

MTX 15VRS

Commissioning

Commissioning Manual
R911393281

Edition 04



Title	MTX 15VRS Commissioning
Type of Documentation	Commissioning Manual
Document Typecode	DOK-MTX***-STARTUP*V15-CO04-EN-P
Internal File Reference	RS-69d58f469dc0a1080a347e8633e67190-4-en-US-3
Change record	Edition 04, 2021-03 Refer to tab. 1-1 "Change record" on page 1
Copyright	© Bosch Rexroth AG 2021 All rights reserved, also regarding any disposal, exploitation, reproduction, editing, distribution, as well as in the event of applications for industrial property rights.
Liability	The specified data is intended for product description purposes only and shall not be deemed to be a guaranteed characteristic unless expressly stipulated in the contract. All rights are reserved with respect to the content of this documentation and the availability of the product.
Editorial Department	Engineering Automation Systems Solution Integration HMI and NC Control, PeSchn (TaDo/MePe)

Table of Contents

	Page
1 About this documentation.....	1
1.1 Validity of the documentation.....	1
1.2 Required and supplementing documentation MTX.....	2
1.2.1 Selection/compilation.....	2
1.2.2 Configuration.....	2
1.2.3 Commissioning.....	3
1.2.4 Operation.....	4
1.2.5 OEM Engineering.....	5
1.2.6 AddOns.....	5
1.3 Information representation.....	5
1.3.1 Names and abbreviations.....	5
1.4 Customer feedback.....	6
2 Important instructions on use.....	7
2.1 Intended use.....	7
2.1.1 Introduction.....	7
2.1.2 Areas of use and application.....	7
2.2 Unintended use.....	8
3 Mounting, demounting and electric installation.....	9
3.1 Installation notes.....	9
3.2 Housing dimensions.....	11
3.3 Mounting the control.....	11
3.4 Mounting S20 I/O modules.....	13
3.5 Mounting extension modules.....	14
3.6 Demounting the control and the control bus base.....	15
3.6.1 Demounting steps.....	15
3.7 Electric installation.....	16
3.7.1 External power supply unit	16
3.7.2 Voltage supply for the control.....	17
3.7.3 24 V voltage supply.....	18
3.7.4 Grounding	19
3.7.5 Shielding.....	20
3.8 Battery.....	20
4 XM42 commissioning.....	23
4.1 Commissioning and operation.....	23
4.1.1 General information.....	23
4.1.2 Commissioning the MTX XM control.....	23
4.1.3 Setting the IP address for the XF5 port.....	23
4.1.4 IndraWorks Engineering via the programming interface XF31.....	24
4.1.5 Control and display indicators.....	24

Table of Contents

	Page
4.1.6	Control-internal data on "/OEM"..... 28
4.1.7	Behavior in case of fatal exception (SysError)..... 28
4.1.8	Creating a core dump..... 29
4.1.9	Initial firmware..... 29
4.1.10	Bootling..... 31
4.1.11	Fallback and recovery..... 32
4.1.12	Selecting the start mode..... 32
4.1.13	Runtime menu..... 35
4.1.14	IndraControl First Touch..... 36
5	General information..... 45
5.1	Hardware..... 45
5.2	IndraWorks Operation Desktop..... 45
5.2.1	General information..... 45
5.3	IndraWorks - Engineering..... 47
5.3.1	General information..... 47
5.3.2	Creating a project..... 47
5.3.3	Compatibility..... 48
5.3.4	User management..... 49
5.3.5	Version control..... 52
5.3.6	Archiving and restoring MTX projects..... 53
5.4	SCP Analyzer..... 65
5.4.1	General information..... 65
5.4.2	Display elements..... 65
5.4.3	SCP object hierarchy..... 65
5.4.4	Interpreting Values..... 66
6	HMI commissioning..... 67
6.1	Introduction..... 67
6.2	ScreenFrame..... 68
6.2.1	General information..... 68
6.2.2	Header (machine status area)..... 68
6.2.3	OP-panel (operating keys)..... 69
6.2.4	F-panel (function keys)..... 69
6.2.5	M-panel (machine keys)..... 70
6.2.6	F-/M-/OP-panel function overview..... 71
6.3	Screen application..... 75
6.3.1	General information..... 75
6.3.2	ACI screens..... 76
6.3.3	DBT1 screens (tool management)..... 90
6.3.4	SystemData screens (system data displays)..... 92
6.3.5	List screens..... 94
6.3.6	Programming screens..... 95
6.3.7	WinStudio..... 133
6.3.8	Operating screens..... 133

	Page
6.4	Screens..... 133
6.5	MTX screens..... 134
6.5.1	General information..... 134
6.5.2	Validating MTX screens..... 135
6.5.3	Active channel number - Transferring to PLC/Changing via PLC..... 137
6.6	NC simulation..... 137
6.6.1	General information..... 137
6.6.2	Simulation models..... 138
7	Control commissioning..... 141
7.1	Introduction..... 141
7.2	MTX control, configuration..... 141
7.2.1	Configuring the MTX..... 141
7.3	Adding the MTX control to the project..... 147
7.3.1	Creating a project..... 147
7.3.2	Adding MTX to the project..... 148
7.4	MTX control, export and import..... 148
7.4.1	Exporting MTX nodes..... 148
7.4.2	Importing MTX nodes..... 149
7.5	MTX Emulation..... 151
7.5.1	General information..... 151
7.5.2	Starting and completing..... 151
7.5.3	Data storage..... 152
7.5.4	Compatibility..... 153
7.6	MTX system status..... 153
7.6.1	Introduction..... 153
7.6.2	General information..... 154
7.6.3	Tabs..... 156
7.6.4	Startup configuration..... 158
7.7	MTX - Setting the time and date..... 160
7.8	MTX diagnostics (MZA, ProVi, VCPs)..... 163
7.8.1	Machine fault and status display (MZA)..... 163
7.8.2	ProVi..... 171
7.8.3	Diagnostic messages on small operator panels..... 171
7.9	Setting up NC cycles..... 173
7.9.1	Input masks..... 173
7.9.2	Parameter settings for cycle calls..... 173
7.9.3	Subroutines..... 174
7.9.4	SD variables..... 174
7.9.5	Using existing projects..... 174
7.10	Configuring the tool management..... 175
7.10.1	Basics..... 175
7.10.2	General configuration tools..... 179
7.10.3	General DB configuration..... 204
7.10.4	Configuring the database..... 213
7.10.5	Tool catalog..... 242

Table of Contents

	Page
7.10.6 User interface.....	259
7.10.7 Interfaces.....	291
7.10.8 Tool management standard tool change.....	303
7.11 Motion.....	341
7.11.1 NC file system (MTX Navigator).....	341
7.11.2 NC configuration (machine parameters).....	356
7.11.3 Mount directories.....	356
7.12 Sercos.....	358
7.12.1 Axis commissioning.....	358
7.12.2 Sercos device failure.....	392
7.12.3 Drive-integrated Safety Engineering.....	394
7.13 Safety.....	410
7.13.1 MTX control, commissioning the SafeLogic.....	410
7.13.2 Commissioning drives with CIP Safety on Sercos.....	416
8 VAM simulator.....	425
8.1 Using the integrated VAM for touch devices.....	425
8.2 General description.....	425
8.3 Configurator.....	426
8.4 The application.....	427
9 Upgrading the CMLx5 control to the XM42 control.....	429
9.1 Differences between the CMLx5 control to the XM42 control.....	429
9.2 Requirements for conversion.....	429
9.3 Analysis of the available hardware.....	429
9.4 General mode of operation.....	429
9.5 Ready contact	441
10 Appendix.....	443
10.1 Basic data type collection.....	443
10.2 Tool lists configuration file.....	449
10.3 Interfaces.....	458
10.3.1 Status bits.....	458
10.3.2 Tool catalog.....	461
10.3.3 Bitmap libraries.....	547
11 Service and support.....	553
Index.....	555

1 About this documentation

Editions of this documentation

Edition	Release date	Note
01	2019-07	First edition
02	2020-07	chapter 6.6 "NC simulation" on page 137 chapter revised and supplemented (WP-00076632) New: chapter "Communication with the PLC" on page 178 chapter "Predefined tool types (Standard Types)" on page 244 revised chapter "Functions of the tool type editor" on page 248 revised und and notes added chapter "Defining the coordinate system display of the tool editor" on page 252 revised und and notes added chapter "Dialog New Edit/Insert Screen" on page 285: Table revised
03	2020-11	CML removed (WP-00074604)
04	2021-03	chapter 6.6.2 "Simulation models" on page 138 revised (Defdb00202612) chapter "General information" on page 204 supplemented (Defdb00202612) chapter "Defining basic settings of the tool management" on page 224: StartMode supplemented (Defdb00202612)

Tab. 1-1: Change record

1.1 Validity of the documentation

General information

This documentation is intended for users commissioning an MTX control. Apart from a complete overview, configuration of the axes and the user interface as well as the PLC data are described.

Overview on target groups and product phases

The following graphic refers to the bordered activities, product phases and target groups of the present documentation.

Example: The target group "Programmer" can execute the "configuration" in the product phase "Engineering" using this documentation.

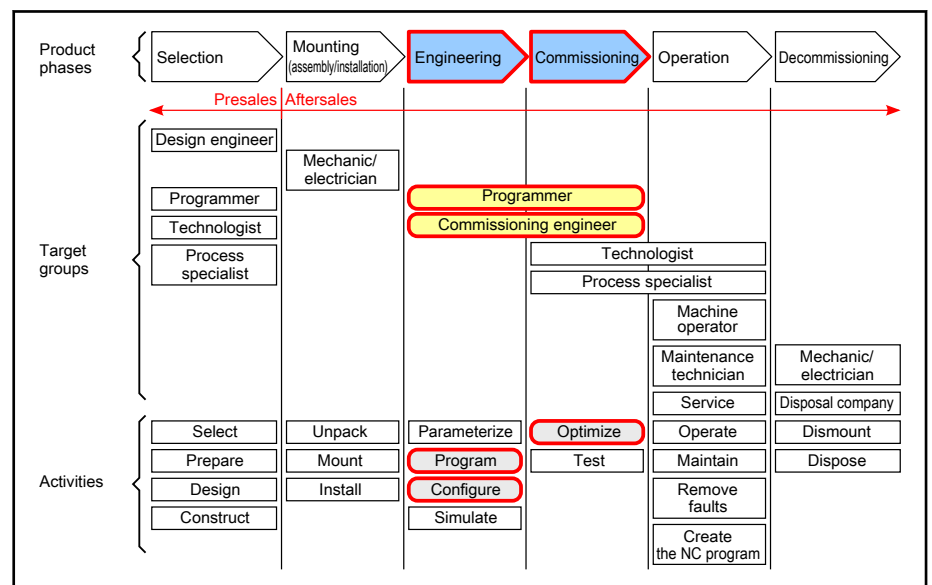


Fig. 1-1: Assigning this documentation to the target groups, product phases and target group activities

1.2 Required and supplementing documentation MTX

1.2.1 Selection/compilation

Documentation titles with type codes and part numbers

MTX 15VRS System Description DOK-MTX***-SYS*DES*V15-PRRS-EN-P, R911397742 This documentation describes the MTX control. It includes the designs, technical data, interfaces as well as the configuration of the control components.
MTX 14VRS SafeLogic System Overview DOK-MTX***-SL**SYS*V14-PRRS-EN-P, R911341698 This documentation describes the use of the safety control SafeLogic in the MTX.
IndraControl XM42 Controls DOK-CONTRL-XM4X*CTRL**-ITRS-EN-P, R911345566 This documentation describes the IndraControl XM42 controls.

RS Corresponding edition
Tab. 1-2: MTX documentation overview - Selecting/composing

1.2.2 Configuration

Documentation titles with type codes and part numbers

MTX 15VRS Machine Parameters DOK-MTX***-MA*PAR**V15-RERS-EN-P, R911400176 This documentation describes structure and the adjustment of available parameters of the MTX. It also includes the functions of the NC configurator and its operation.
MTX 15VRS PLC Interface DOK-MTX***-PLC*INT*V15-PRRS-EN-P, R911400172 This documentation describes the interface signals and the program function blocks for the integrated PLC.
MTX 15VRS Functional Description - Basics DOK-MTX***-NC*F*BA*V15-RERS-EN-P, R911401401 This documentation describes the basic functions of the MTX. The basic commissioning steps and the control functions are provided as description and handling instruction.
MTX 15VRS Functional Description - Extension DOK-MTX***-NC*F*EX*V15-RERS-EN-P, R911393316 This documentation describes the extended functions of the MTX. The basic commissioning steps and the control functions are provided as description and handling instruction.
MTX 15VRS Functional Description - Special Functions DOK-MTX***-NC*F*SP*V15-RERS-EN-P, R911393309 This documentation describes the special functions of the MTX. The basic commissioning steps and the control functions are provided as description and handling instruction.

MTX Free Form Surface Milling

DOK-MTX***-FREEFORM***-APRS-EN-P, R911341435

This documentation describes the free form surface milling process with the MTX control. CNC programs generated by a CAD/CAM system are used as basis for the entire process. Overview over the following topics: Description of the MTX or CNC function for freeform surface milling, strategy during the NC parameterization process, boundary conditions when generating CNC programs with CAM software.

MTX Conversion of MTX Projects

DOK-MTX***-PROCONV****-PRRS-EN-P, R911342484

This documentation provides support during the conversion of MTX 1.x projects to MTX 2G. This project planning manual provides information on how to convert a project and identifies potential difficulties during the conversion.

RS

Corresponding edition

Tab. 1-3:

MTX documentation overview - Configuring

1.2.3 Commissioning

Documentation titles with type codes and part numbers

MTX 15VRS Commissioning

DOK-MTX***-STARTUP*V15-CORS-EN-P, R911393281

This documentation describes the commissioning of the MTX control. Apart from a complete overview, commissioning and configuration of the axes and the user interface as well as the PLC data are described.

IndraWorks 15VRS Basic Libraries, IndraLogic 2G

DOK-IL*2G*-BASLIB**V15-LIRS-EN-P, R911398633

This documentation describes the system-comprehensive PLC libraries.

IndraWorks 15VRS Field Buses

DOK-IWORKS-FB*****V15-APRS-EN-P, R911393284

This documentation describes the field bus and local periphery connections supported by the MLC and MTX systems. The focus of this documentation lies in the configuration, parameterization, commissioning and diagnostics of different periphery connections. It is the basis for the online help.

IndraWorks 14VRS WinStudio 7.4

DOK-IWORKS-WINSTUD*V14-APRS-EN-P, R911341585

This "User Manual and Technical Reference Book" supports the user in achieving the best results with the "Rexroth WinStudio"™ software. The document provides technical information and step-by-step instructions to create web-enabled HMI/SCADA programs.

IndraWorks 15VRS Software Installation

DOK-IWORKS-SOFTINS*V15-CORS-EN-P, R911393450

This documentation describes the IndraWorks installation.

IndraWorks 15VRS Engineering

DOK-IWORKS-ENGINEE*V15-APRS-EN-P, R911393303

This documentation describes the use of IndraWorks in which the Rexroth Engineering tools are integrated. It includes instructions on how to work with IndraWorks and how to operate the oscilloscope function.

IndraWorks 15VRS PLC Programming System IndraLogic 2G

DOK-IWORKS-IL2GPRO*V15-APRS-EN-P, R911396137

This documentation describes the PLC programming tool IndraLogic 2G and its use. The documentation includes the basic use, first steps, visualization, menu items and editors.

IndraWorks 15VRS HMI

DOK-MLC***-HMI****V15-APRS-EN-P, R911399270

This documentation describes the functions, configuration and operation of the user interfaces IndraWorks HMI Engineering and IndraWorks HMI Operation.

MTX 15VRS Shape Cutting Technology

DOK-MTX***-TECHCUT*V15-CORS-EN-P, R911399147

This documentation describes the shape cutting technology used by the machines to cut materials using an NC-controlled tool head.

RS

Corresponding edition

Tab. 1-4:

MTX documentation overview - Commissioning

1.2.4 Operation

Documentation titles with type codes and part numbers

MTX 12VRS Block Pre-Run

DOK-MTX***-BLK*RUN*V12-APRS-EN-P, R911334379

This documentation explains to the machine manufacturer how to setup the "Block pre-run" function at the machine for the end user.

MTX 15VRS Programming Manual

DOK-MTX***-NC**PRO*V15-RERS-EN-P, R911393318

This documentation describes the standard programming of the MTX control. Apart from the basics of NC programming, the use of NC functions according to DIN 66025 as well as the NC functions with high-level language syntax and CPL functions are described.

MTX 15VRS Standard NC Operation

DOK-MTX***-NC*OP***V15-APRS-EN-P, R911393314

This documentation describes the operation of the standard user interface of the NC control of the MTX. It includes the operation of the interface, the NC program development as well as the tool management.

MTX 15VRS Multitouch

DOK-MTX***-MULTI***V15-APRS-EN-P, R911393311

The Multitouch user interface of the NC control MTX is described in this documentation.

MTX 15VRS Standard NC Cycles

DOK-MTX***-NC*CYC**V15-PRRS-EN-P, R911394940

This documentation describes the application of the standard cycles of the different technologies for the MTX control.

MTX 15VRS NC Simulation

DOK-MTX***-NC*SIM**V15-APRS-EN-P, R911393273

This documentation describes the NC simulation for the MTX control.

MTX 15VRS Measuring Functions

DOK-MTX***-MES*FUN*V15-APRS-EN-P, R91194938

This documentation describes the measuring cycles of the MTX control.

RS

Corresponding edition

Tab. 1-5:

MTX documentation overview - Operating

1.2.5 OEM Engineering

Documentation titles with type codes and part numbers

MTX 13VRS Automation Interface DOK-MTX***-AUT*INT*V13-APRS-EN-P, R911337274 This documentation describes the script-based access to IndraWorks project data via the interface of the Automation Interface. Different objects including code examples are described. The Automation Builder is also described in this manual.
MTX 15VRS OPC Communication DOK-MTX***-OPC*COM*V15-PRRS-EN-P, R911399272 This documentation describes the syntax and the structure of the items for the communication with Bosch Rexroth devices.
IndraWorks OPC UA Communication DOK-IWORKS-OPC*UA*****-APRS-EN-P, R911379309 This documentation describes the OPC UA communication of the MLC and MTX control systems.

RS Corresponding edition
Tab. 1-6: MTX documentation overview - OEM engineering

1.2.6 AddOns

Documentation titles with type codes and part numbers

MTX 11VRS Action Recorder DOK-MTX***-ACR*****V11-APxx-EN-P, R911329943 This documentation describes the MTX action recorder. It includes the installation and commissioning as well as interface signals, application and operation.
MTX 15VRS Efficiency Workbench MTX cta, MTX ega DOK-MTX***-EWB*****V15-APRS-EN-P, R911400178 This documentation describes the mode of operation and the area of application of the analysis tool MTX cta and MTX ega.
MTX Remote Condition Monitoring DOK-MTX***-RCM*****V01-APRS-EN-P, R911334383 This documentation describes the operation of the Remote Condition Monitoring System.
MTX visIREC User Documentation DOK-MTX***-VISIREC*V01-APRS-EN-P, R911344242 This documentation describes the analysis tool visIREC used to optimize the free form surface milling process. 2D or 4D display of path-related data. 2D or 4D display of coordinate-related data. Analysis of critical areas (path and orientation deviation). Comparing the programmed NC blocks to the interpolated NC blocks.

xx / RS Corresponding edition
Tab. 1-7: MTX documentation overview - AddOns

1.3 Information representation

1.3.1 Names and abbreviations

Term	Explanation
IWE	IndraWorks Engineering
IWO	IndraWorks Operation
OWG	Optical waveguide
NC	Numerical Control

IL	IndraLogic
OEM	Original Equipment Manufacturer
Profibus	Communication connection
Sercos	Communication connection
ProfiNet	Communication connection

Tab. 1-8: Terms and abbreviations

1.4 Customer feedback

Customer requests, comments or suggestions for improvement are of great importance to us. Please email your feedback on the documentations to Feedback.Documentation@boschrexroth.de. Directly insert comments in the electronic PDF document and send the PDF file to Bosch Rexroth.

2 Important instructions on use

2.1 Intended use

2.1.1 Introduction

Bosch Rexroth products are developed and manufactured according to the state-of-the-art. The products are tested prior to delivery to ensure operating safety and reliability.

The products may only be used as intended. If they are not used as intended, situations occur that result in damage to property or injury to persons.



Bosch Rexroth shall not assume any warranty, liability or payment of damages in case of damage resulting from a non-intended use of the products; the use shall solely bear all risks from unintended use of the products.

Before using Bosch Rexroth products, the following requirements have to be met to guarantee the intended use of the products:

- Anybody handling Bosch Rexroth products in any way is obliged to read and consent to the relevant safety instructions and the intended use.
- Hardware products may not be altered and have to remain in their original state; i.e. no structural changes are permitted. It is not permitted to decompile software products or alter source codes.
- Do not install damaged or defective products or use them in operation.
- It has to be ensured that the products have been installed as described in the relevant documentation.



Ensure that the data present in the control or entered or read in by the user is correct before applying it to exclude unwanted axis motion. It can be the following invalid or old data:

- Part programs
- ZO tables
- Compensation tables
- Tool tables
- Permanent CPL variables
- Remanent PLC data
- Permanent system data

2.1.2 Areas of use and application

The MTX control is used to

- program contour and machining technology (path feed, spindle speed, tool change) of a workpiece.
- guide a machining tool along a programmed path.

Feed drives, spindles and auxiliary axes of a machine tool are activated via Sercos interface.



This additionally requires I/O components for the integrated PLC which - together with the actual CNC - control the machining process as a whole and also monitors this process with regard to technical safety.

It may only be operated with the explicitly specified hardware component configurations and combinations and only with the software and firmware specified in the appropriate documents and functional descriptions.

The CNC system MTX provides the perfect CNC system solution for cutting and forming for the following technologies:

- Rotate
- Milling
- Drilling
- Grinding
- Bending
- Nibbling
- Punching
- Contour cutting
- Handling

2.2 Unintended use

Using the CNC system MTX outside the above-referenced areas of application or under operating conditions other than the conditions described in the documentation and the specified technical data, is defined as "unintended use".

The CNC system MTX must not be used if ...

- it is subjected to operating conditions not corresponding to the specified ambient conditions. Operation under water, under extreme temperature fluctuations or under extreme maximum temperatures is prohibited.
- Furthermore, the CNC system MTX shall not be used for applications not expressly approved of by Bosch Rexroth. Therefore, please read the information given in the general safety instructions!
- The CNC system MTX may not be used in systems or machines connected to the internet via an unsecure network connection. Otherwise, malfunctions or a control failure can result due to unauthorized access.

3 Mounting, demounting and electric installation

3.1 Installation notes

NOTICE

Module destruction due to electrostatic discharge

The XM42 control contains components that can be damaged or destroyed by electrostatic discharge. Comply with the required safety measures against electrostatic discharge (ESD) acc. to EN 61340-5-1 when operating the control.

