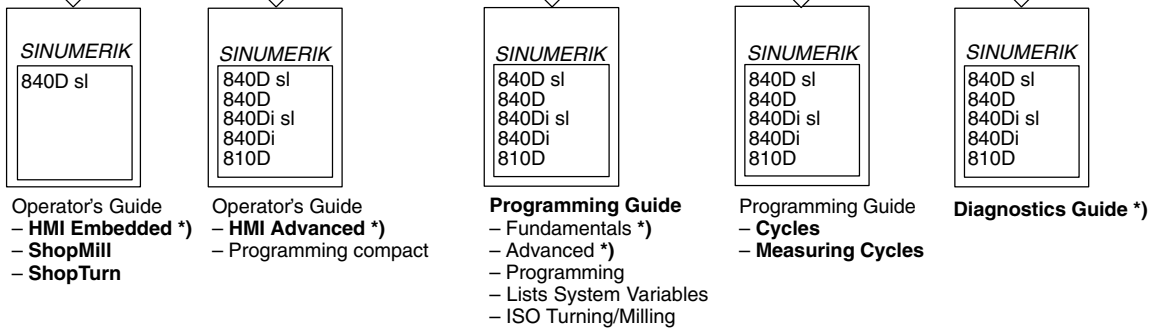
\*) These documents are a minimum requirement

# Overview of SINUMERIK 840D sl/840Di sl Documentation (01/2006)

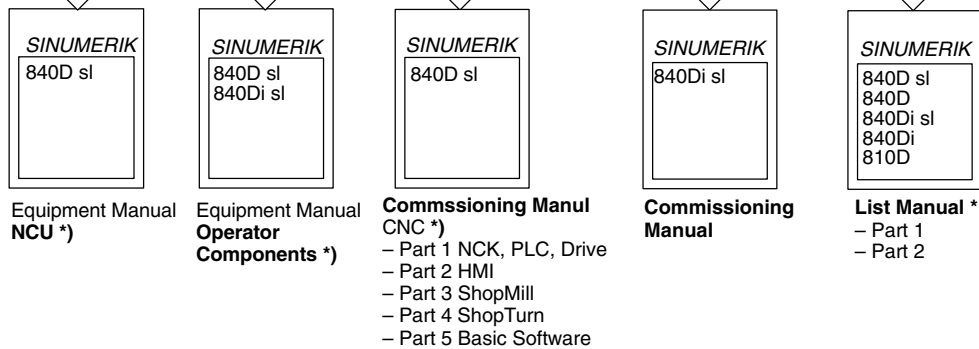
## General Documentation



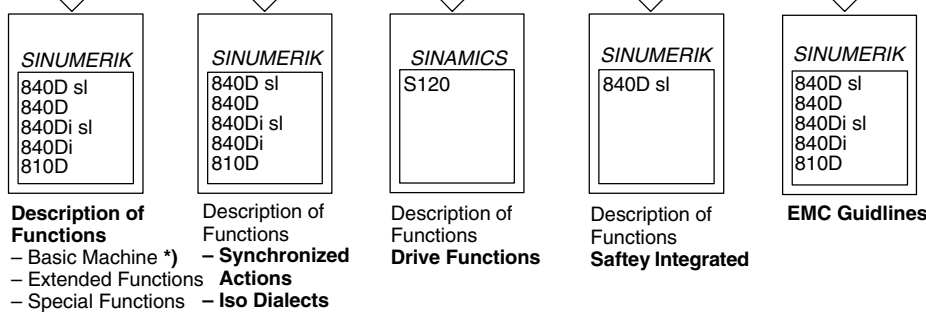
## User Documentation



## Manufacturer/Service Documentation



## Manufacturer/Service Documentation



## Electronic Documentation



\*) These documents are a minimum requirement