- **Mounting location**

The XM42 control has the degree of protection IP20 and is thus intended for the use in a closed control cabinet or control box (terminal box) of the degree of protection IP54 or higher.

- **Mounting rail**

Mount the XM42 control on a 35 mm standard mounting rail. The preferred overall height of the mounting rail is 7.5 mm (corresponds to TH 35-7.5 acc. to EN 60715).

Mount the control and the bus base as described in the following.

The fastening distance of the mounting rails may not exceed 200 mm. This distance is required to ensure stability while mounting and demounting the control or the S20 I/O modules.

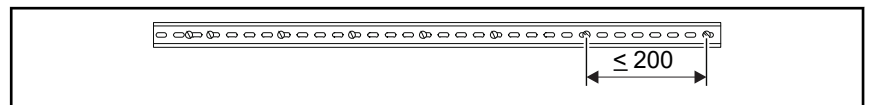


Fig. 3-1: Mounting rail fastening (in mm)

NOTICE

Electronic damage due to fastening elements

Danger of malfunction

If the fastening elements (screws, rivets,...) are too high, the bus bases do not correctly engage on the mounting rail. Use only elements with a structural height of up to 3 mm to fasten the mounting rail.

- **End clamp**

Fasten end clamps on both sides of the IndraControl XM station (control, extension modules, S20 I/O modules).

End clamps guarantee the correct fastening of the XM42 control and the S20 I/O modules connected to them on the mounting rail and they are used as lateral end elements.

Always fasten the left end clamp of the station before mounting the control. This ensures the following:

- It impedes the shifting of the control.
- The installation place for the end clamps is guaranteed.
- If the control has to be replaced, there is sufficient space to separate the control from the S20 bus base.
- They have a counterpressure against the engaging forces when mounting the S20 bus bases in series to the control.

- Do not route cables parallel to motor cables or other strong interference sources to avoid coupling of interferences
- The cabling of the Ethernet wires may not leave the building
- The LED displays on the operating panel must not be covered
- Observe the bending radius of the cables used when laying the connecting lines
- Use strain reliefs for all cables
- Install the control only horizontally on a mounting rail attached to a wall
- Keep the maximum distance possible from interference sources
- Provide the following minimum distances for sufficient cooling:

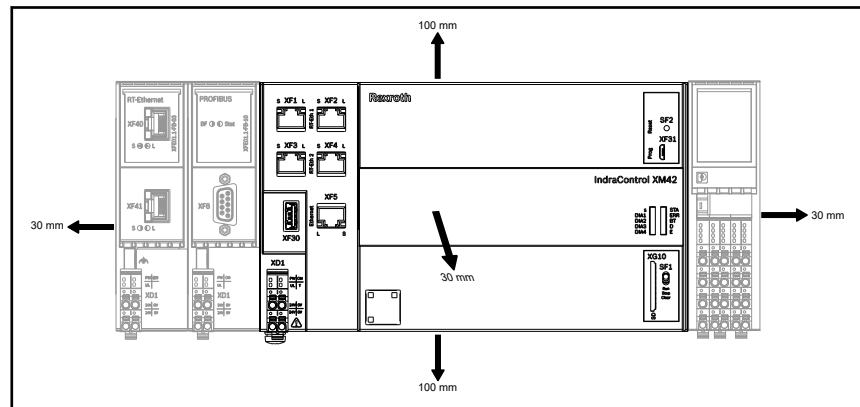


Fig. 3-2: Minimum distances for the circulation of ambient air

In case of a several line design, the supply air has to be measured under each line and its limit value has to be observed. Also note the values given in the chapter "Ambient condition".

- Additionally, provide sufficient distance for mounting, demounting, plugs and cables

For more information on mounting, demounting and connecting lines, refer to the application description "IndraControl S20: System and Installation", part no. [R911335988](#).

3.2 Housing dimensions

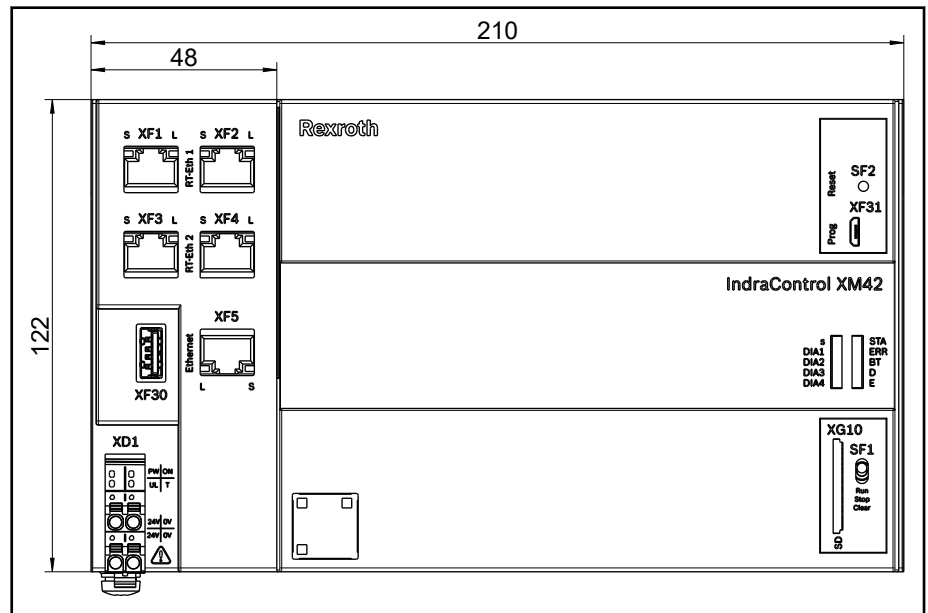


Fig. 3-3: Front view

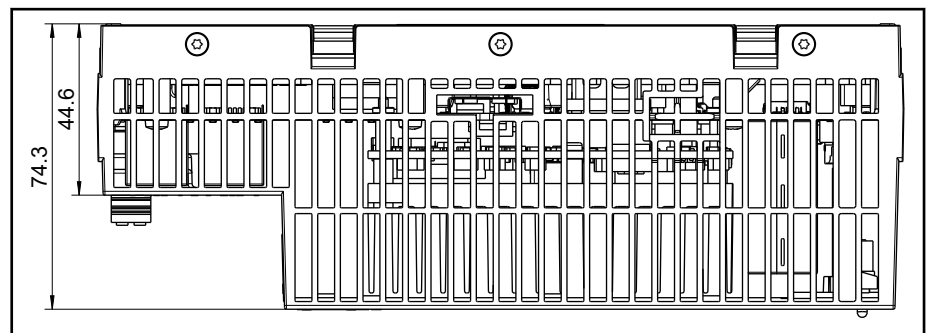


Fig. 3-4: Bottom view

3.3 Mounting the control

NOTICE

Damages at the contacts by tilting the modules

Attach the modules vertically on the bus base and remove them vertically from the bus base.

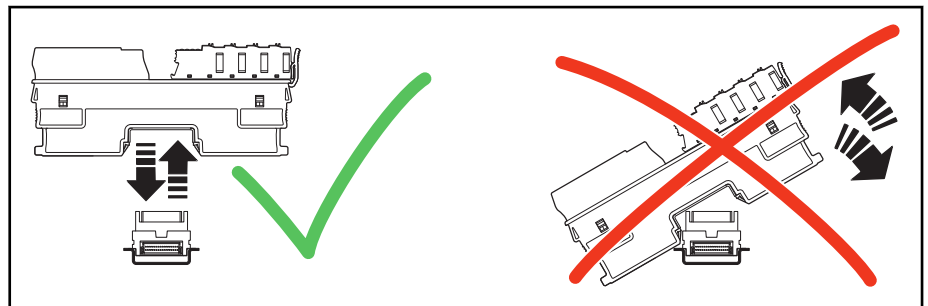


Fig. 3-5: Positioning and removing modules only vertically

NOTICE

Component destruction due to plug mounting under voltage!

- Before mounting or demounting components, disconnect the control - including its components - from voltage
- Connect the voltage only after the control and its components have been set up

NOTICE

Possible damage to property due to improper top-hat rail mounting

- Fasten the top-hat rail adequately.
- Connect the top-hat rail to a functional earth.
- Mount the control on the top-hat rail.
- Install the control in a control cabinet or an appropriate housing.

Mounting the control bus base

1. Correctly position the bus base as per the descriptions "Right Side" and "Left Side".

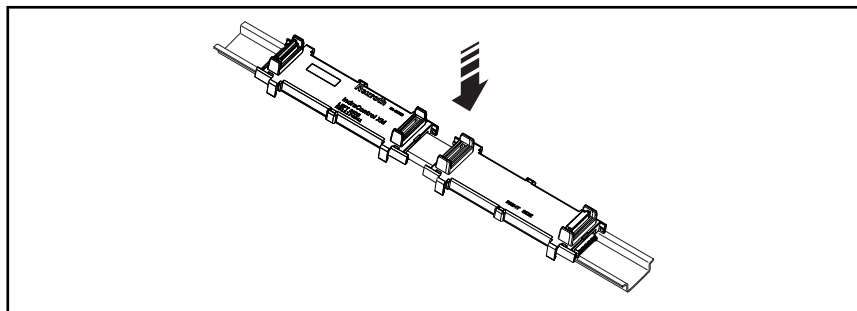


Fig. 3-6: Control bus base

2. Press the bus base on the top-hat rail until all latching are safely closed.
3. Mount the S20 bus bases and the bus bases for the extension modules, refer to [fig. 3-8 "Interconnect S20 bus bases and to the control bus base" on page 13](#) and [fig. 3-10 "Interconnect extension bus bases and connect them to the control bus base" on page 14](#).

Mounting the control on the control bus base



Before mounting the control on the top-hat rail, mount the bus bases for the modules to the right and to the left of the control.

1. Press the control on the premounted control bus base until all latching are safely closed.

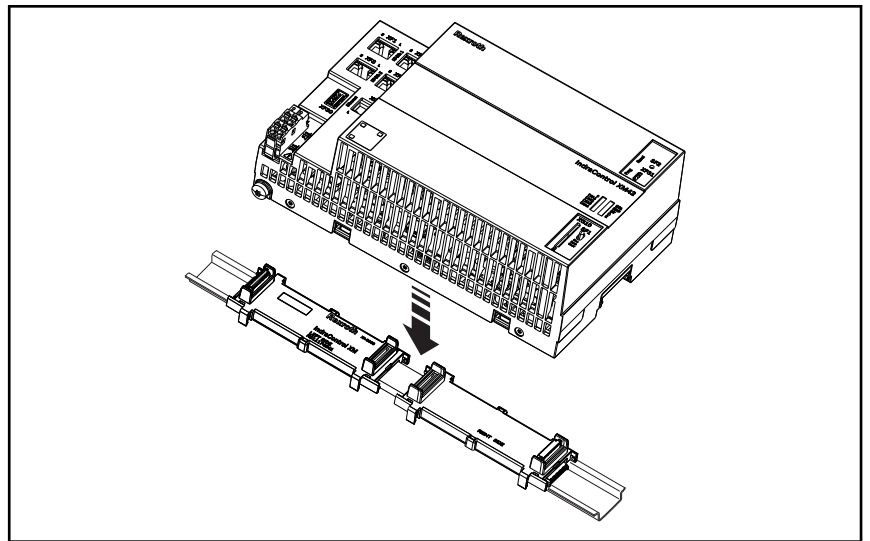


Fig. 3-7: Snapping the control



Note the minimum distances for control cooling (see [chapter 3.1 "Installation notes" on page 9](#)).

3.4 Mounting S20 I/O modules



The S20 bus bases can only be mounted if the control is not attached to the bus base.

Mounting S20 bus bases

1. Press the S20 bus bases for the S20 modules on the top-hat rail on the right next to the control bus base until all latchings are safely closed, see "A" in the following figure.
2. Move the first S20 bus base (see "B") to the left and connect the S20 bus base to the bus base of the control, see "D" in the following figure.
3. Move the other S20 bus bases to the left until all bus bases are interconnected, see "C" in the following figure.

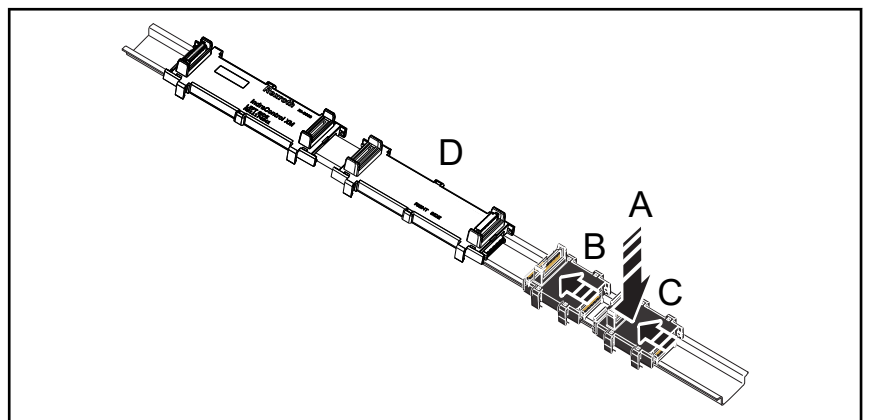


Fig. 3-8: Interconnect S20 bus bases and to the control bus base

Mounting S20 module to the S20 bus base

1. Press the S20 modules on the premounted bus bases.

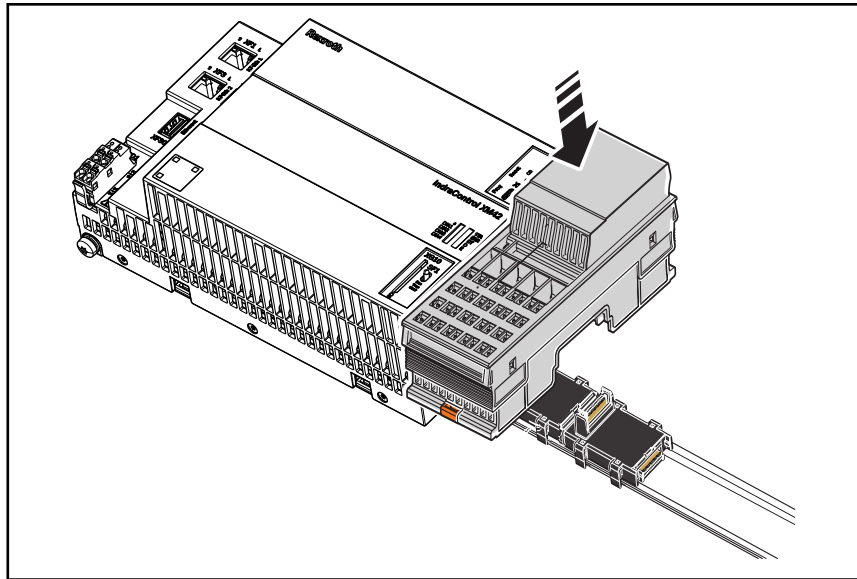


Fig. 3-9: Positioning S20 modules

3.5 Mounting extension modules



The extension bus bases can only be mounted if the control is not attached to the control bus base.

Mounting extension bus bases

1. Press the extension bus base for the extension modules on the top-hat rail on the left next to the control bus base until all latchings are safely closed, see "A" in the following figure.
2. Move the first extension bus base (see "B" to the right) and connect the extension bus base to the bus base module of the control, see "D" in the following figure.
3. Move the other extension bus base to the right until all bus bases are connected to each other, see "C" in the following figure.

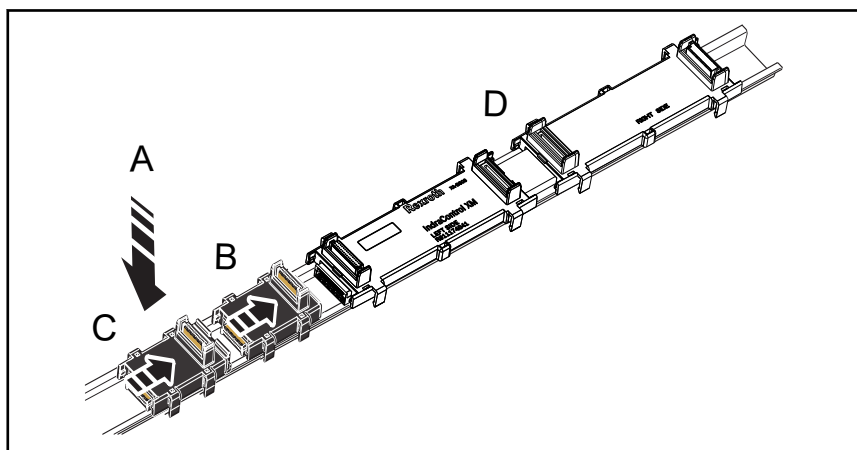


Fig. 3-10: Interconnect extension bus bases and connect them to the control bus base

Mounting extension modules to the extension bus bases

1. Press the extension modules on the premounted extension bus bases.

3.6 Demounting the control and the control bus base



To demount, use a common tool such a slotted screwdriver with a 2.5 mm blade.

3.6.1 Demounting steps

NOTICE

Component destruction due to demounting under voltage!

Before mounting or demounting components, disconnect the control - including its components - from voltage.

Removing the control from the top-hat rail

1. Remove the left end clamp.
2. Use a suitable tool (e.g. slotted screwdriver) and put it first into the upper and then into the lower disengaging mechanism (base latch) of the control and disengage the control, see "A" in the following figure. The base latches are locked in the open position.
3. Remove the control vertically to the mounting rail, see "B" in the following figure. The base latches engage again in idle position.

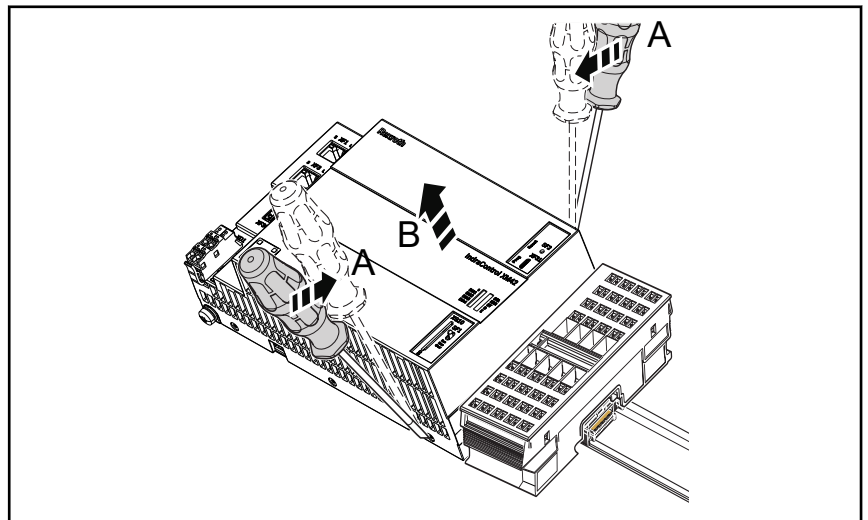


Fig. 3-11: Removing the control from the top-hat rail

The control bus base remains on the mounting rail.

Demounting the control bus base

1. Separate the bus base module of the control from the connection of the first S20 bus base by moving the bus base of the control to the left by approximately 5 mm, see "A" in the following figure.
2. Separate the bus base of the control from the connection of the first extension bus base by moving the bus base of the control to the right by approximately 5 mm.
3. Use a suitable tool (e.g. slotted screwdriver) to enter into the latching on one side, see "B", "B1" and "B2" in the following figure.
4. Lift up the bus base and remove it, see "C" in the following figure.

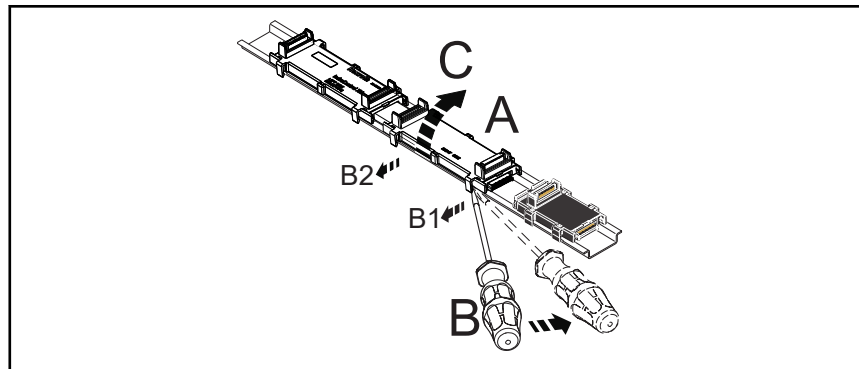


Fig. 3-12: Removing the control bus base from the top-hat rail

3.7 Electric installation

⚠ WARNING

Danger of personal injury due to incorrect mounting or electric installation!

- Any dangerous system states, which might cause personal injury, must be prevented!
- Protection upon direct and indirect contact must be ensured by the specified measures (connection to protective conductor, insulation, etc.).

NOTICE

Danger of material damage due to incorrect mounting or electric installation!

- Any dangerous system states, which might cause material damage, must be prevented!
- Protection upon direct and indirect contact must be ensured by the specified measures (connection to protective conductor, insulation, etc.).

3.7.1 External power supply unit

⚠ DANGER

Danger of lethal injury due to hazardous electric voltage

- Connect power supply units generating protective extra-low voltage (24 V) only to supply voltages designed for these power supply units. Note the overvoltage categories (refer to the documentation of the power supply unit)
- Do not apply the supply voltage to the protective extra-low voltage

All control components are supplied from 24 V voltage supplies (SELV/PELV).

Use the Bosch Rexroth power supply unit VAP01.1H-W23-024-010-NN, part number R911171065, for the logic supply. For further information on the external power supply unit and on the creation of overvoltage categories, refer to the documentation of the power supply unit.

All lines of the 24 V voltage supply have to be routed separately from lines carrying higher voltages.

All peripherals, such as digital sensors or actuators connected to the interfaces of the control, also have to comply with the criteria of safety-separated circuits.



The same external power supply unit has to supply the 24 V voltage supply of the XMx control and the extension modules. Connect the voltage supply to the XD1 plugs of the XMx control and then bridge the cables to the power connectors XD1 of the extension modules.



The 24 V voltage supply of the S20 I/O modules may not be branched off from the XD1 plug of the XMx control, but with an own connecting cable to the external power supply unit instead.



The 24 V voltage supply has to be grounded. For more detailed information, refer to the documentation of the power supply unit.

3.7.2 Voltage supply for the control

The control is supplied via the XD1 plug on the lower left.



Connect the power connector with cables with a conductor cross-section of AWG 16 (1.5 mm²).



Use only copper wires to connect the connection terminals.



Use only cables approved for temperatures of at least 60°C. In case of ambient temperatures above 55°C, use cables approved for temperatures of at least 75°C.



Observe the color-coding of the plugs.



Only the "XA-CN01" power connector is approved to connect the 24 V supply voltage for the XM42 control.



If multiple XMxx controls (24 V supply voltage U_{LS}) are connected parallelly to a central 24 voltage supply, ensure that the 0 V connection of the U_{LS} is wired without interruption to all XMx controls. If this is not observed, XMx controls might become defective.

Power connector XD1

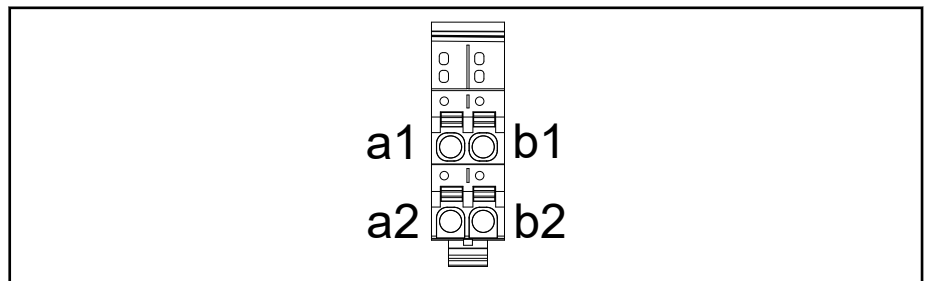


Fig. 3-13: Power connector XD1

Plug contact	Signal	Function
a1, a2	24 V	+24 V DC supply voltage (U_{LS})
b1, b2	0 V	GND (U_{LS}) (ground supply voltage)

Tab. 3-1: Pin assignment of the power connector XD1

 In the power connector, the contact a1 is bridged with a2 and the contact b1 is bridged with b2.

3.7.3 24 V voltage supply

The voltage of 24 V can be supplied with or without electrical isolation.

Setup with potential binding between internal logic and periphery supply

An easy connection method is the setup with an electrical isolation between the internal logic and the peripheral supply.

In this case, a power supply unit is sufficient to supply the control.

 In the device, the 0 V connection (U_{LS}) is electrically isolated to the ground connection.

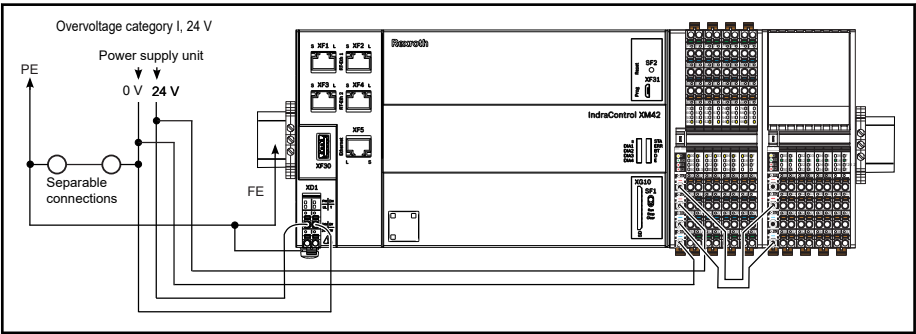


Fig. 3-14: Setup with power supply unit

 **Control does not start in case of reverse input voltage**
The feeding of 24 V at the XD1 connector is protected against polarity reversal. A polarity reversal of the 24 V and 0 V feeding (U_{LS}) does not damage the device. However, the control does not start and the U_L LED is not on.

Reference conductor connected to the protective conductor and set up with electrical isolation

Provide electrical isolation between the logic of the central processing unit and the I/O interfaces of the peripheral component groups according to DIN EN 60204-1. Accordingly, the voltage U_{LS} (24 V logic voltage) at the control is electrically isolated from the supply voltage for the I/O modules.

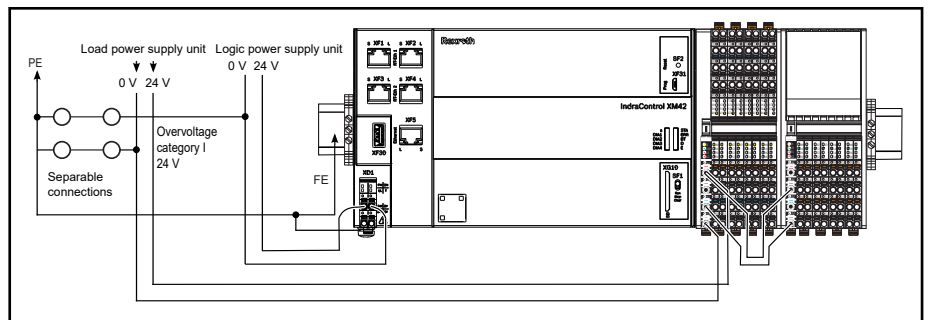


Fig. 3-15: Setup with two power supply units

Dimensioning the voltage supply

Observe the maximum currents when dimensioning the voltage supply. The operating voltage allowed has to be applied directly to the device.

The voltage must also not be exceeded if:

- there are variations in line voltage, e.g. caused by different loads of the mains
- there are varying load states, such as short-circuit, normal load, lamp load or no load

Connecting the reference conductor to the protective conductor

If the reference conductor 0 V (U_{LS}) is connected to the protective conductor system, this connection has to be arranged at a central place (e.g. at the load power supply unit). Hence, the supply current circuit is a PELV circuit.

Floating setup

For a floating setup, disconnect the reference conductor 0 V (U_{LS}) from PE and FE, refer to "separable connection" in the previously mentioned routing images (fig. 3-14 "Setup with power supply unit" on page 18 and fig. 3-15 "Setup with two power supply units" on page 19). The control is connected with 0 V (U_{LS}) and the ground connection. Thus, ground connection monitoring devices can be used.

NOTICE

Due to an earth-free setup, the Safety module XFE01.1-SY-01 might fail.

Do not use the Safety module XFE01.1-SY-01 with an earth-free setup.

3.7.4 Grounding

NOTICE

Control failure due to insufficient grounding

An optimum grounding is required to keep away possible interferences from the control and to discharge them to the ground.

Functional earth

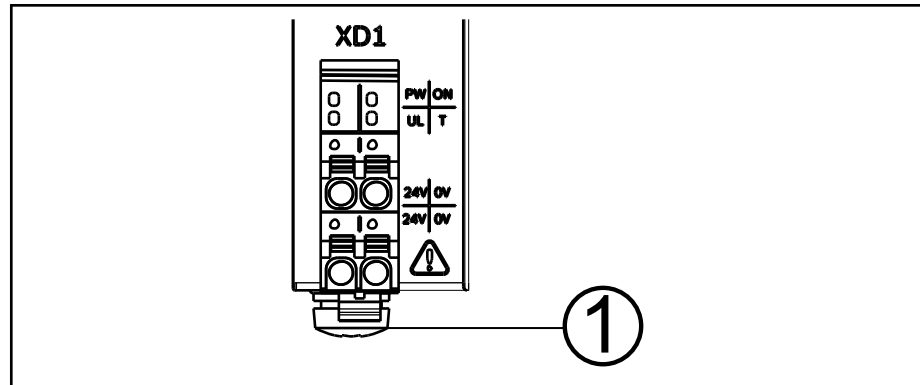


Only the functional earth (FE) is used for the XMx controls, extension modules and the S20 components. The functional earth is only used to discharge disturbances. For individuals, the functional earth is not intended as protection against electric shock.

The top-hat rail, on which the control is mounted, has to be mounted to a grounded metal carrier, e. g. the rear panel of the control cabinet.

Grounding screw

A functional grounding via the grounding screw is required to ensure optimum noise immunity. The grounding screw is positioned under the control, see [fig. 3-16 "Position of the grounding screw" on page 20](#). Connect the functional earthing a cable, which is as short as possible or, preferably, even via a ground strap. The maximum cable or ground strap length is 1 m with a cross-section of 6 mm².



①

Grounding screw

Fig. 3-16: Position of the grounding screw

Potential equalization

Potential equalization acc. to DIN VDE 0100 part 540 has to be provided between the system parts and the voltage supply.

3.7.5 Shielding

NOTICE

Control failure due to insufficient shielding

Provide sufficient shielding.

The shielding reduces any effects of interferences on the system.

Observe the following when shielding:

- Fasten the shielding as extensively as possible
- Ensure proper contact between connector and terminal
- Avoid damaging or squeezing conductors
- Note the wire specifications when connecting the shielding
- Shield the closest possible to the signal terminal points



Route all power cables and data cables in separate cable channels.

3.8 Battery



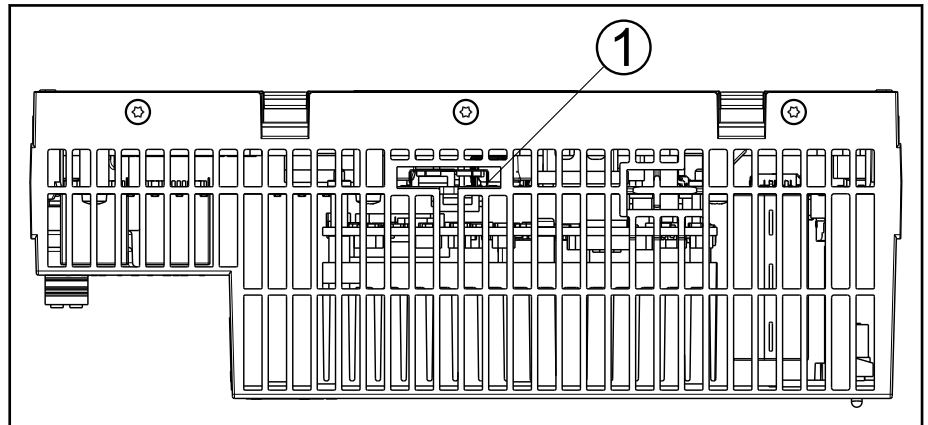
A battery is included in the device upon delivery. Battery name: Lithium battery 3.0 V CR1632 (120 mAh), manufacturer: Renata.



The battery is used to extend the buffer time of the real-time clock.



The charging condition of the battery is not monitored. Change the battery every five years. A discharged battery causes an incorrect system time.



① Battery case

Fig. 3-17: Position of the battery

Proceed as follows to exchange the battery:

1. Switch off the control.
2. Remove the control from the bus base.
3. Use a slot screwdriver to open the battery case at the opening at the XBAT connection.
4. Insert the battery and note the polarity (stamping on the housing).

4 XM42 commissioning

4.1 Commissioning and operation

4.1.1 General information

This section describes the commissioning of the hardware, the basic functions of the IndraWorks Engineering environment and general procedures.

4.1.2 Commissioning the MTX XM control

Commissioning steps

To commission the device, proceed as follows:

1. Mount the control
2. Connect the voltage supply to the XD1 connection of the control
3. Switch on the control → The initial firmware starts

See chapter [chapter 4.1.9 "Initial firmware" on page 29](#)



You can skip this section if an MTX firmware is already available.

4. Connect the project planning PC to the control via XF5
5. Use the commissioning tool "IndraControl First Touch" to make the respective network settings (IP address, subnet mask, gateway address) for the XF5 connection

Refer to [chapter 4.1.3 "Setting the IP address for the XF5 port" on page 23](#)

6. Create IndraWorks project with a control on the basis of IndraControl MTX XM42. Install the project planning and programming software "IndraWorks Engineering" in the system variant "IndraWorks MTX*" on the project planning PC.
7. Start the operating software IndraWorks Operation



Control malfunctions due to control switch-off during initialization

Note that hardware is initialized after the system firmware installation and the subsequent reboot. Wait till completion. Do not switch off the control. This procedure has to be completed. Subsequently, continue the commissioning.

4.1.3 Setting the IP address for the XF5 port

For the initial commissioning and the specification of the network settings, the commissioning tool "IndraControl First Touch" is provided. The address can only be changed without a loaded MTX firmware via this commissioning tool. If the MTX firmware is loaded via the already set IP address (default: 192.168.1.1), the IP address can subsequently be changed via the configuration and programming software "IndraWorks Engineering".

The web-based software tool "IndraControl First Touch" allows to commission and maintain the controls of the IndraControl XM generation.



For a detailed description on the commissioning tool "IndraControl First Touch", refer to [chapter 4.1.14 "IndraControl First Touch" on page 36](#).

4.1.4 IndraWorks Engineering via the programming interface XF31

Instead of the Ethernet interface XF5, the connection XF31 can be used to commission the control. Connect the USB interface of your computer to the micro USB port XF31 via a common USB data cable/charging cable (micro USB plug). Upon the first connection, the corresponding device driver (USB remote NDIS device) is installed on the computer.

It is communicated between computer and control via Ethernet, but it is only implemented as point-to-point implementation. Bridging to other networks or interconnecting switches is not possible.

The MTX system status dialog provides only a very restricted functional scope via this interface. It is recommended to use the Ethernet interface XF5.



The IP address of the XF31 port is always 192.168.234.234. The user cannot change the address.

4.1.5 Control and display indicators

LED on the XD1 plug

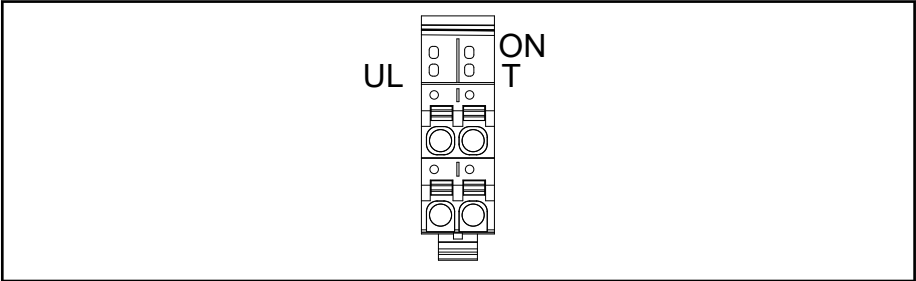


Fig. 4-1: LEDs on the XD1 plug

The following functions are assigned to the LEDs on the XD1 plug:

LED	Color	Function
U _L	Green	24 V supply voltage connected and within the permitted range
ON	Green	All FPGAs (Field Programmable Gate Array) in the system are ready for operation. If this LED is not on, an error is present and the user has to trigger a "fallback", refer to chapter 4.1.11 "Fallback and recovery" on page 32
T	Red	Maximum temperature exceeded. The power supply unit switched off the control

Tab. 4-1: Functions and LEDs on the XD1 plug

Reset button

The "Reset" button is on the top right of the control housing.



Fig. 4-2: Position of the reset button

Press the "Reset" button at running control to trigger a voltage cycle (reset).

The "Reset" button is provided with two functions triggered when the reset button is pressed while applying the supply voltage:

- Press the "Reset" button < 5 seconds: Triggering the fallback mode (see [chapter 4.1.11 "Fallback and recovery" on page 32](#))
- Press the "Reset" button > 5 seconds until the DIA3 and DIA4 LEDs flash alternately: Triggering the initial firmware (see [chapter 4.1.9 "Initial firmware" on page 29](#))



The "Reset" button can only be actuated with a sharp tool (e. g. pencil).

Operation mode switches SF1 and LEDs

The LEDs and the operation mode switch SF1 are used for operation and diagnostics.

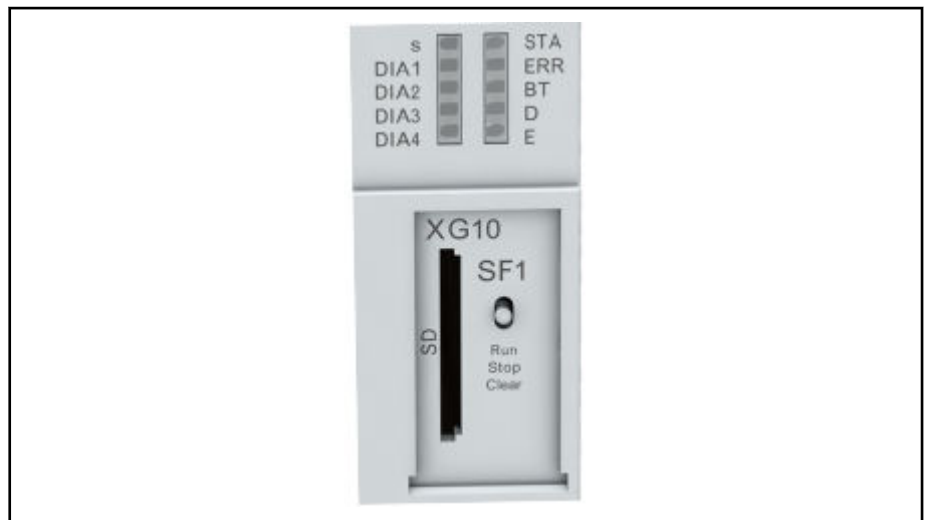


Fig. 4-3: Operation mode switches SF1 and LEDs

The XM controls are operated via the operation mode switch SF1. The user can select and enable specific operational steps via specific menus. To operate these menus, the period the switch is pressed is important:

- To select the subsequent menu items, the operating switch has to be pressed less than 3 seconds in the non-engaging position **Clear**
- To perform the action of the respective menu item, the operating switch has to be pressed more than 3 seconds in the non-engaging position **Clear**

- The control signals that the respective action is accepted indicating a static, green light of the "DIA1" LED
- To enable the selected menu item, release the operation mode switch
- If an error occurs while executing the action or during the operation, the "DIA1" LED is temporarily red

The LEDs have the following significance:

LED	Description
s	Sercos status
DIA1	Feedback of the operation mode switch
DIA2	Basic diagnostics
DIA3 (3.3V)	Boot LED BSP + initial firmware
DIA4 (5V)	Boot LED BSP + initial firmware
STA	PLC status
ERR	Error status
BT	Boot status
D	Diagnostics: S20 bus
E	Error status: S20 devices

Display of the "BT" LED

The control startup is visualized with flashing and color codes via the "BT" LED:

Flashing/color code	Control state	Description
 Off/orange	Control startup	The MTX firmware is started
 orange	BOOTSTOP	The control startup was stopped by pressing the operation mode switch in "Clear" position. The control is now in the "BOOTSTOP" menu
 Red	Error at startup	Error occurred during control startup. Startup is canceled
 off	MTX firmware startup completed	Control startup and MTX firmware start were successfully completed

Tab. 4-2: Flashing codes of the "BT" LED

Start options

The control startup can be affected via the position of the operation mode switch:

Position of the operation mode switch	Function
Run	The control is started
Stop	
Clear	The "BOOTSTOP" menu is enabled. The operation mode switch has to remain in this position when the control is switched on

Tab. 4-3: Position of the operation mode switch



After control startup, the PLC status is displayed via the "STA" LED, the error status of the system via the "ERR" LED and the communication phase of the Sercos bus via the "s" LED.

Display of the "STA" LED

Color code	Control state	Description
 Green	RUN	At least one PLC application was started
 orange	READY	No PLC application was loaded. If there is a PLC application on the control, it is in "Stop" state
 Red	STOP	All loaded PLC applications are stopped. Switching the operation mode switch to the "Run" position, all loaded PLC applications can be started

Tab. 4-4: Color codes of the "STA" LED

Display of the "ERR" LED

Color code	Control state	Description
 Green	No error	No error is present
 orange	Warning	A warning is active
 Red	Errors	An error occurred

Tab. 4-5: Color codes of the "ERR" LED

Display of the "s" LED

Flashing/color code	Sercos state	Description
 off	NRT mode	No Sercos communication active
 orange	CP0	Sercos communication phase 0 is active
 Green/orange	CP1	Sercos communication phase 1 is active
 Green/orange	CP2	Sercos communication phase 2 is active

Flashing/color code	Sercos state	Description
 Green/orange	CP3	Sercos communication phase 3 is active
 Green	CP4	Sercos communication phase 4 is active
 Green/red	Sercos class 5 error	An MTX Sercos error of class 5 occurred.
 Red	Sercos class 8 error	An MTX Sercos error of class 8 occurred. The control returns to CP0

Tab. 4-6: Color codes of the "s" LED

Display of the "DIA2" LED

Extended diagnostics are visualized via the "DIA2" LED using flashing codes:

Flashing/color code	Control state	Description
 Orange/red	Do not switch off control!	Do not switch off the control in this state in order not to damage important firm-ware components
 Orange/green	Keys are generated for encrypted connections	Upon the first control boot, the keys are generated for encrypted connections. This can take up to two minutes
 Off/orange	Switch off control!	The control has to be switched off in this state, as there is no automatic reboot

Tab. 4-7: Flashing codes of the "DIA2" LED

4.1.6 Control-internal data on "/OEM"

Do not change, move, rename or delete the following control-internal data:

- Files in the directory "/OEM/ProjectData/"
- Files in the directory "/OEM/ ProjectDataProtected/"
- File "/OEM/mx42usrfep.pxf"

4.1.7 Behavior in case of fatal exception (SysError)

If there is a fatal exception at the control (in the MTX often referred to as SysError), the following behavior can be result at the control:

LED		
DIA4	 off/red	
DIA3	 red/off	

DIA2	 orange/green	During the core dump generation
	 Green	After completing the core dump The following memory order applies: 1. Usb stick (Root directory) 2. SD card (Root directory) 3. /USER/Coredumps/ (USER partition on the control)
	 Red	Insufficient memory The core dump can subsequently be saved using the commissioning tool "IndraControl First Touch". ATTENTION: Do not switch off control. The "IndraControl First Touch" starts automatically.

Tab. 4-8: Behavior in case of fatal exception (SysError)

Subsequently, the "IndraControl First Touch" starts. Indicator is the green LED "E".

If the core dump could not be saved, it can be saved subsequently via the "First Touch" web interface using a USB stick or an SD card with sufficient memory.

It is also possible to delete unrequired files on the "/USER/" partition to provide sufficient memory.

Also refer to [chapter 4.1.14 "IndraControl First Touch" on page 36](#)

4.1.8 Creating a core dump

"Core dump" refers to the backup of the complete RAM of the control to a file. This is only required for certain errors and requested by the support.

On how to create a core dump, go to the image [fig. 4-5 "Runtime menu" on page 35](#).

4.1.9 Initial firmware

General information



The control is not provided without system firmware upon delivery. System-neutral initial firmware is installed on the control by default.

The initial firmware provides commissioning and maintenance functions. Basic functions can be activated manually without any additional tools. The controls of the device family IndraControl XM with the operation mode switch SF1 and the LED display panel provide a simple interface.

The commissioning tool "IndraControl First Touch" (see [chapter 4.1.14 "IndraControl First Touch" on page 36](#)) provides comfortable functions. If connected to a PC, it can be opened with a web browser. The comprehensive mechanisms for commissioning using the "IndraWorks Engineering" software are also supported by the IndraControl XM.

LED states

For the initial firmware, the LEDs indicate the following function states:

LED	Function	Color	Status
DIA2	Save core dump on SD card or USB medium (for diagnostic purposes)	Green/orange flashing	In process
		Green	Done
		Red	Errors
DIA3, DIA4	Operating system state	Alternatingly red flashing	Initial firmware is active
ERR	Reset to delivery state	Red	Function selected/processed
		Flashing red	Confirm selection
BT	Reset user name/password	Red	Function selected/processed
		Flashing red	Confirm selection
D	Delete network configuration	Red	Function selected/processed
		Flashing red	Confirm selection
E	Start First Touch	Red	Function selected/processed
		Flashing red	Confirm selection
		Green	First Touch running

Tab. 4-9: LED states

Starting and opening the initial firmware opening the initial firmware

No system firmware is installed on the IndraControl XM upon delivery. Apply the supply voltage and the control starts the initial firmware. After the system firmware has been loaded successfully, the control does not boot the initial firmware anymore. To call the initial firmware again, press the "Reset" button when switching on the 24 V supply voltage until the LEDs DIA3 and DIA4 flash alternatingly in red. Subsequently, open one of the following functions via a menu structure:

Menu item no.	Function(s)	LED
1	Start First Touch	E
2	Start First Touch, delete network configuration	E, D
3	Start First Touch, delete network configuration, reset user name/password	E, D, BT
4	Start First Touch, delete network configuration, reset user name/password, Delete Files -> Factory Reset	E, D, BT, ERR

Tab. 4-10: Menu items of the initial firmware



If the function was executed successfully, the LED "E" changes from red to green after some seconds.

The operation mode switch SF1 provides the following options to open the menu items:

1. Press SF1 < 3 seconds: Menu item navigation and selection.

2. Press SF1 > 3 seconds: Triggering the menu item function.
3. To confirm the "Factory reset" (menu item no. 4), the operation mode switch SF1 has to be set to the "Run" position.



To select menu item 4 "Factory reset", all files installed since commissioning, are deleted on the IndraControl XM. We recommend to create a system firmware backup (see [chapter "Firmware management" on page 37](#)) before using this function. The initial firmware is part of the hardware and not deleted.

Installing backup from external SD card

If the control is as in delivery state, a backup can also be loaded to the control without PC.

Proceed as follows:

1. Copy a valid control backup to a Bosch Rexroth SD card XA-SD02 (part no. R911175017) for the XM4x. Use either a PC with SD card reader or the file manager of the First Touch commissioning tool.
2. Rename the image file to "system.fw".
3. Plug in the SD card to the base of the IndraControl XM labeled with "SD".
4. Switch on the 24 V supply voltage.
5. The initial firmware automatically starts loading the backup to the control. While loading, the "ERR" LED is lit in orange. After successful loading, the "ERR" LED is off and the "E" LED is lit in green.
6. Reboot to start the control with system firmware.

The installation can be facilitated together with the "Backup system firmware" function (see [chapter "Firmware management" on page 37](#)).

Licenses

The initial software for the IndraControl XM controls was developed under the following license declarations:

- Apache License Version 2.0, January 2004
- The GNU General Public License (GPL) Version 2, June 1991
- The MIT License
- OpenSSL License
- The zlib/libpng License

4.1.10 Booting

The control starts booting after switching on the 24 V voltage supply. Booting can be monitored and checked using different LEDs.

The following boot sequence is defined:

1. U_L LED at the 24 V plug is lit in green if 24 V are connected.
2. ON LED at the 24 V plug is lit in green if the complete programmable logic is properly loaded.
3. DIA3 and DIA4 LED in block of 10 flash alternatingly in red while booting.
4. The LEDs DIA3 and DIA4 LED in a block of 10 go off when booting is complete and the operating system is running.



If it is not booted in this sequence, an error is present.

4.1.11 Fallback and recovery

When the control starts, the internal, programmable function blocks are loaded. After successful loading, the ON LED is green.

If the ON LED is not lit in green, loading failed. In this case, the user has to initiate the "Fallback Mode". Therefore, press the "Reset" button when switching on the 24 V supply voltage until the ON LED is lit in green.

4.1.12 Selecting the start mode

General information

The startup mode determines the behavior of the MTX at startup. The startup mode changes are applied at next startup.

Selecting start mode

To enable the start mode menu during the start phase after the two red loading phase LEDs (DIA3 and DIA4) are off, press the switch SF1 into the switch position "Clear" until the boot stop is reached.

The following LEDs display this boot stop:

DIA1	Switched off
DIA2	
ERR	
BT	Switched off

Tab. 4-11: Boot stop display of the LEDs

Press switch SF1 for a second time into the switch position "Clear" until the start mode menu is enabled.

This is shown using the following LEDs

LED	Color
DIA1	Switched off
DIA2	
ERR	
BT	

Tab. 4-12: Startup menu display of the LEDs

If you release the switch, the LED DIA1 is on in the signal of the first startup mode (normal start) 

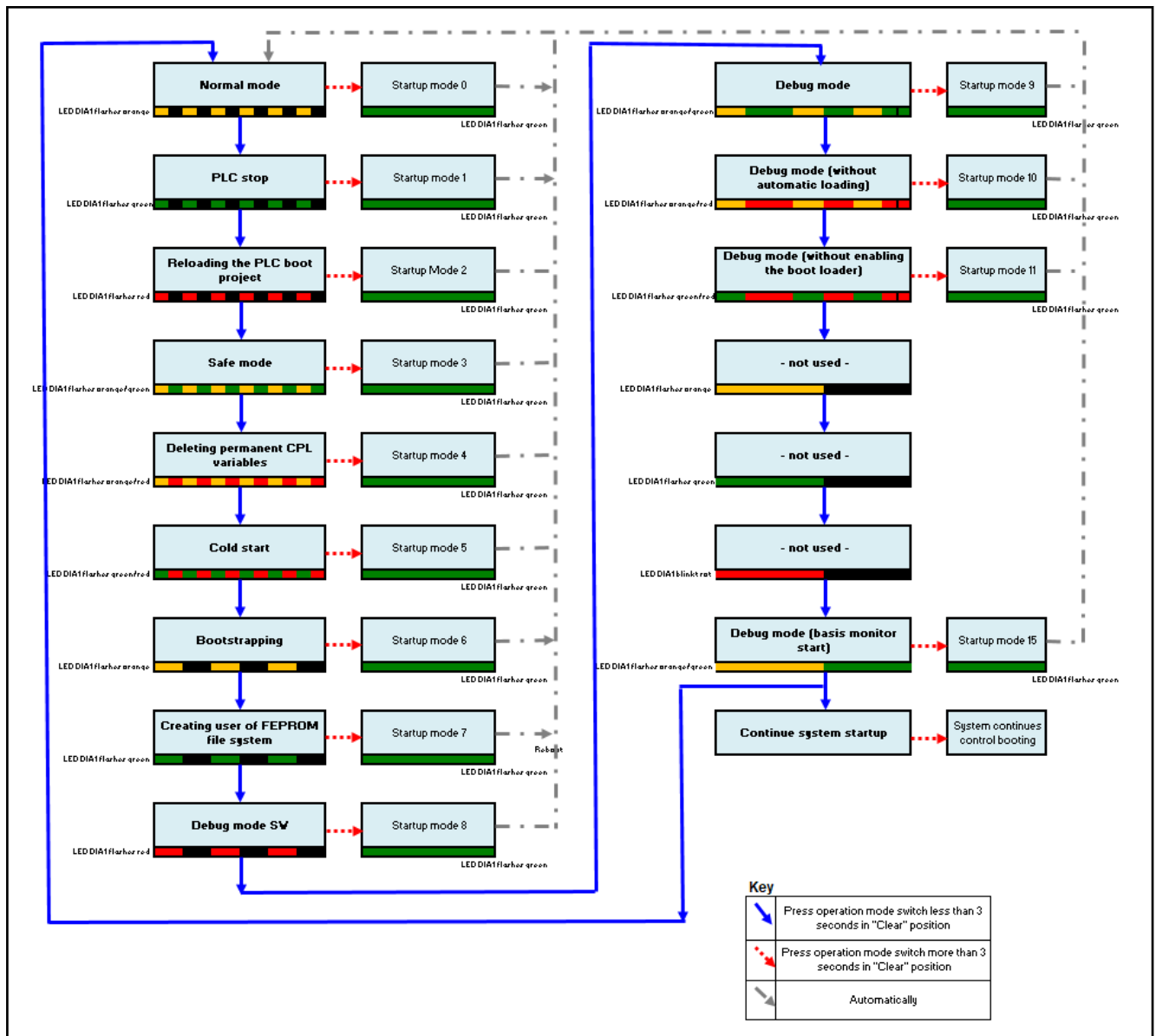


Fig. 4-4: Sequence of MTX boot menu

Startup mode	Function(s)	Color LED DIA 1
0	Normal mode All existing data and file systems are retained. The root file system is checked at startup. If a defective file system is detected, a critical system error is displayed. A new (empty) root file system is automatically created during the next startup	
1	PLC stop The behavior corresponds to Startup mode 0 with the difference that the PLC remains in the STOP state and the PLC user program is not processed	
2	Reload the PLC boot project The PLC boot project is loaded from the user FEPRM. Any PLC boot project that exists in the root file system is discarded. Otherwise, the behavior corresponds to the startup mode 0	

XM42 commissioning

Startup mode	Function(s)	Color LED DIA 1
3	Save startup In rare cases, due to faulty machine parameter specifications, a control startup can be impossible. Startup mode 3 is used to carry out a startup in this error situation irrespective of the set machine parameters. A startup with the minimum configuration is executed. The machine parameters set are ignored. After the startup, the invalid machine parameter settings can be corrected and a new startup in Startup mode 0 can be carried out	
4	Delete the permanent CPL variables The permanent CPL variables are deleted. Otherwise, the behavior corresponds to Startup mode 0	
5	Cold start The power-up management logic is not run through. Otherwise, the behavior corresponds to Startup mode 0.	
6	Bootstrap A new root file system is created. As a result, all old file system data is lost. If an intact user FEPROM file system exists, the PLC boot project and configuration data are loaded from there	
7	Creating the user FEPROM file system again The user FEPROM is created again. As a result, all the old file system data is lost. This is required, for example, if a user FEPROM file system is defective. The root file system is retained. The permanent CPL variables are deleted	
8	Debug mode SW	
9	Debug mode This is normally for debugging if the control does not automatically boot after a reset. After the basic monitor has been initialized, the boot loader is enabled and the sub-systems are automatically loaded	
10	Debug mode (without automatic loading) After the basic monitor has been initialized, the boot loader is enabled. Further loading can take place via TCP/IP	
11	Debug mode (without activating the boot loader) The basic monitor is initialized. Further loading can take place via TCP/IP	
12	- Not used -	
13	- Not used -	
14	- Not used -	
15	Debug mode (basic monitor start) Only the basic monitor is enabled.	

Tab. 4-13: Startup mode of the MTX

The operation mode switch SF1 provides the following options to open the menu items:

1. Press SF1 < 3 seconds: Menu item navigation and selection.
2. Press SF1 > 3 seconds: Triggering the menu item function.

Select the start mode. The startup then starts in the selected start mode.

**WARNING****Clearing data is impossible**

Only the technical personnel or the service may modify the start mode and only in case of emergency! Always ensure an available backup!

4.1.13 Runtime menu

To enable the Runtime menu, press SF1 down until the LED DIA1 flashes in orange (normal mode).

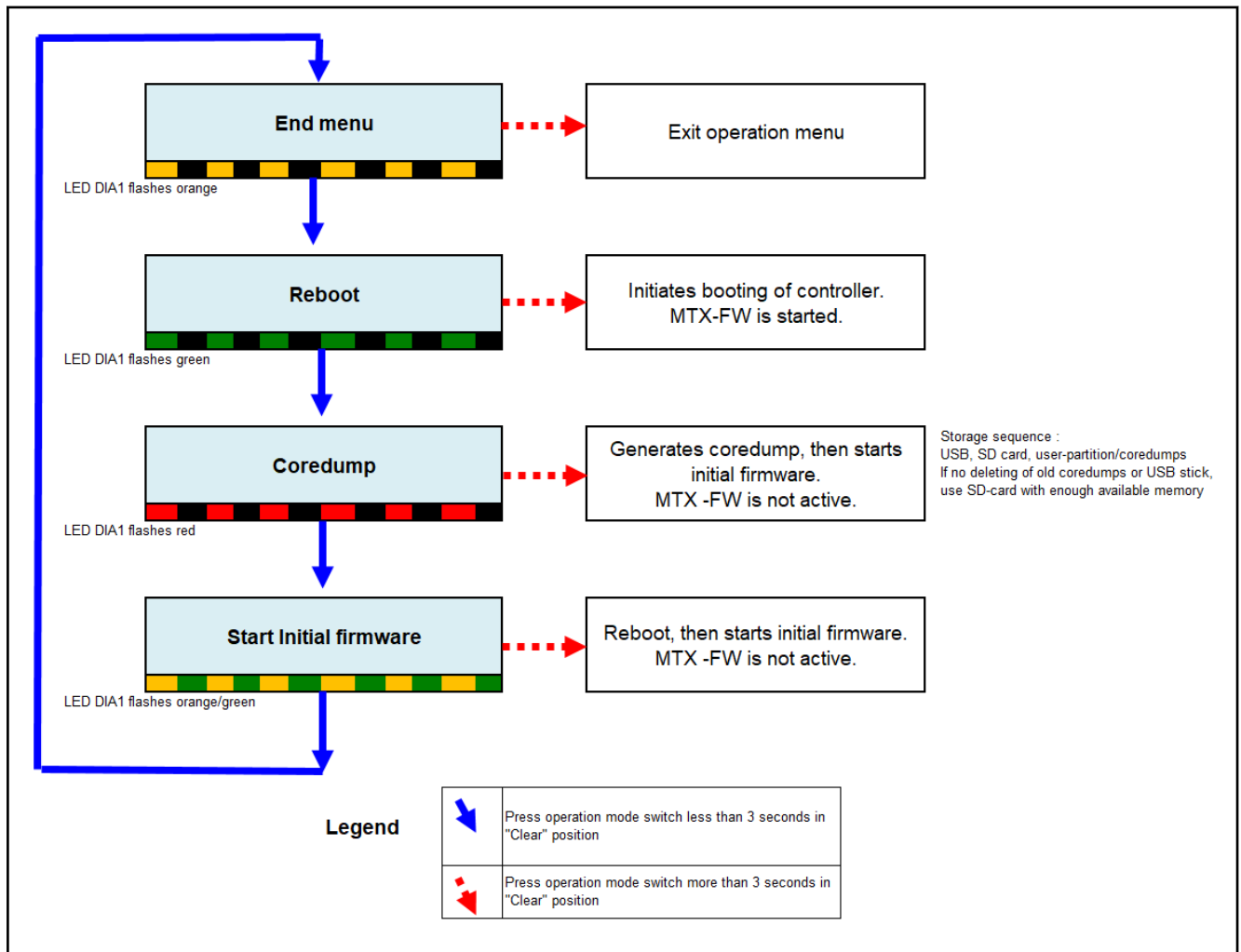


Fig. 4-5: Runtime menu

The operation mode switch SF1 provides the following options to open the menu items:

1. Press SF1 < 3 seconds: Menu item navigation and selection.
2. Press SF1 > 3 seconds: Triggering the menu item function.



The selected action is executed without prompt. A running NC program is closed immediately.

**WARNING****Machine damage is possible**

Only the technical personnel or the service may use the "Runtime" menu and only in case of emergency!

4.1.14 IndraControl First Touch

General information

The "IndraControl First Touch" commissioning tool interactively commissions and maintains controls of the MTX XM/MTX VPB generations. For the commissioning tool, go to the IP address (standard IP: 192.168.1.1) of the control. It runs on most of the current web browsers.



The "IndraControl First Touch" commissioning tool is only available in English.

Supported web browsers:

- Chrome 3+
- Firefox 9+
- Internet Explorer 10+

The standard URL of the IndraControl First Touch website is <https://192.168.1.1/>. The First Touch website communicates via the encrypted https protocol. Enter the URL and confirm the security certificate before the website is loaded.

After loading the First Touch website, a "Welcome" page is displayed. For all other functions or the navigation to another First Touch site, a user login is required. To log in, enter user name and password into both the form fields of the navigation bar.

The default user ID is:

- User name: boschrexroth
- Password: boschrexroth

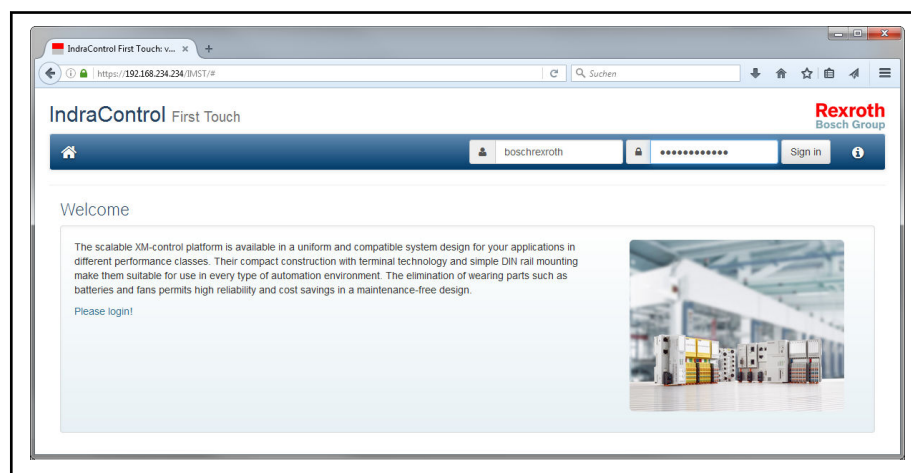


Fig. 4-6: "Welcome" dialog

Hardware data

The "Hardware" site lists all device-specific data of the MTX XM42 control in a table.

- Material index

- Hardware type plate
- Serial number
- Part number

Network settings

The "Network" site is used to configure the network settings for the XF5 port during the initial commissioning. The currently configured network settings are shown in the following three fields and can also be changed using these fields:

- IP address
- Gateway
- Subnet mask

Enter a valid network setting and write that setting to the control (press the **Write configuration** button). If writing the network settings is completed successfully, a message is displayed in the header area of the site. The **Reboot control** button is additionally displayed. It is used to restart the control and thus applies the network setting.

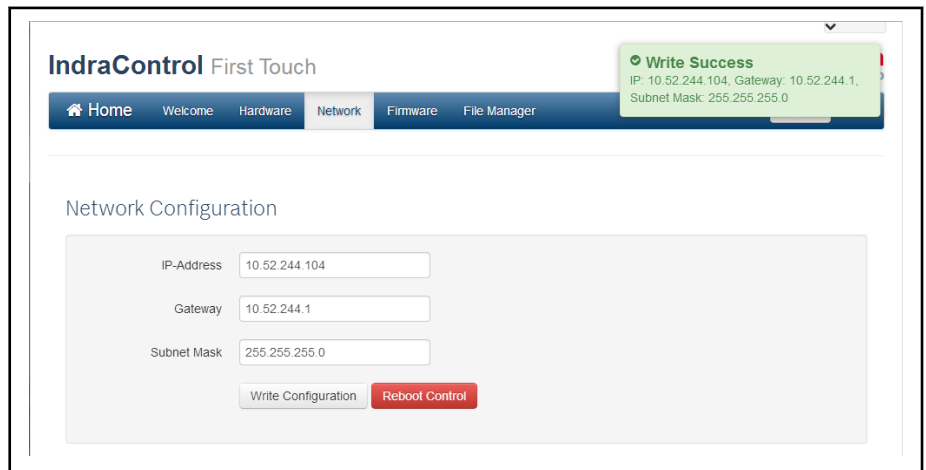


Fig. 4-7: "Network configuration" dialog



The network settings are only applied after a control restart.

Firmware management



The menu items are only available in this dialog if the initial firmware is active.

The firmware management provides backup and restore functionalities of the internal read-only memory of the control. This allows to move the partitions SYSTEM, OEM and USER to an archive that can subsequently also be restored via the dialog.



Basically, MTX firmware archives can also be loaded via the "Firmware management" dialog. In this use case, give priority to the IndraWorks firmware dialog (if available). When loading via the IndraWorks firmware dialog, the settings such as mount drives or time server settings are applied. These settings are lost upon the firmware download using the "IndraControl First Touch".

The "Firmware management" site provides the following information:

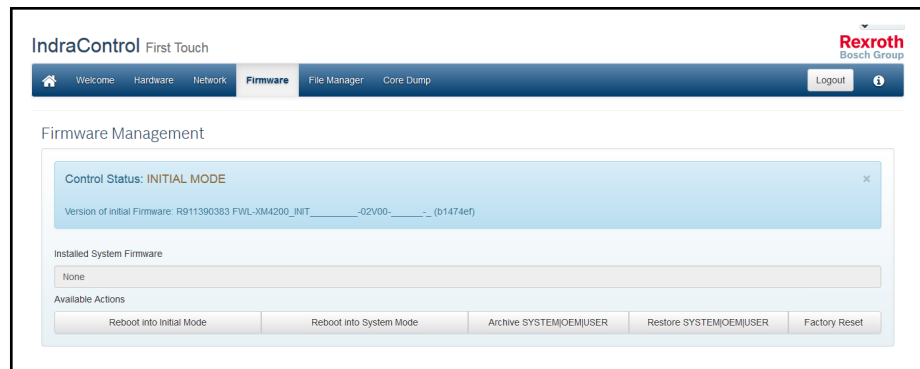


Fig. 4-8: "Firmware management" dialog

- Current control mode:
 - Initial mode: Initial firmware is loaded or active (initial firmware is shown in the message window)
 - System mode: Control was started with the previously installed system firmware
- Installed system firmware version ("none" if no system firmware was installed)

Press the respective function buttons for the following actions:

- Archive SYSTEM/OEM/USER (backup system firmware):
Backs up the complete directory structure, non-volatile data and the network settings to a backup archive (.fw file)
- Restore SYSTEM/OEM/USER (update system firmware):
Installs a previously generated backup from the local file system or from the USER partition of the control
- Reboot into System Mode:
Boots an installed system firmware
- Factory Reset:
Restores the delivery state of the control. The complete directory structure is reset



Attention: Loss of data!

An existing system firmware installation is deleted. The network settings are reset to its delivery state.

- Reboot Initial Mode:
Reboots the system and restarts the initial mode

Archive SYSTEM/OEM/USER (backup system firmware)

The function archives the internal storage medium including all partitions and their file and directory structure. Subsequently, an ".fw" file is stored on a previously selected memory medium. This file can be loaded to a local PC at a later point in time (see [chapter "File manager" on page 41](#)).



The file name of a system archive can consist of 4-16 alphanumeric characters and has to have the file extension ".fw".

The backup can only be started if the selected file name corresponds to the archive rules. During as well as after the backup, a corresponding message is displayed.

Fig. 4-9: Dialog "Archive SYSTEM/OEM/USER"

Fig. 4-10: Dialog "Archive SYSTEM/OEM/USER" – Completed successfully

Restore SYSTEM/OEM/USER (update/install system firmware)

If the action "Restore SYSTEM/OEM/USER" is selected, a modal dialog opens to select a system firmware file. The file can either be selected from the local file system or the USER partition of the control.

If the file is selected from the local file system, the system firmware is installed in two steps:

- Transferring the file to the control
- Installing the firmware

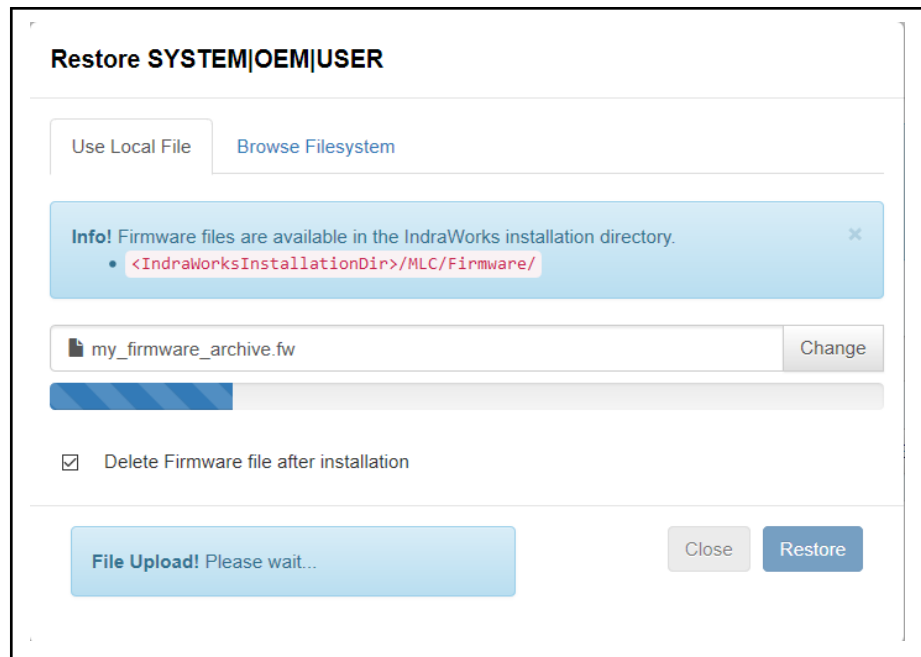


Fig. 4-11: Dialog "Restore SYSTEM/OEM/USER" – Upload

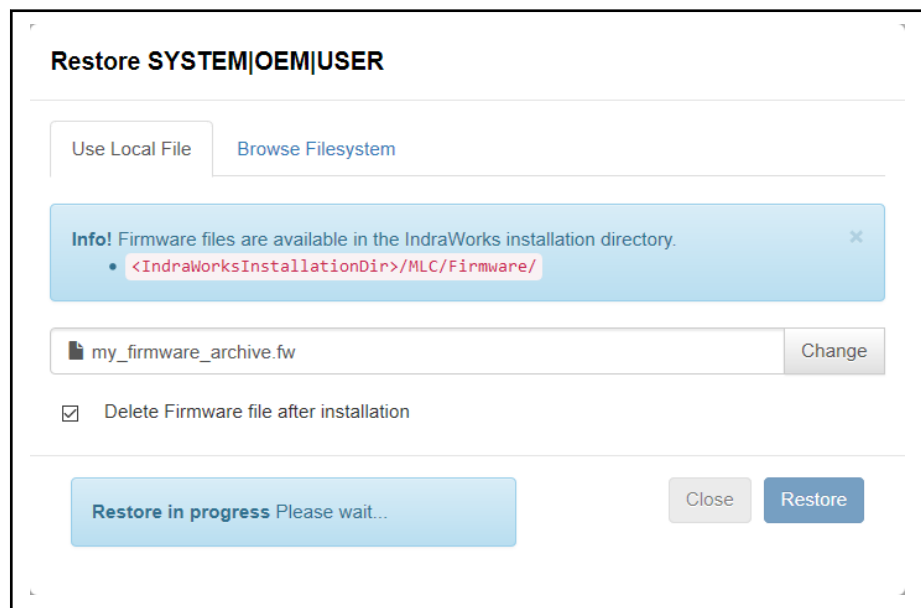


Fig. 4-12: Dialog "Restore SYSTEM/OEM/USER" – Installation running

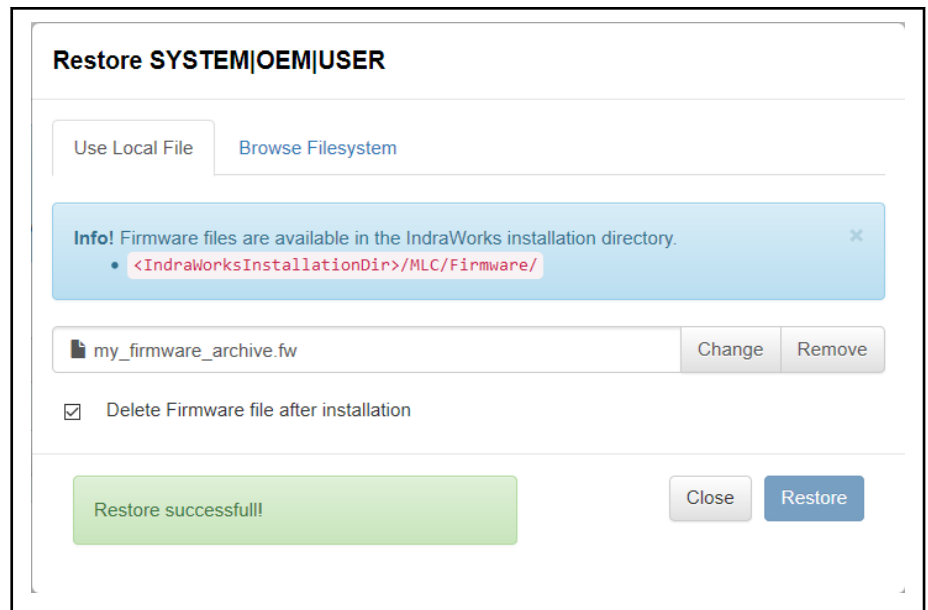


Fig. 4-13: Dialog "Restore SYSTEM/OEM/USER" – Completed successfully

If the archive file is selected from the USER partition, it is directly installed after pressing "Install". The table lists all firmware files found in the root directory of the USER partition. Select the firmware file to be installed by selecting the corresponding table row.

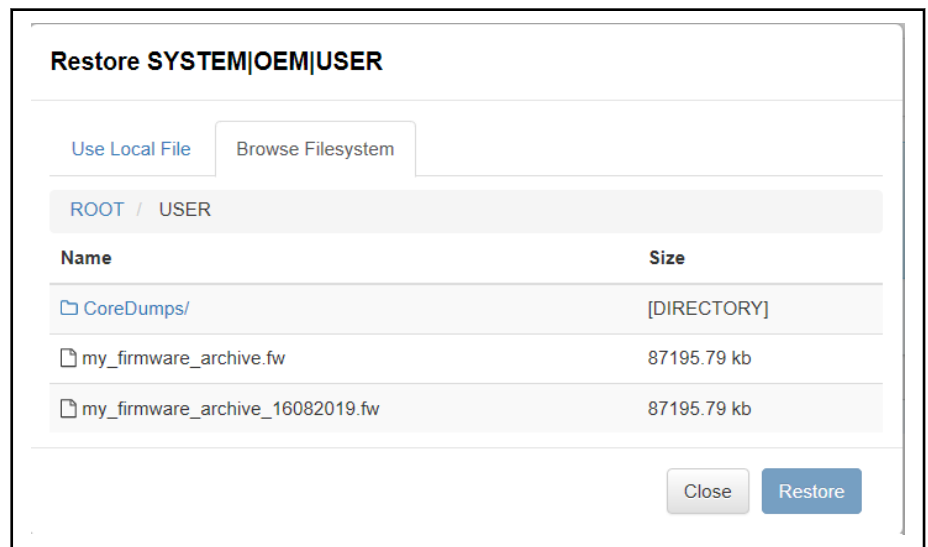


Fig. 4-14: "Install firmware" dialog – From USER partition

File manager

The file manager dialog is used:

- To browse through the directory structure (click on a directory name)
- To transfer files from the control to the client (click on a file name)
- To transfer files from the client to the control (action: Upload file)
- To create directories (action: Create directory)
- To delete (empty) directories (click on the "Delete" icon in the table row)
- To delete files (click on the "Delete" icon in the table row)

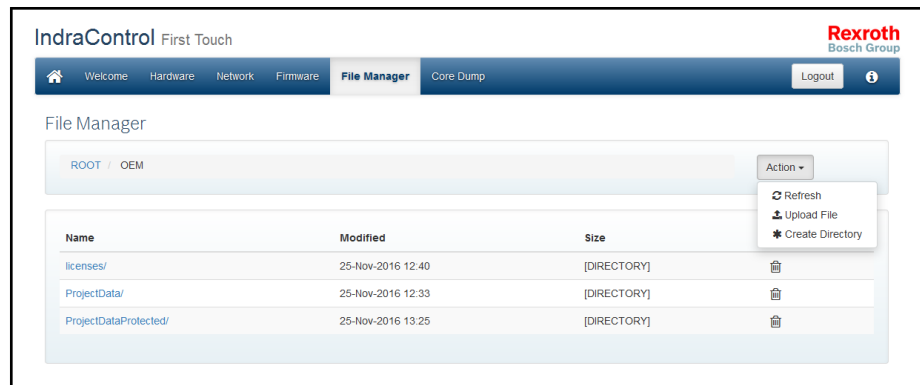


Fig. 4-15: "File manager" dialog

Managing a core dump

In case of a system error, the control can create a complete core dump of the system that can be provided for an error analysis. IndraControl First Touch supports two use cases with this functionality:

1. Transfer of the active core dump from the control to the user PC.

All files assigned to the active core dump are displayed with name and file size in the "Core dump" menu item. Click on the file name to transfer the files to the user PC via the browser-specific download dialog.

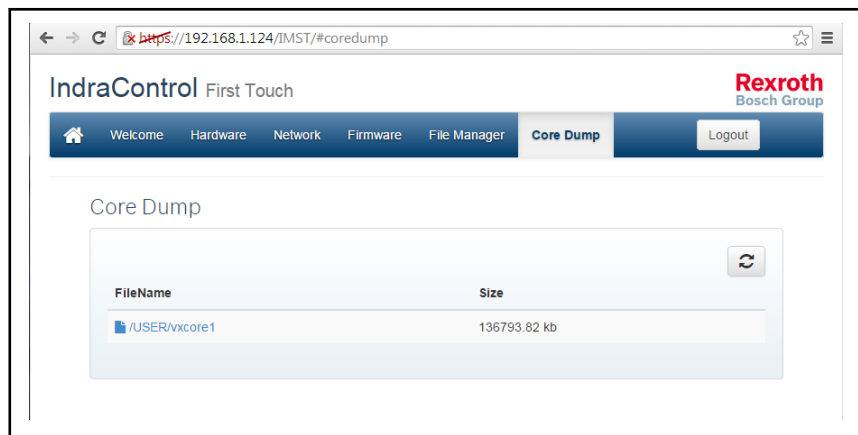


Fig. 4-16: Displaying the active core dump in IndraControl First Touch

2. Support of the user if the core dump could not be saved in the file system of the control (e.g. not sufficient free memory space available).

The user can now free up storage space in the default target directory for core dump files (/USER/). Use the "File manager" menu item. An additional data carrier (e.g. USB stick) can also be connected to the control. In both cases, saving the core dump has to be restarted. The IndraControl First Touch provides a dialog in the "Core dump" menu item. This dialog allows to select the target directory if unsaved core dump files are available in the RAM of the control.

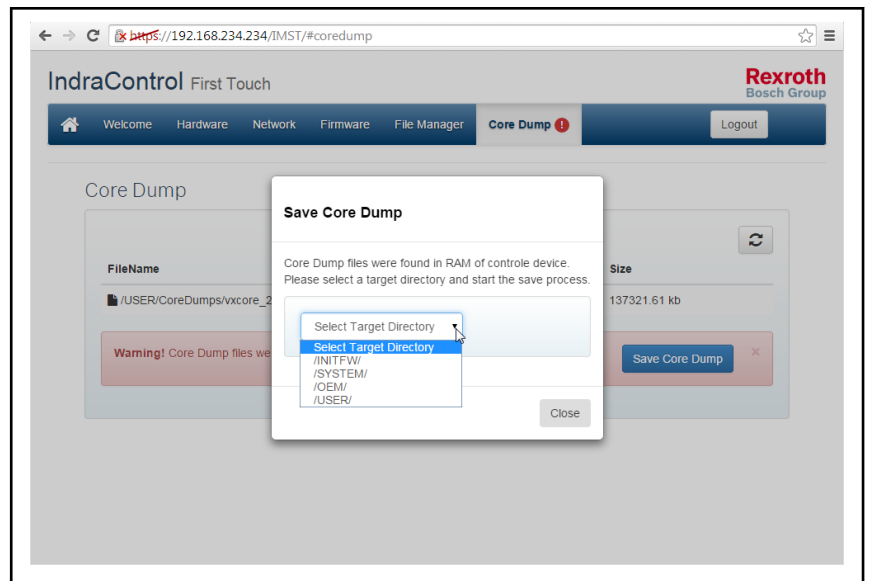


Fig. 4-17: Selecting the target directory to save the core dump

After selecting a valid target directory, press **Save** to save the core dump. If saving was successful, a corresponding message is shown. The dialog can now be closed. After closing the dialog, the core dump is displayed with its respective data.

The displayed sized corresponds to the size of an uncompressed core dump. The core dumps in the MTX control are zipped. The actual size is only displayed after saving. Generally, the compressed core dump is smaller than 250 MB.

After completion, the exclamation mark behind the "Core dump" tab disappears and the actual size is displayed.

5 General information

5.1 Hardware



Each hardware supported by the "MTX" system is described in detail in the "System Description" manual.

A typical MTX system consists of the following components:

- One or multiple controls (e.g. XM42-P) for an open-loop and closed-loop control of a tool machine
- One or multiple operating stations (e.g. VPP40) to operate and monitor a tool machine
- One or multiple field buses to connect status and control signals of the tool machine with the control
- Multiple drive control units and motors to move the traversing axes of the tool machine in a controlled motion

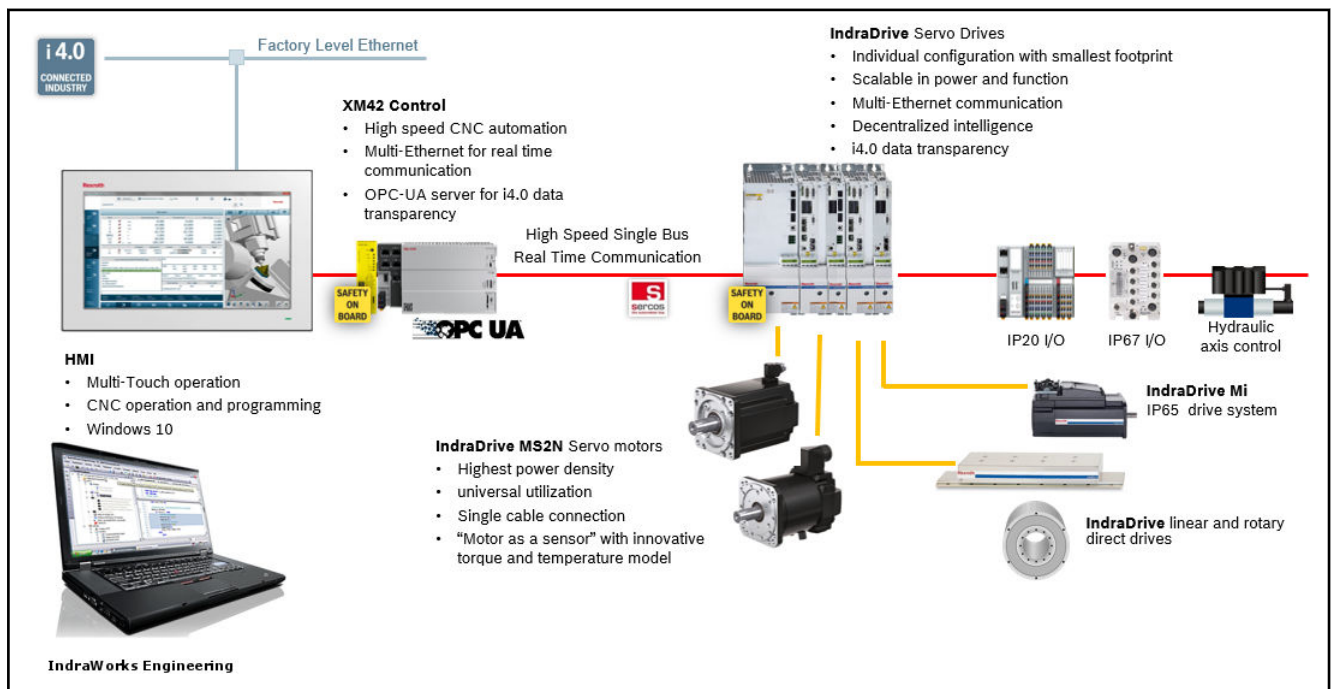


Fig. 5-1: Typical MTX system configuration

5.2 IndraWorks Operation Desktop

5.2.1 General information



All basic functions as well as the basic handling of IndraWorks Operation is described in detail in the "IndraWorks HMI" manual.

The MTX-specific topics for setting up and operate IndraWorks Operation in the documentation "Standard NC operation".

For information on touch devices, refer to the MTX 15VRS Multi-touch (DOK-MTX***-MULTI***V15-APRS-EN-P).

General information

The Operation Desktop is the main tool when working with the "MTX" control system. The status displays (position, velocity, override, etc.) of the drives are summarized in the NC screen.

The Operation Desktop comprises the machine operation keys (M-keys), the function keys (F-keys) and the OP-keys to switch the individual operating functions (tool management, programming screen, NC screen, channel over-view, etc.). The machine operation keys affect the PLC or the interface so that PLC functions or switching functions of the interface can be triggered. "WinStudio" provide user-defined screens.

Using machine parameters, the view of the NC screen can be configured user-specifically.

The screenshot displays the Rexroth MTX Operation Desktop interface. The top bar shows the channel status as '\$1 Channel 1' with 'Automatic Contin. Block' and 'inactive' modes. The date and time are '4.3.2006 | 11:47:00 AM'. The main area contains a table of drive positions and a table of machine parameters.

WCS			Command	End position	Dist. to go	Program
X (X1)		mm	10.0000	10.0000	0.0000	0.0000
Y (Y1)		mm	0.0000	0.0000	0.0000	0.0000
Z (Z1)		mm	0.0000	0.0000	0.0000	0.0000
W (W1)		mm	0.0000	0.0000	0.0000	0.0000
C (C1)		mm	0.0000	0.0000	0.0000	0.0000
S (S1)		mm	0.0000	0.0000	0.0000	0.0000

		Program	Command	Actual	Override	Max	Gear
F		mm/min	0.0	0.0	0.0	70%	120%
S1		1/min	0.0	0.0	0.0	0%	120%

No program selected

G01	G90	G94	G8
G62	G40	G43	G45
OVE			
G53.1	G53.2	G53.3	G53.4
/database/Z01.zot			
Tool-active		Curr. edge	D-Number
ED0		D0	
/database/DC1.dct			

NC Screen

Program Selection F2 Active Program F3 Overview Act. Corrections F4 Active ZP Offsets F5 Active D-Corrections F6 Active Variables F7 >> F8 F9

Prepare Machine Program Tool Management System Production Data Maintenance Diagnostics

Fig. 5-2: Example of the Operation Desktop

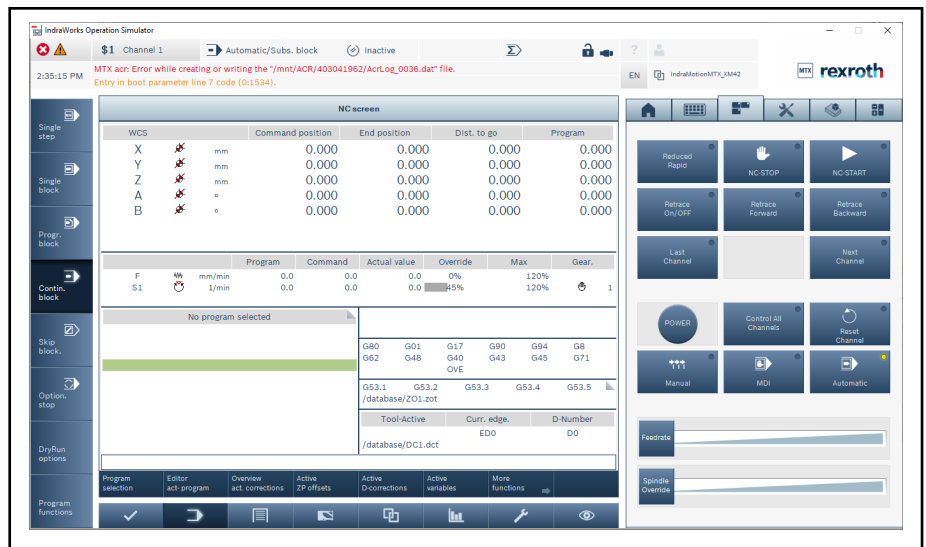


Fig. 5-3: Example of the touch operation desktop

5.3 IndraWorks - Engineering

5.3.1 General information



All basic functions as well as the basic handling of IndraWorks Engineering are described in detail in the "IndraWorks HMI" manual.

The Bosch Rexroth Engineering Tools integrate themselves into the carrier system IndraWorks.

Comprehensive functions such as project navigation and project administration as well as the creation of project and configuration data are centrally executed.

All tools for the project planning, commissioning, maintenance and diagnostics of a tool machine are provided with the Engineering Desktop:

- Creating, archiving and restoring IndraWorks projects
- Changing parameters in control and drive
- Editing PLC data
- Editing HMI data
- Exporting/importing
- Version control of the IndraWorks project
- License management
- Language management
- etc.

5.3.2 Creating a project

General information

To create a new project, go to **File ► New ► Project**. The "Create new project" dialog opens.

Two selections are at least possible:

- General templates

- Basic projects
- My project templates



If there are no projects, use an example project.

General templates

Create a project. Then, create a control and an HMI device in the project.

Basic projects

The basic project is a Bosch Rexroth created example project that can be extended at any time. All important components are available. Any devices can be used. For more information on the project, refer to the project node "Documentation".

The basic project can be extended by own components (e.g. visualization device VPP40).

My project templates

Own project templates are only displayed if an own project template was created from an existing project. Refer to the IndraWorks 15VRS Engineering documentation on how to create own project templates (DOK-IWORKS-ENGINEE*V15-APRS-EN-P).

5.3.3 Compatibility

General information

MTX IndraWorks projects support the version and release compatibility. This means that - within the product family (XM42) - the current "IndraWorks Engineering" version can be used to edit all projects created with an older version.

Each project is assigned a version ID informing the user which IndraWorks installation is required to edit the project.

If a project with an older version ID is opened, it is not automatically converted. Instead, the project is operated in the so-called compatibility mode. This means that this project can also be edited for the older IndraWorks version in case of changes made to the project. New functions causing an incompatible project change are not available.



Projects with CML controls can be converted into projects with XM42. There is no automatic conversion!

See [chapter 9 "Upgrading the CMLx5 control to the XM42 control"](#) on page 429



Projects created with IndraWorks Engineering 12VRS or older have to be converted into 14VRS. It can then be converted into an XM42 project.

The compatibility mode can now be explicitly modified. The project compatibility to the original version is lost.



For more detailed information, see the "IndraWorks Engineering" documentation.

Compatibility mode

The compatibility mode specifies for an IndraWorks project which "IndraWorks Engineering" version is required to edit the project.

The compatibility mode can be selected when creating a project or at a later point via the property dialog of the project. It defines the possible functional scope to edit the project in the selected version.

Device version

Controls and visualization devices in the IndraWorks project have their own versionings. These versionings define the functional scope available for a device in the selected version. The device version is always smaller or equal to the compatibility mode of the project.

The device version can be selected when creating a project or at a later point via the property dialog of the device.

More information about "MTX" can be found in the chapter about the commissioning of the control of this documentation ([chapter 7 "Control commissioning" on page 141](#)).

5.3.4 User management

General information

A user management to protect the project data from unauthorized changes is available in "IndraWorks Engineering".

A second user management allows to limit the access to sensitive machine operation areas in the "IndraWorks Operation".

Furthermore, the access to data block stored on the control can be limited.

Access protection for the project planning

To protect project data from unauthorized change, create users and user group and assign right to the users and the user groups in "IndraWorks Engineering".

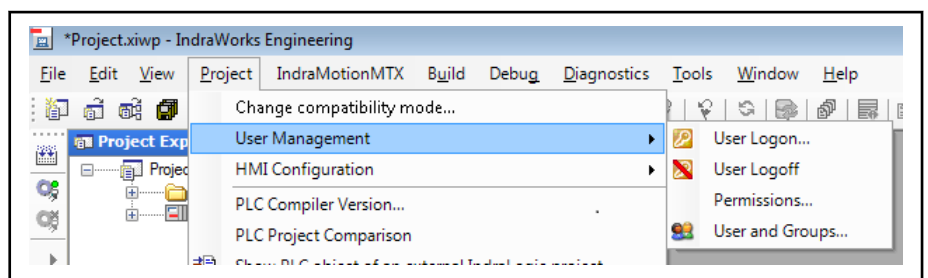


Fig. 5-4: Calling the user management to protect the project data



Details about the project data protection can be found in the "IndraLogic 2G PLC Programming System" documentation in the "IndraWorks user management" chapter.

Access protection for the project planning

To facilitate access to selected functions or areas of the user interface in "IndraWorks Operation" for defined individuals, these functions or areas can be assigned rights via a separate user management.

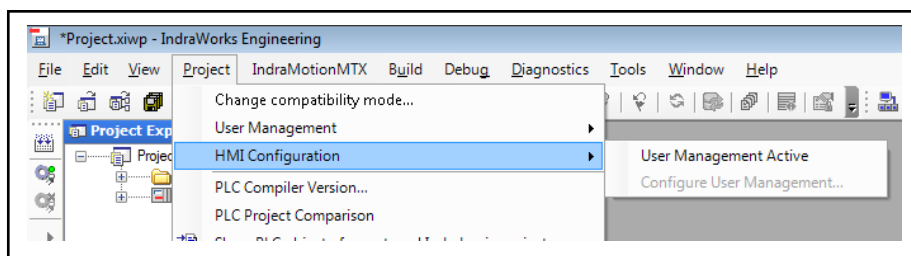


Fig. 5-5: Calling the user management for machine operation protection



Details about the access protections during machine operation can be found in the "Engineering" documentation in the "IndraWorks user management" chapter.

Access protection for data on the control

Data elements on the MTX control can be assigned user levels. During the scheme definition, data blocks are assigned read or write access by entering the attribute definitions (L1, L2, L3, L4, L5) in the data block scheme. When accessing the data block at a later point, the registered user has to have the corresponding level (L1...L5).

Which functions can be executed by the user are assigned via the group to which the user is assigned. The user receives the privileges which are determined for this group.

The following 8 standard groups of the user management are already assigned to user levels:

- Operator: L1
- Setter: L1, L2
- Maintenance: L1, L2
- Commissioning: L1, L2, L3
- Machine manufacturer: L1, L2
- Service: L1, L2, L3, L4
- Engineer: L1, L2, L3, L4, L5
- Administrator: L1, L2, L3, L4, L5

For standard groups, these functions can be displayed via "Privileges" but cannot be modified.

With this dialog, the allowed functions are determined for the freely configurable groups. This is only allowed for a member or standard group "Administrator".

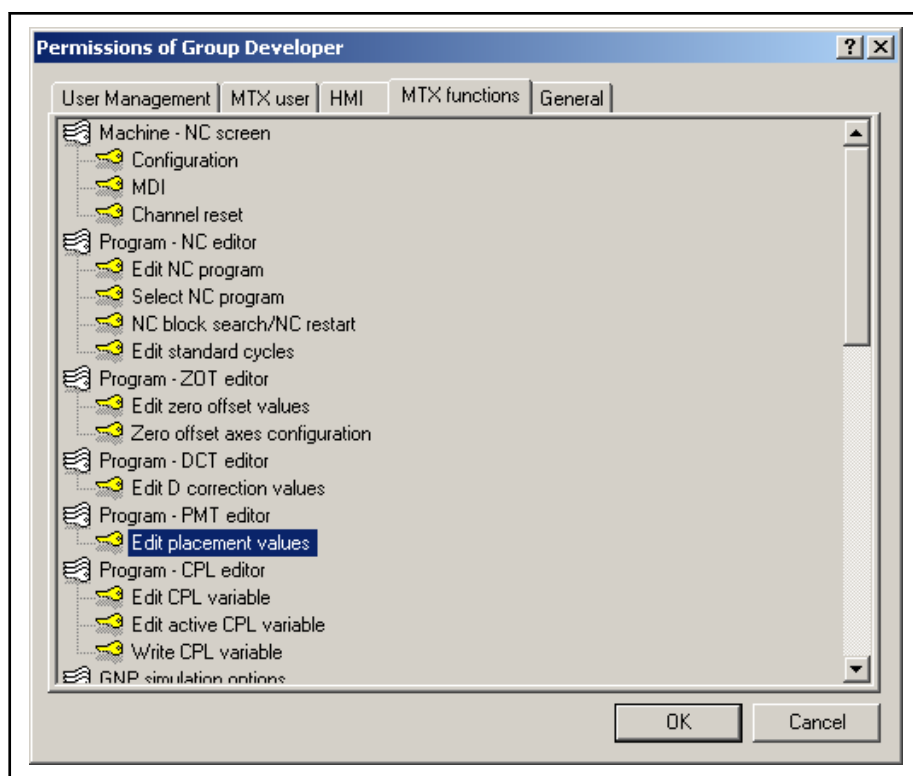


Fig. 5-6: Editing privileges/functions

If the user has the privilege "X", the users which are a member of this group can execute the function "X".

Context	Group	Function	L1	L2	L3	L4	L5	L6	L7	L8
Machine	NC screen	Configuration			x	x	x	x	x	x
		MDI		x	x	x	x	x	x	x
		ChannelReset		x	x	x	x	x	x	x
Program	NC editor	EditNpg		x	x	x	x	x	x	x
		SelectNpg	x	x	x	x	x	x	x	x
		SearchBlock		x	x	x	x	x	x	x
		EditDefaultCycles		x	x	x	x	x	x	x

Context	Group	Function	L1	L2	L3	L4	L5	L6	L7	L8
Program	ZOT editor	Edit		x	x	x	x	x	x	x
		Configuration			x	x	x	x	x	x
	DCT editor	Edit		x	x	x	x	x	x	x
	PMT editor	Edit		x	x	x	x	x	x	x
	MTX Navigator	Configuration		x	x	x	x	x	x	x
		DeleteFile		x	x	x	x	x	x	x
		RenameFile		x	x	x	x	x	x	x
		PasteFile		x	x	x	x	x	x	x
		NewFile		x	x	x	x	x	x	x
		EditFileOptions		x	x	x	x	x	x	x
	CPL editor	Edit		x	x	x	x	x	x	x
		EditActive		x	x	x	x	x	x	x
		WriteValue		x	x	x	x	x	x	x
Tools	General	Import		x	x	x	x	x	x	x
		Export	x	x	x	x	x	x	x	x
		IncrementalInput	x	x	x	x	x	x	x	x
		AbsoluteInput		x	x	x	x	x	x	x
		ToolInsert		x	x	x	x	x	x	x
		ToolDelete		x	x	x	x	x	x	x
		ToolMove		x	x	x	x	x	x	x
		ToolPos	x	x	x	x	x	x	x	x
		ListConfiguration			x	x	x	x	x	x
	Tool Editor	Invoke	x	x	x	x	x	x	x	x
		Configuration			x	x	x	x	x	x
IMT	General	Invoke	x	x	x	x	x	x	x	x
	MTX Configuration	Invoke			x	x	x	x	x	x

Tab. 5-1: Editing privileges/functions

5.3.5 Version control

General information



The basic handling of the version control is described in detail in the "IndraWorks Engineering" documentation.

IndraWorks Engineering allows working with version-controlled projects located on one team server. Multiple IndraWorks users can work simultaneously on such version-controlled projects.

The following MTX controls support the version control:

- MTX XM42 standard
- MTX XM42 performance
- MTX XM42 advanced

MTX project data

Version-controlled project data

The following project data is version-controlled:

- PLC project data (POUs, variable definitions, I/O configuration, function modules, etc.)
- HMI project data (screens, M- and F-keys, WinStudio, etc.)
- Drive data (offline parameterization MLD)

Non-version-controlled project data

The following data is not version-controlled:

- NC project data (machine parameters, NC program, system data, etc.)
- Virtual operating field/simulation
- Communication settings of the control

5.3.6 Archiving and restoring MTX projects

General information

An IndraWorks project consists of several parts:

- HMI data (HMI screen, F-keys, M-keys,...)
- Control data (machine data, extended data, drive data, etc.)
- PLC project

Data backup includes all data and settings generated by the machine tool manufacturer and the end user. The drive parameters (Sercos parameters) are an exception. These have to be saved before a project data backup to ensure that the parameters are recorded in the project tree. The drive parameters are then backed up with the project backup.



Archiving and restoring data of an IndraWorks project is described in detail in the "IndraWorks Engineering" manual.

In contrast to that documentation, the control data (see [chapter "Control data" on page 56](#)) is backed up during the device data backup for the "MTX".

IndraWorks project (complete data backup)

Handling instruction: Data backup of the IndraWorks project (complete data backup)

This present handling instruction describes the data backup of the MTX control data ranges.

IndraWorks Engineering / project: Archiving

1. Start the archiving function in the Engineering Desktop under the directory in the main menu "Project".
2. Follow the instructions of the wizard.

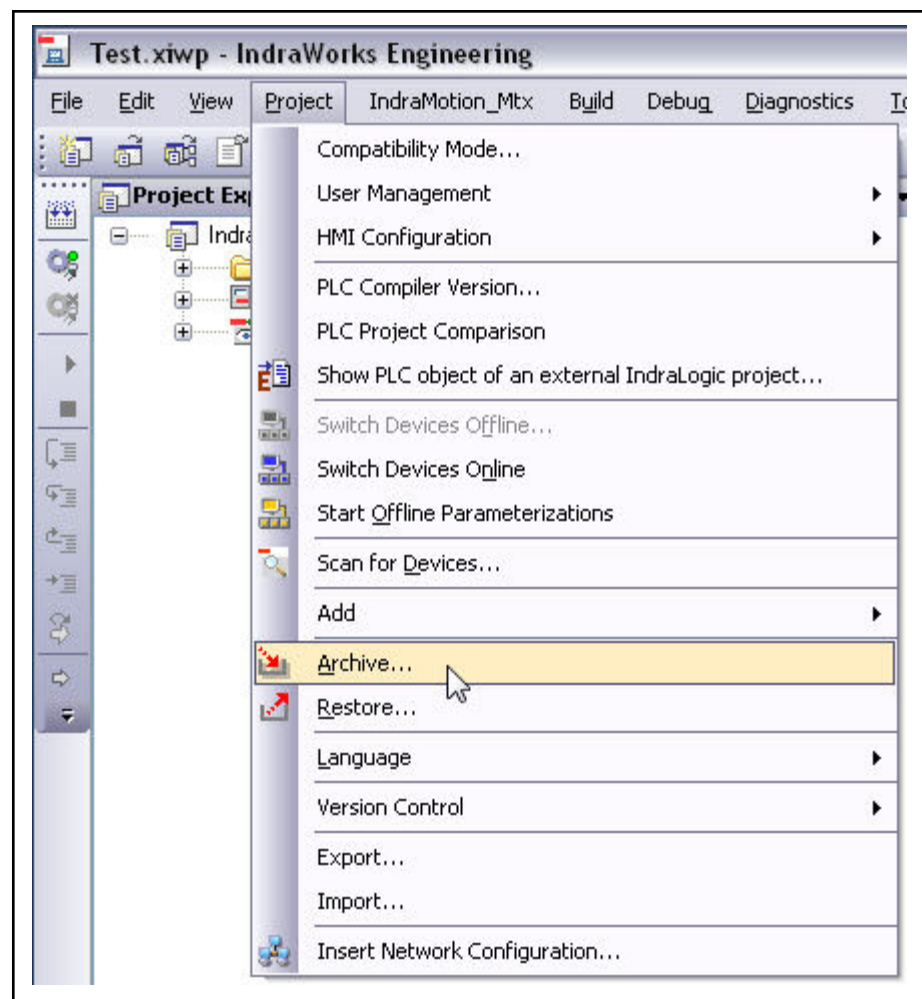


Fig. 5-7: Menu: Archiving a project



In this data backup variant, the following data is backed up:

- Device data (control data, extended data, drive data)
- PLC project
- HMI data

These areas cover the entire scope of control data. All data is backed up here.

If an HMI project is to be backed up, individual device data types can be deselected. This archive only contains the HMI components of the project.

Documentation:	IndraWorks Engineering	Archiving and restoring projects

Handling instruction Restoring IndraWorks project data

This present handling instruction describes the restoration of IndraWorks project data.

IW Engineering / project: Restore

1. Start the function "Restore" in Engineering Desktop under the directory in the main menu "Project".
2. Follow the instructions of the wizard.

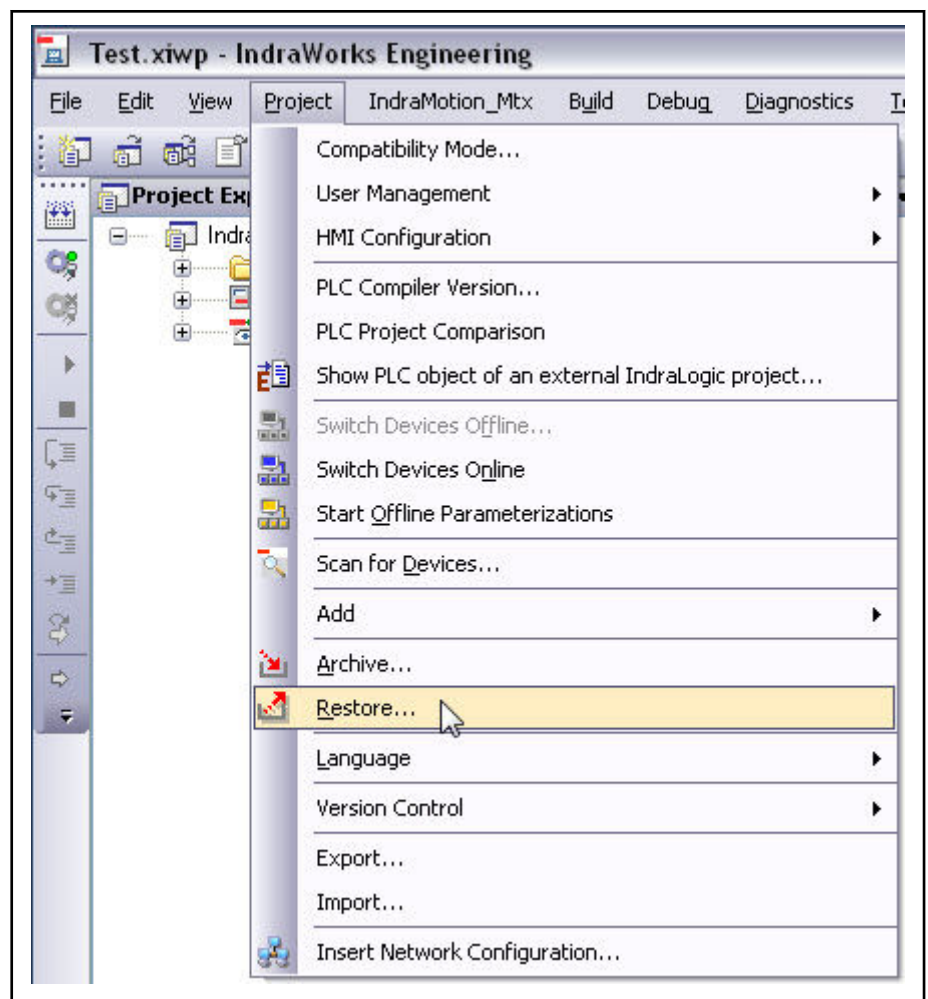


Fig. 5-8: Menu: Restoring a project



The device data has to be restored separately following the instruction "Restoring control data".

Documentation:	IndraWorks Engineering	Archiving and restoring projects

Device data

The device data contains all data of a device (control) that can be archived.

- Control data
- Extended data (mount directories, user partition, user-defined directories)
- Drive data

Archiving and restoration can be started as follows (multiple options):

IndraWorks Engineering

Project node ► Archive / Restore

MTX project node ► Archive / Restore

A wizard opens to support in the following actions.

IndraWorks Operation

OP8 - Maintenance ► More functions ► Archive device data...

Control data

General information

Control data can be stored as archive with the file extension "*.tar".

Archiving control data - User input

Archive Specify name and storage location of the archive to be created. Click on the button on the right of the archiving path to enter the name of a new archive file.



The name of the archive file has to comply with the Windows® convention and must not be longer than 29 characters (excluding the file path).

Last archives Optionally, an existing archive file from the "last archives" list can be overwritten by selecting this archive file from the list. If an archiving path is already provided, it is additionally to be found as the first entry in the list to return to this default path at any time.

Contents Select the content to be stored in the archive. For a complete backup, select all checkboxes.

The following options can be selected:

- Machine parameters
- System data
- Tool tables
- User FEPROM file system
- RAM file system
- Permanent CPL variables
- Remanent PLC data

After completing the entries, press "Next".

Settings

- Switch PLC to STOP state before
- Switch PLC to START state afterwards



To select the "Non-volatile PLC data" option, switch the PLC to the STOP state to avoid inconsistent states in the archive.

Error messages If an error occurred in a file, this file is marked as faulty.

Refer to the following table for error numbers and causes:

No.	Error message	Cause	Recovery
1	File not processed (yet)!	Internal	Contact service
2	Error while calling the stat() function!	Internal	Refer to no. 1
3	File or directory not processed!	Internal	Refer to no. 1
4	Directory could not be opened!	Internal	Refer to no. 1
5	Error reading the file!	Internal	Refer to no. 1
6	Access to file impossible!	Read-protected file	Change file protection
7	File already exists!	Internal	Refer to no. 1
8	File not available!	Internal	Refer to no. 1

No.	Error message	Cause	Recovery
9	Memory not available!	Memory in the control is full	Delete unnecessary files
10	Read-only file system!	Internal	Refer to no. 1
11	Protection different!	Internal	Refer to no. 1
12	Size different!	Internal	Refer to no. 1
13	Date different!	Internal	Refer to no. 1
14	File data different!	Internal	Refer to no. 1
15	Other differences!	Internal	Refer to no. 1
16	Date/protection not restored!	Internal	Refer to no. 1
17	Error while creating a directory!	Internal	Refer to no. 1
18	Unknown error!	Internal	Refer to no. 1
19	Error when applying machine data!	Internal	Refer to no. 1
20	Error while applying tool table!	Internal	Refer to no. 1
21	File destroyed!	A file to be archived was changed during archiving. This can be caused e.g. by a CPL program	Do not change files during archiving; e.g. deselect all CPL programs
22	Write protection canceled!	A file in the archive is write-protected. This write protection is canceled after the file restoration and has to be set by the user if desired	This is a message for the user to decide what to do about file protection
23	Error while applying remanent PLC data!	Internal	Refer to no. 1
24	Error while applying remanent PLC data!	Internal	Refer to no. 1
25	Error while applying CPL variables!	Internal	Refer to no. 1
26	RAM file system is not available in archive!	The partial archive is not in the archive	Select another archive or change the selection of the partial archives in the dialog if desired
27	User FEPROM not available in archive!	Refer to no. 26	Refer to no. 26
28	Machine data not available in archive!	Refer to no. 26	Refer to no. 26
29	Tool table not available in archive!	Refer to no. 26	Refer to no. 26
30	Remanent PLC data not available in archive!	Refer to no. 26	Refer to no. 26
31	Remanent PLC data not available in archive!	Refer to no. 26	Refer to no. 26
32	Permanent CPL variables not available in archive!	Refer to no. 26	Refer to no. 26
33	PLC not in STOP state!	Internal	Refer to no. 1
34	Error while applying system data!	Internal	Refer to no. 1
35	System data not available in archive!	Internal	Refer to no. 1

No.	Error message	Cause	Recovery
36	Data contains invalid data block K1/K2!	Backup file contains data block that is not in the tool table	Refer to no. 1
37	Data does not correspond to data block schema!	Data block structure from backup file does not correspond to data block structure in tool table	Refer to no. 1

Tab. 5-2: Error messages

Restoring Control Data - User Input

Archive Click on the button to the right of the archiving path to enter name and storage location of the archive file to be restored.



Loss of data!

Any existing files are overwritten during the restoration. Save your files by archiving before executing the restoration.



The name of the file path must not contain more than 29 characters.

Last archives Optionally, select an existing archive from the "Last archives" list.

Contents Select the content to be restored.

The following contents can be selected:

- Machine parameters
- System data
- Tool tables
- User FEPROM file system
- RAM file system
- Permanent CPL variables
- Remanent PLC data

After completing the entries, press <Next>.

- Settings**
- Switch PLC to STOP state before
 - Switch PLC to START state afterwards

Execute NC restart The data of the control data archive is only applied after an NC restart. Select the checkbox to execute an NC restart after the restoration of the archive.

Clearing RAM file system The target directories can be cleared before the archive is restored. Select the checkbox if the RAM file system is to be cleared. This option can only be selected if the archive contains the RAM file system (at least one file) and if "RAM file system" was selected.



Files which are not in the archive are lost!

Clearing user FEPROM file system

The target directories can be cleared before the archive is restored. Select the checkbox if the USRFEP file system is to be cleared. This option can only be selected if the archive contains the FEPROM file system (at least one file) and if "User FEPROM file system" was selected.



Files which are not in the archive are lost!



If all checkboxes are selected, **only** the archive data is stored in the control after the restoration.

If there is no PLC program in the archive that can be run, it is to be compiled again and to be loaded after completion.

After completing the entries, press **Next**.

Archiving advanced data

General information

Extended archiving allows any directory and file of the control computer to be created and restored as an archive with the file extension "*.zip". Thus, it is for example possible to save and restore the mount directories belonging to an MTX, the user partition of the Compact Flash medium (if available) as well as all non-control-based peripheral data.

Extended archiving can be accessed in IndraWorks Engineering via the context menu of the MTX device node

Archiving advanced data - User input

Archive

Specify name and storage location of the archive to be created.

Click on the button on the right of the archiving path to enter the name of a new archive file.

Contents

All locally mounted directories of the MTX are provided.

The user partition of the Compact Flash card (if available) is also provided.



The user partition can only be backed up and restored completely. Selecting single files is not possible.

Additionally, it is possible to add and remove up to 99 user-defined folders. Thus, select the buttons "Add" or "Remove".

Permitted are:

- Local folders
(Example: "C:\MyDocuments\MyMTXDocuments")
- UNC network shares as \\HOST\SHARE
(E.g.: "\\MyServer\MyShareDirectory")

Not permitted are:

- Complete drives
(Example: "C:\")
- Connected network drives
(Example: "X:\")
- Windows® system folder
(Example: "C:\WINDOWS\SYSTEM32")



No check of required memory space!

It is not checked for the required memory space! The user is responsible for the selection of the folders to be stored.

**Restriction for folders of the same name!**

If multiple folders of the same name and with different drive specifications are added, only the first folder is saved.

If there are the following entries

"C:\MyDocuments\MyMTXDocuments" and

"D:\MyDocuments\MyMTXDocuments" in the list, only the folder

"C:\MyDocuments\MyMTXDocuments" is backed up.

Error messages

If an error occurred in a file, this file is marked as faulty.

Refer to the following table for error numbers and causes:

No.	Error message	Cause	Recovery
1	File not processed (yet)!	The file has not been processed, yet.	-
2	You are not authorized!	You do not have the required authorization to perform the requested file operation	Grant authorization
3	Path is too long!	The file path is too long or does not comply with the Windows® convention	Check path
4	Path is invalid!	The file path does not exist or does not comply with the Windows® convention	Check path
5	File is write-protected!	File is write-protected and cannot be restored or overwritten	Cancel file protection
6	File not found!	File does not exist or could not be restored	Check archive content

Tab. 5-3: Error messages

Restoring Extended Data - User Input**Archive**

Click on the button to the right of the archiving path to enter name and storage location of the archive file to be restored.

**Loss of data!**

Any existing files are overwritten during the restoration.

Save your files by archiving before executing the restoration.



It is tried to restore this backup if an error occurred during the process. Before the restoration, a backup image of all target directories is created. If the user partition of the Compact Flash card is restored, the data is downloaded to the PC first.

Contents

The files in the archive can optionally be displayed as list or tree. Select the files and folders to be restored by selecting and deselecting the checkboxes.



The user partition can only be backed up and restored completely. Selecting single files is not possible.

Press the button **Select none** to deselect a file and **Select all** to select all files. In the tree view, use the buttons **Expand** and **Collapse** for a better navigation. When the entries are completed, press **Next** or **Finish**.

Clearing target directories

Target directories can be cleared before restoring the archive. Select the checkbox to clear the target directories.

Drive data



Files which are not in the archive are lost!

General information

To transfer drive parameter sets from one machine to the other or to save them to a data carrier (e.g. a hard disk), multiple options are available for the user.

Parameter backup using:

- IndraWorks drive in the user IndraWorks project
- Command specification via the drive display on a multimedia card

How to back up data is described below, followed by how to restore drive data (drive parameters) using IW Drive.

Backing up or restoring drive parameters using a multimedia card and the commands available on the IndraDrive display are described in detail in the drive documentation "Functional Description - Loading and Saving Parameters" and are therefore not discussed.

In IW Drive, there are generally two options to backup and restore drive parameters.

Backing up and loading the drive parameters of a drive, of a selected number of drives and of all drives at the same time to the current IndraWorks project. The storage directory and the file name are assigned automatically by IndraWorks Drive.

Handling Instruction: Backing Up Drive Parameters of an Axis

The present handling instruction describes how to back up the drive parameters of an individual axis or any number of axes.

IW Engineering/Sercos: Backing up parameters

1. Switch the MTX online in the main menu under "Project - Online" on the Engineering Desktop.
2. Highlight the "Sercos" node in the project tree and right-click to activate the "Backup..." dialog in the context menu to start the data backup of the drive parameters.

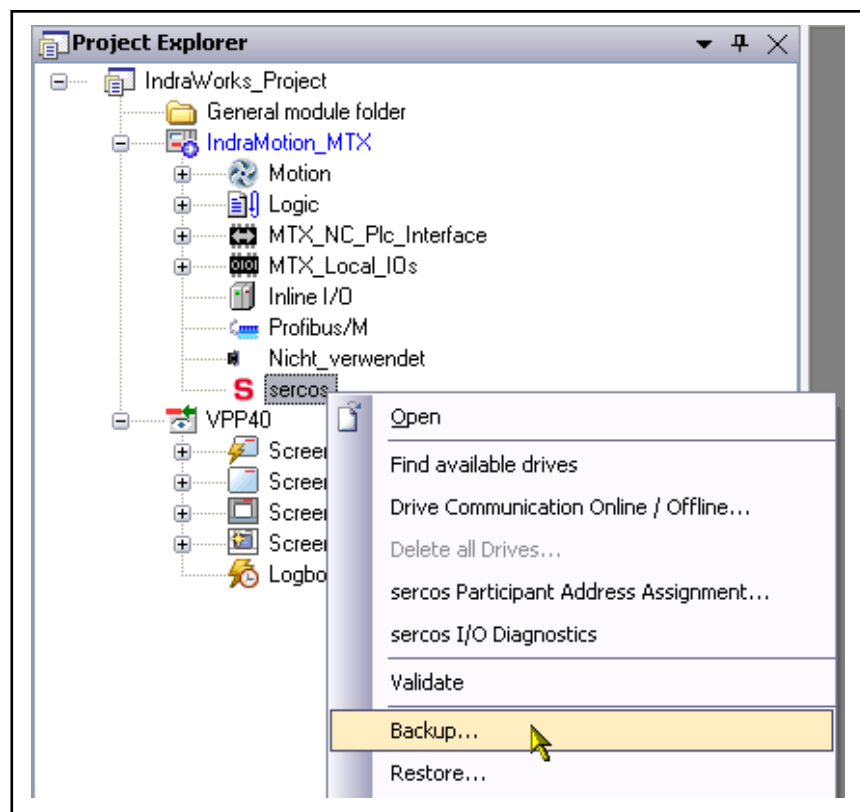


Fig. 5-9: Sercos (archiving drive parameters)

3. Select the drive(s) whose parameters are to be archived.

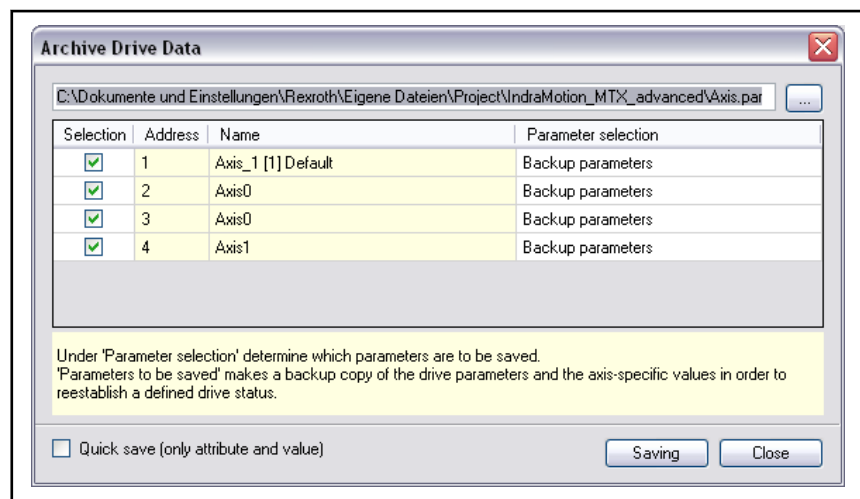


Fig. 5-10: Archive Drive Data (select drives)

After the confirmation, the selected drives are switched online and the data is exported into a parameter file. The progress is displayed comprehensively for all axes.



Fig. 5-11: Export parameters (progress bar)



The files of the drive parameter backup files have the extension "*.par". This extension is to be retained. Otherwise, restoring (importing) the parameter backup is no longer possible.

		Documentation chapter 7.12.1 "Axis commissioning" on page 358
Documentation:	IndraWorks Commissioning	Axis commissioning

Handling Instruction: Restoring Drive Parameters of an Axis

The present handling instruction describes how to restore the drive parameters of an individual axis or any number of axes.

IW Engineering/Sercos: Restoring parameters

1. Switch the MTX control online in the main menu under "Project - Online" in the Engineering Desktop.
2. In the project tree under "Sercos", enable the context menu "Restore ..." to start the restoration of the drive data.

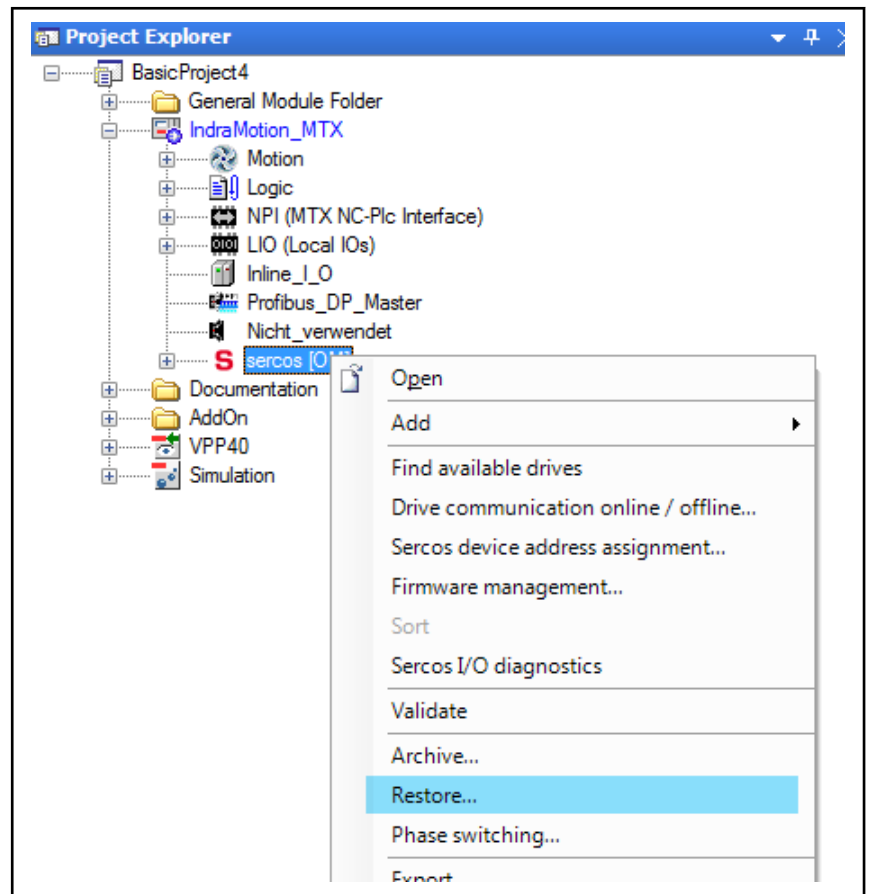


Fig. 5-12: Sercos (restore drive data)

3. Select the drive(s) whose drive parameters are to be restored. Additionally, the assignment of the exported parameter sets to the axes can be changed.

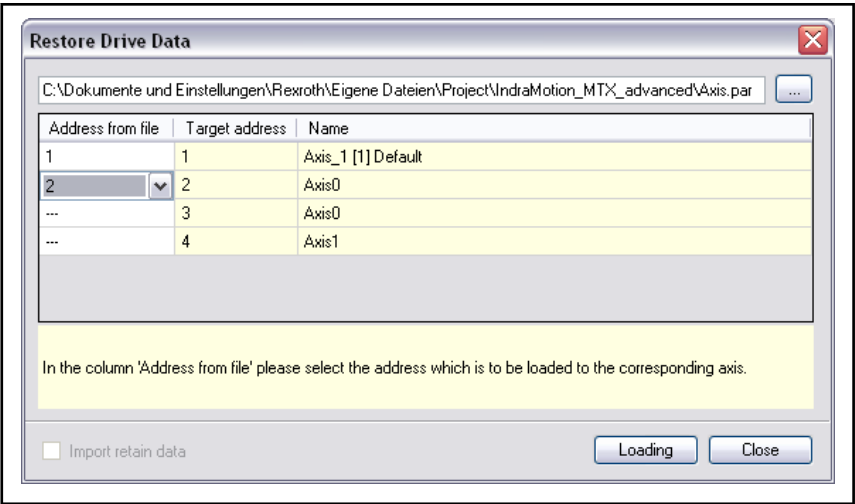


Fig. 5-13: Restore Drive Data (select drives)

- 4. Confirm the selection with **OK** and the restoration of the drive data starts.

The selected drives are switched online and the data is imported into the drive. The progress is displayed comprehensively for all axes.

- 5. To transfer the parameters to the drive, the drives have to be switched to Sercos phase 2.

The following error message is displayed if the drives are not in Sercos phase 2.

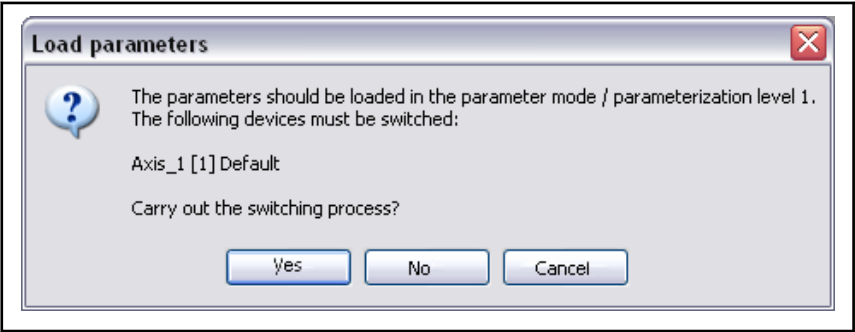


Fig. 5-14: Load parameters (error message while restoring drive parameters)

- 6. A phase can be switched in the context menu under the "Sercos" mode using the "Phase switching..." function.



The backup files of the drive parameters have the extension **"*.par"**. The file type is preset to **"*.par"** in the Explorer window.

		Documentation chapter 7.12.1 "Axis commissioning" on page 358
Documentation:	IndraWorks Commissioning	Axis commissioning

5.4 SCP Analyzer

5.4.1 General information

The SCP Analyzer is an analysis tool to detect a reduced OPC or SCP communication. All client processes requesting data of a local SCPServer are displayed with their object hierarchy (processes, devices, clients, groups, items). When the application is started, it synchronizes automatically with the local server and displays the current object hierarchy.

Save The current state of the SCPServer can be exported to a file for further processing. A file named "SCPAnalyzer.log" is created. It records and saves the current state of all SCP objects. Press the "Save" button in the toolbar or use the menu items **File ► Save**.

Clipboard The selected item of the group item list can be copied to the clipboard. Press the "Save to clipboard" button in the toolbar.

Always on Top The application can always be made visible via the menu items **Edit ► Always on Top**.

5.4.2 Display elements

SCP process tree The left process tree represents the object hierarchy with regard to the logical devices via the local SCPServer. Filter buttons in the toolbar show or hide the individual levels.

Group item list Items running dynamically in one communication group (group items) are dynamically recorded across all clients and entered into the group item list. Some measured variables are visualized there as values and some in bar representation.

- Column "Status":
Group item or related group is active/inactive
- Column "Group Item":
Name of the group item
- Column "Update Rate [ms]":
Command update rate of the group items in [ms]
- Column "Cache [ms]":
Actual update rate between the SCPServer and the logical device in [ms]
- Column "Client Update [ms]":
Actual update rate between SCPServer and client in [ms]

5.4.3 SCP object hierarchy

Each SCP/OPC client is identified as best as possible and entered in an object tree.

Processes:

On the first level, there are the processes that communicate via the local SCPServer.

Devices:

Each process is connected to multiple logical devices (MTX, logbook, drives, etc.).

Clients:

The clients connect to logical devices.

Static items:

The static items are located below the clients.

Groups:

The created groups are located below the clients.

Group items:

The group items are located below the group.

Restriction for OPC clients It is not possible to uniquely identify all clients communicating with the SCPsServer using the OPC interface. Each OPC client logs on anonymously to the SCPsServer and can only be detected as such due to a reference identifying the root and the SCPsServer as pair. Thus, identifying the client name or the process ID (PID) is not possible for OPC clients. However, OPC clients can be easily identified via the items used. WinStudio uses the unique TAG definitions with a defined syntax for example and is thus noticeable.

5.4.4 Interpreting Values

Column "Update Rate [ms]"	This column shows the maximum achievable command update rate of the group item. Usually, this corresponds to the group update rate in which the item is running. For example, if the item runs in a group with 100 ms, this is the maximum achievable update rate of the item.
Column "Server Cache [ms]"	This column shows the access rate of the item, i.e. the rate used by the SCPsServer to monitor or query the changes of the item on the logical device. The more the value approximates the nominal value, the larger the bar (100% = nominal). If the SCPsServer cannot access the logical device, "--" is displayed.
Column "Client Update [ms]"	This column shows the update rate of the item, i.e. the rate used by the SCPsServer to transmit the changes of the item to the client. If the item does not change, "--" is displayed.

6 HMI commissioning

6.1 Introduction



All basic functions as well as the basic handling to create and change a visualization is described in detail in the "IndraWorks HMI" manual.

(also refer to [chapter 1.2.3 "Commissioning" on page 3](#))

For MTX-specific topics to operate an "MTX standard application", refer to the documentation "Standard NC Operation".

(also refer to [chapter 1.2.4 "Operation" on page 4](#))

By creating a visualization of the type "MTX standard application", an executable MTX user interface is created. That interface can be adjusted and extended by the user.

A visualization to operate and monitor a machine tool consists of multiple screens.

These screens consist of multiple elements that can be divided into two categories:

- Screen frame (ScreenFrame)
- Screen application (ScreenApplication)

The screen frame generally consists of

- Machine status line (header) for a compact overview on the machine state
- Operating areas (OP-panel) for a logical grouping of screens
- Function keys (F-panels) to call screen-related functions
- Machine keys (M-panels) to execute screen-related machine functions

One screen application (ScreenApplication) can be one of the following

- A control (ACI screens) can be delivered with the system or programmed by the user
- A tool data display (DBT1 screens) for different views on the tool stock
- A system data display (SystemData screens) for different views on system data
- A WinStudio screen (WinStudio) to visualize any data
- An operating screen (Operating screens) for special operations such as traversing axes

There is a node for editing in the Project Explorer for each of the elements listed above.

Individual elements to obtain a screen are merged in the node "Screens" or "Screens MTX".

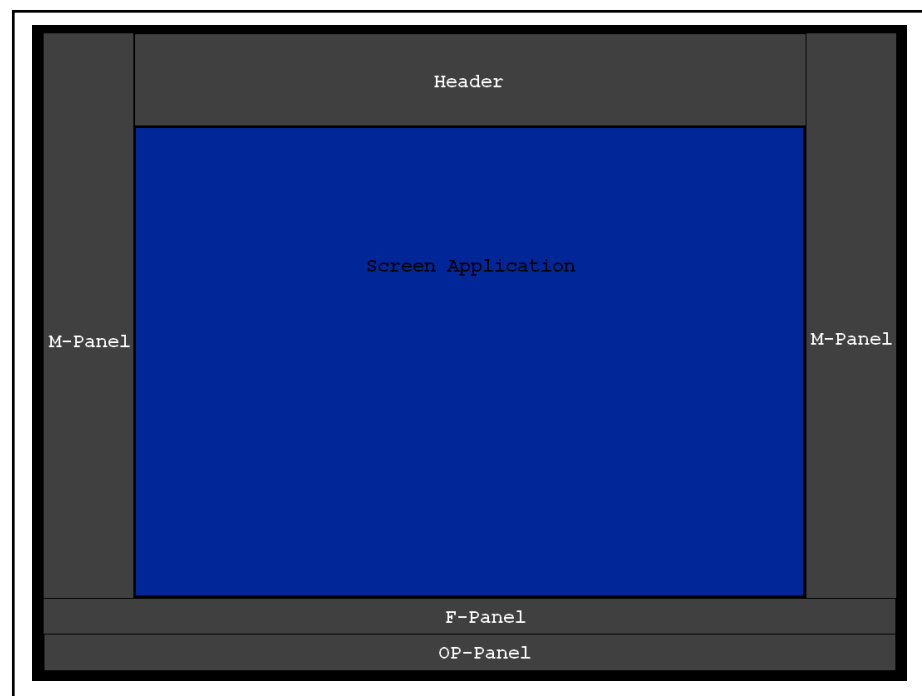


Fig. 6-1: Screen design

6.2 ScreenFrame

6.2.1 General information

By creating a visualization of the type "MTX standard application", an executable MTX user interface is created. That interface can be adjusted and extended by the user.

A visualization to operate and monitor a machine tool consists of multiple screens.

These screens consist of multiple elements that can be divided into two categories:

- Screen frame (ScreenFrame)
- Screen application (ScreenApplication)

The screen frame generally consist of:

- Machine status line (header) for a compact overview on the machine state
- Operating areas (OP-panel) for a logical grouping of screens
- Function keys (F-panels) to call screen-related functions
- Machine keys (M-panels) to execute screen-related machine functions

6.2.2 Header (machine status area)

General information

The machine status area (header) provides a compact overview on the machine status.



Settings in the machine status area (header) can exclusively be made on the operating station in IndraWorks Operation!

The configuration of the machine status area (Header) is described in detail in the "Standard NC Operation" manual!



In IndraWorks Engineering, this data is archived, added to version control as well as exported and imported.

Data exchange between IndraWorks Operation and IndraWorks Engineering is started via the context menu of the HMI node.

- IndraWorks Operation --> IndraWorks Engineering: Visualization Data \ Loading changes from operating station
- IndraWorks Engineering --> IndraWorks Operation: Visualization Data \ Transfer and Activate

6.2.3 OP-panel (operating keys)

General information

OP-panel = Operating keys = OP-bar

An OP-panel is a panel of eight keys located in the lower part of the screen.

OP-key = Operating key

The individual key of an OP-panel is called the OP-key. An OP-key is always part of an OP-panel.

The keys of the OP-panel are mainly used to switch the operating areas. It is additionally possible to activate an individual screen via an OP-key.



The general handling of operating areas is described in the "IndraWorks HMI" manual.

In contrast to M- and F-keys, activating functions and switching PLC variables is not possible via OP-keys.

After the creation of a standard MTX application, all keys of the OP-panel are pre-assigned with operating areas. Operating areas are configured below the "Screens" node in IndraWorks Engineering.



All functions and screen of a standard MTX application are described in detail in the "Standard NC Operation" manual.



Fig. 6-2: Example of an OP-panel

6.2.4 F-panel (function keys)

General information

F-panel = Function key bar = F-bar

A field of eight F-keys is called an F-panel. Each F-panel can be assigned to one or more screens. The F-panel is located on the lower edge of a screen, directly above the operating (OP) keys.

F-key = Function key

The individual key of an F-panel is called the F-key. An F-key is always part of an F-panel.

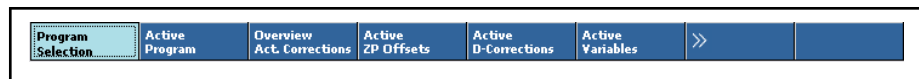


Fig. 6-3: Example of an F-panel

MTX standard F-panel

By creating a visualization of the type "MTX standard application", an executable MTX user interface is created. That interface can be adjusted and extended by the user. The provided screens are mandatory for correct functioning of the "MTX standard application"! To prevent accidental deleting of data, these screens are not visible in the Project Explorer but can be assigned in the operating area editor. The F-panels of the screens can be extended with functions. By deleting or renaming such an F-panel, the "MTX standard screen" the F-panel is part of would not function correctly anymore!

All affected F-panels are listed in the following chapters.

6.2.5 M-panel (machine keys)

General information

M-panel = Machine key bar = M-bar

A field of eight M-keys is called an M-panel. Each M-panel can be assigned to one or more screens. The M-panel is located at the left or right edge of a screen.

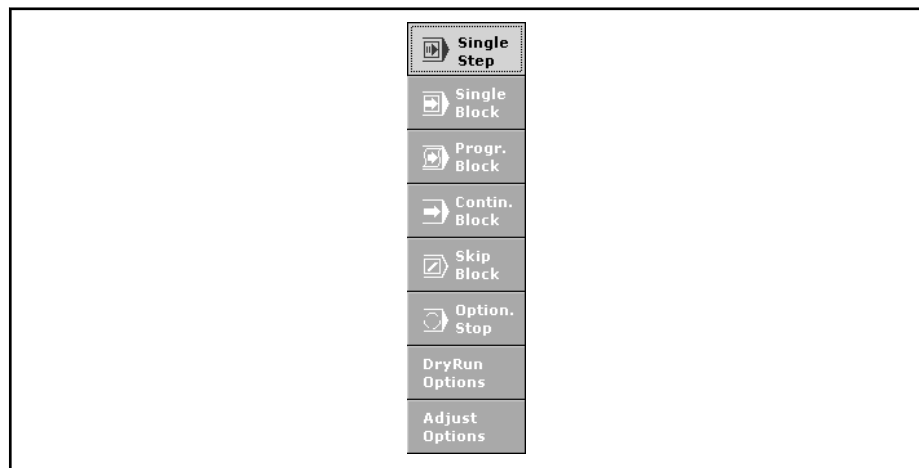


Fig. 6-4: Example of an M-panel

M-key = Machine key

A single key to the left or to the right of the BTV is called an M-key. It is always part of an M-panel.



Fig. 6-5: Example of an M-key

Commissioning the M-keys

Apart from activating defined functions via the machine keys, actions in the PLC can also be triggered.

For a correct assignment between pressed M-key and PLC variable, proceed with the following steps.

Implement the PLC function block "IL_MKeys" and declare its instance in IndraWorks Engineering

1. Implement the "IL_MKeys" function block from the "RIL_HMI_Uilities" library into the PLC program.
2. The inputs L1-L8 and the inputs R1-R8 do not have to be assigned. TRUE is to be assigned to the "Enable" input. If the touch operation is also to be supported, TRUE also has to be assigned to the "TouchEnable" input.



Ensure that the program is considered during the PLC program execution (e.g. calling from the main program or directly from the task).

3. The complete instance path of the declared function block "IL_MKeys", has to be available in the symbol configuration.
4. Compile PLC program and load it into the control.
5. Open the "M-Panels" dialog in IndraWorks Engineering and enter the instance path to the function block "IL_MKeys" declared in IndraLogic.
6. Close the dialog with "OK".

6.2.6 F-/M-/OP-panel function overview

Function	Applicable	Brief description
Adjustment states	F-/M-panel	Opens the first adjustment screen for the block pre-run
Axes adjustment states	F-/M-panel	Opens the adjustment screen to adjust the axis positions for the block pre-run
Adjustment states of auxiliary functions	F-/M-panel	Opens the adjustment screen to adjust the auxiliary functions for the block pre-run
Adjustment states of NC functions	F-/M-panel	Opens the adjustment screen to adjust the NC functions for the block pre-run
Adjustment states of spindles	F-/M-panel	Opens the adjustment screen to adjust the spindle states for the block pre-run
Adjustment states of tools	F-/M-panel	Opens the adjustment screen to adjust the active tool for the block pre-run
Current Close screen + Re-entry	F-/M-panel	Calling the re-entry function by closing the active screen
Active variables	F-/M-panel	Opens the variable editor with the currently active list of variables
Active D-corrections	F-/M-panel	Opens the display with the active list of D-corrections
Active zero offsets	F-/M-panel	Opens the display with the active list of ZO corrections
Active placements	F-/M-panel	Opens the placement editor with the currently active placement list
Close active screen	F-/M-panel	Closes the screen currently active in the display
System overview /detail	F-/M-panel	Opens the detailed view of the system overview
System overview/filter	F-/M-panel	Filter for the system overview
Backing up user data	F-/M-panel	Backs up user-specific settings in the Operation Desktop
Enabling the operating area	OP-panel	Switches the transmitted operating area and closes all screens of the active operating area
Operating screen information	F-/M-panel	Status display in operating screen toggles between value, text and I/O address

HMI commissioning

Function	Applicable	Brief description
Exits IndraWorks Operation	F-/M-/OP-panel	Closes the Operation Desktop
Command to ACI screen	F-/M-panel	Via this function, different commands can be send to the ACI screen
Command to ACI screen	F-/M-panel	Via this function, different commands can be send to the ACI control
Log in user	F-/M-/OP-panel	Opens login dialog to enter the user name and password
Log out user	F-/M-/OP-panel	Logs out the logged in user
Zeros the reference position	F-/M-panel	Opens the screen to zero the reference position
Screen keyboard	F-/M-panel	
Display change	F-/M-/OP-panel	Function to change system/user screen displays
DBT2_ChangeConfig	F-/M-panel	Changes the display for DBT2 by calling a new list configuration
DBT2_ChangeQuery	F-/M-panel	Changes the display for DBT2 by calling a new list query
DBT2_Edit	F-/M-panel	Edits DBT2 data block
DBT2List	F-/M-panel	Calling a DBT2 list
Details about the connection interruption	F-/M-panel	Opens a screen with detailed information about the connection interruption
Diagnostics	F-/M-panel	Opens the dialog overview
Diagnostics/detail	F-/M-panel	Opens the detailed view of the diagnostic view
Diagnostic/filter	F-/M-panel	Opens a filter dialog to filter the dialog view
Diagnostics/Delete	F-/M-panel	Deletes the diagnostic message from the diagnostic overview
Dialog Pause and Energy mode	F-/M-panel	Opens dialog to specify a new energy mode or a pause time for the energy management in the PLC
DP diagnostics	F-/M-panel	Opens a screen to diagnose the Profibus
Level change	F-/M-panel	Function to switch the F/M-panels
Editor of active program	F-/M-panel	Opens the NC program editor with the currently active program
Energy management energy functions	F-/M-panel	Opens the screen with an overview of all energy functions and its status
Energy management events	F-/M-panel	Opens screen with events of the energy management
Energy management component information	F-/M-panel	Opens screen with energy management information about components
Energy management machine information	F-/M-panel	Opens screen with energy management information about the machine
Energy monitor	F-/M-panel	Opens the screen to display the energy consumption
Archive device	F-/M-panel	Starts archiving of control data (TAR file)
IndraWorks Engineering	F-/M-/OP-panel	Starts the Engineering Desktop
Interface PLC-NC	F-/M-panel	Opens a display showing the status of the individual interface signals between PLC and NC (channel, axis and spindle signals)
Reset channel	F-/M-panel	Initiates reset for the active channel
Switching the channel	F-/M-panel	Switches to the next channel
Circularity test	F-/M-panel	Opens the screen to execute the circle test

Function	Applicable	Brief description
Criteria analysis	F-/M-panel	Opens the criteria analysis
Power monitor	F-/M-panel	Opens the screen to display the performance data
Logbook	F-/M-panel	Opens the logbook display
Resets maximum value	F-/M-panel	
Probe 1	F-/M-panel	Opens the initialization screen for probe 1
Probe 2	F-/M-panel	Opens the initialization screen for probe 1
Initializes the probe	F-/M-panel	Opens the initialization screen for manual measurement functions
MTX system status	F-/M-panel	Opens the MTX system status to diagnose the NC control
MTX system status connection interruption	F-/M-panel	Opens the screen with detailed information about the connection interruption
MTX acr	F-/M-panel	
NC Analyzer	F-/M-panel	Starting the NC-Analyzer
NC-diagnostic monitor Show/hide	F-/M-panel	
NC editor plane change	F-/M-panel	Plane switching for cycle input masks
Cancels entries NC editor	F-/M-panel	Cancel input for cycle input masks
NC editor channel selection	F-/M-panel	Channel selection dialog for cycle input masks
Applies parameters NC editor	F-/M-panel	Parameter application for cycle input masks
Set cycle group NC Editor	F-/M-panel	Calls a cycle group
Determines the zero point	F-/M-panel	Opens screen to probe zero point
Enables the zero point table	F-/M-panel	Opens the drop-down menu to enable a zero point table
Block pre-run options	F-/M-panel	Opens the dialog to define block pre-run options
Test feed/test rapid traverse options	F-/M-panel	Opens the dialog to define block pre-run/rapid traverse options
Zero position display on/off	F-/M-panel	Sets the position display to zero
Program selection	F-/M-panel	Opens the window for program selection
Computer	F-/M-panel	Calls the calculator
Block search	F-/M-panel	Opens the screen for block search
Block search/block pre-run	F-/M-panel	Calls the operating screen for block search and block pre-run
Block pre-run	F-/M-panel	Opens the screen for block pre-run
Sercos diagnostics	F-/M-panel	Opens screen for Sercos diagnostics
Set channel	F-/M-panel	Change to a certain channel
Single Tool_Import	F-/M-panel	Opens the dialog to import of an individual tool
Language selection	F-/M-panel	Switches the interfaces/project language for OPD
PLC list	F-/M-panel	Opens the screen for PLC list display
Station selection	F-/M-panel	Opens the screen for station switching

HMI commissioning

Function	Applicable	Brief description
System control reset	F-/M-panel	Triggers control reset for all channels
SystemData_Print	F-/M-panel	Starts printing the system data
SystemData_PrintPreview	F-/M-panel	Starts the print preview to print the system data
Keyboard code trigger	F-/M-panel	Triggers a keyboard code
Hide key	F panel	
Execute key	F panel	
TL_Active_Query_Filter	F-/M-panel	Activates the defined query filter for the tool list
TL_ActivateNextList	F-/M-panel	Function to change the focus to the next DBT list in case of several lists in a DBT screen
TL_ChangeConfig	F-/M-panel	Changes the display for tools by calling a new list configuration
TL_ChangeQuery	F-/M-panel	Changes the display for tools by calling a new list query
TL_Copy	F-/M-panel	Starts the dialog to copy a tool data block
TL_DefineQueryFilter	F-/M-panel	Opens a screen to define a query filter for tool lists
TL_Delete	F-/M-panel	Deletes the selected tool data block
TL_Edit	F-/M-panel	Starts the online tool editor
TL_Edit_Additive	F-/M-panel	Switches between the input modes "Absolute" and "Additive" in the online tool editor and the tool list
TL_Editor_Command	M-panel	Selects the next tool in the tool editor Parameter: Edit_Next_Tool
		Selects the previous tool in the tool editor Parameter: Edit_Prev_Tool
		Deletes the basic data block in the tool editor Parameter: Delete_BasicData_ToolType
		Saves the basic data block in the tool editor Parameter: Save_BasicData_ToolType
TL_Export	F-/M-panel	Starts the dialog to export tool lists or individual tools
TL_Import	F-/M-panel	Starts the dialog to import tool lists or individual tools
TL_Insert	F-/M-panel	Starts the offline editor to insert a new tool
TL_Move	F-/M-panel	Starts the dialog to move a tool data block
TL_SORT	F-/M-panel	Starts the dialog to sort the tool list
TL_SPS_Copy	F-/M-panel	Starts the function Copy tool via the PLC
TL_SPS_Delete	F-/M-panel	Starts the function Delete tool via the PLC
TL_SPS_Move	F-/M-panel	Starts the function Move tool via the PLC
ToolCursor	F-/M-panel	The current cursor position (sector and location) is written to a PLC variable
ToolCursorEditor	F-/M-panel	Tool currently edited in the tool editor (sector and location) is written to a PLC variable

Function	Applicable	Brief description
ToolList	F-/M-panel	Calls a tool list
ToolList1-16	F-/M-panel	Switches the display to the appropriate tool list Only available due to compatibility reasons Do not use anymore!
Overview on current corrections	F-/M-panel	Opens a display with an overview on the active ZOs, D-corrections and tools
Switch coordinate system	F-/M-panel	Switches the display between the machine and the tool coordinate system
Switches the technology display	F-/M-panel	
Switch variant position display	F-/M-panel	Switches between the variants for the position display defined in Operation Desktop
Variables	F-/M-panel	Opens the screen for variable display
Variables/parameters	F-/M-panel	Opens the screen to display variables and parameters
Probe tool	F-/M-panel	Opens screen to probe the tool
Re-entry	F-/M-panel	Calls the screen for re-entry in the NC program
Two channel display, currently active channel change	F-/M-panel	Switching the active channel in the two channel display
Calls the two channel display	F-/M-panel	Opens the screen for two channel display
Exits the two channel display	F-/M-panel	Exits the screen for two channel display
Two channel display channel selection	F-/M-panel	Opens the dialog to select the channel in the two channel display
Two channel display options	F-/M-panel	Opens the dialog to define the two channel display options
Two channel display: setting the passive channel	F-/M-panel	Opens the dialog to set the passive channel
Two channel display: changing the passive channel	F-/M-panel	Opens the dialog to change the passive channel
Executing the cycle	F-/M-panel	Executes the cycle in Manual mode

Tab. 6-1: Overview of the configured function on the F-, M- and OP-panel

6.3 Screen application

6.3.1 General information

By creating a visualization of the type "MTX standard application", an executable MTX user interface is created. That interface can be adjusted and extended by the user.

A visualization to operate and monitor a machine tool consists of multiple screens.

These screens consist of multiple elements that can be divided into two categories:

- Screen frame (ScreenFrame)
- Screen application (ScreenApplication)

One screen application (ScreenApplication) can be one of the following:

- A control (ACI screens) can be delivered with the system or programmed by the user
- A tool data display (DBT1 screens) for different views on the tool stock
- A system data display (SystemData screens) for different views on system data
- A WinStudio screen (WinStudio) to visualize any data
- An operating screen (Operating screens) for special operations such as traversing axes

6.3.2 ACI screens

General information

In "ACI screens" (ACI == Application Container Interface), ActiveX- or .NET-Controls can be embedded and arranged as partial screens. The "ACI screen" is then embedded as screen call of type "UserAciScreen" in one screen together with "M-panels" and "F-panels".



The general handling of "ACI screens" is described in the "IndraWorks HMI" manual.

By creating a visualization of the type "MTX standard application", an executable MTX user interface is created. That interface can be adjusted and extended by the user. This also includes multiple "ACI screens". These contain different controls displaying the runtime process data of the control.

Process displays

Overview

The process displays are each provided with machine and interface-dependent parameters:

- Machine-dependent parameters are set in the NC configurator in IndraWorks Engineering. For details on the parameters, refer to the "MTX Machine Parameters".
- Interface-dependent parameters are set in the "Properties" dialog of the relevant displays in IndraWorks Operation.

General parameters - NC configurator

NC configurator path	MACODA	Name
/GUI/DispUnit	6020 00030	Measuring unit for the display

Tab. 6-2: General parameters for NC configurator

General parameters - Properties

Name	Values	Description
Maximum decimal digits	{10,11,12}	Specifies the maximum number of the decimal digits that can be displayed in columns. The parameter defines the width of the columns in the columns in which decimal digits are displayed
Font size	{Default,7...16}	Specifies the font size of the display. The value "optimal" determines the best font size possible

Name	Values	Description
Display title	{Yes, No}	Specifies whether the title of the display is to be displayed
Optimum line height	{Yes, No}	Specifies whether the optimum line height is to be displayed
Line separator	{Yes, No}	Specifies whether a line separator is to be displayed
Colored lines	{Yes, No}	Specifies whether the lines of the display are to be colored. This improves readability

Tab. 6-3: General parameters - Properties

Communication parameters - NC configurator

NC configurator path	MACODA	Name
-	-	-

Tab. 6-4: Communication parameters for NC configurator

Communication parameters - Properties

Name	Values	Description
Channel	{Default, 1...<Max. channel no.>}	Specifies whether the display is to be fixed on a channel. In this case, the display always shows the selected channel, irrespective of the active channel of the basic NC screen. The parameter enables the configuration of NC screens displaying information across multiple channels. In the "default" setting, the display shows information on the active channel.
Operating station (IP, host)	{default, [IP address] [host name]}	Specifies whether the display is to be set for a specified LDX device. In this case, the display always shows the selected device, irrespective of the device active in the basic NC screen. To identify a device, specify the IP address or the host name of the computer into which the device is integrated. This computer has to be registered as DCOM server. For further information on this topic, refer to the following manual: "MTX Integration of OEM applications". The parameter enables the configuration of NC screens displaying information across multiple devices. In the "default" setting, the display shows information on the active channel.

Tab. 6-5: Communication parameters - Properties

Position display - NC configurator

NC configurator path	MACODA	Name
/GUI/PosDisp/CoordSys/Ch[i]/Wcs	6005 00100	Coordinate system "Wcs" available in the channel
/GUI/PosDisp/CoordSys/Ch[i]/Mcs	6005 00100	Coordinate system "Mcs" available in the channel
/GUI/PosDisp/CoordSys/Ch[i]/Bcs	6005 00100	Coordinate system "Bcs" available in the channel
/GUI/PosDisp/CoordSys/Ch[i]/Acs	6005 00100	Coordinate system "Acs" available in the channel

HMI commissioning

NC configurator path	MACODA	Name
/GUI/PosDisp/Ord/Ch[i]/NofAxPosDisp	6005 00023	Maximum number of axes in axis display
/GUI/PosDisp/Ord/Ch[i]/Win/Dr[j]/ModeAxChWin	6005 00022	Display axis in the channel window
/GUI/PosDisp/Ord/Ch[i]/Win/Dr[j]/PrioAxChWin	6005 00022	Order of the axis in the channel window
/GUI/PosDisp/Ord/Ch[i]/AddWin/Dr[j]/ModeAddAx	6005 00020	Display axis in the additional window
/GUI/PosDisp/Ord/Ch[i]/AddWin/Dr[j]/PrioAddAx	6005 00020	Order of the axis in the additional window
/GUI/PosDisp/Ord/Ch[i]/AddWin/ChCoordPack[j]/ModeChPack	6005 00021	Display channel coordinate package in the additional window
/GUI/PosDisp/Ord/Ch[i]/AddWin/ChCoordPack[j]/PrioChPack	6005 00021	Order of the channel coordinate package in the additional window
/GUI/PosDisp/Prec/Dr[i]/PrecPosMetr	6020 00011	Number of decimal places for axis display (metric)
/GUI/PosDisp/Prec/Dr[i]/PrecPosInch	6020 00012	Number of decimal places for axis display (inch)

Tab. 6-6: Position display - NC Configurator

Position display - Properties - View**View**

Name	Values	Description
Axis name length	{4...16}	Specifies the maximum length of the axis names that can be displayed. The parameter determines the width of the column of axis names.
Active coordinate system	{ACS, MCS, BCS, WCS}	Specifies the active coordinate system. The coordinate system can also be switched via the key <F4 coordinate system> in the menu of this display or via the key <F6 coordinate system> in the submenu of the basic NC screen. Only coordinate systems entered as existing in the NC configurator are provided.
Active variant	{1..10}	Specifies the active display variant. The variant can also be switched via the key <F2 variant> in the menu of this display or via the key <F5 variant position display> in the submenu of the basic NC screen. Available variants are set in the variant editor in the "Properties" dialog.
Channel assignment	{Yes, No}	Specifies whether an extra column is displayed showing the channel assignment of the axes.
Measuring unit	{Yes, No}	Specifies whether an extra column is displayed showing the measuring unit of the axes.

Name	Values	Description
System axis names	{Yes, No}	Specifies whether the system axis names are to be displayed in brackets behind the channel axis names. The parameter is only effective if the channel axis name differs from the system axis name.
Radius in diameter programming	{Yes, No}	Specifies whether the axis positions are to be displayed as radius when programming the diameter. The parameter is only effective in the coordinate system "ACS" and for axes programmed in diameter. "r" = Axis position is a radius value "Ø" = Axis position is a diameter value
Unit symbol	{Yes, No}	Specifies whether the unit symbol [mm] or [inch] is to be displayed in the title line of the display. The parameter is only effective if the display is not in the unit mode "automatic unit according to G70/G71". This mode is set in the NC configurator (measuring unit for the display).

Tab. 6-7: Position display - Properties

Variants The variant editor is intended to create and edit display variants. Variants are optical designs with different column information created globally for all displays of this type and that can be assigned and selected individually for each display. Multiple designs can also be assigned to one variant. The variants are switched using the F-key <F2 variant> in the F-key menu of a focused display.

Display filter The displayable axes set in the NC Configurator under "/GUI/PosDisp/..." apply globally to all position displays in IndraWorks Operation. Moreover, individual axes can be shown or hidden individually for each position display using their system axis name addresses. The hiding thus overwrites the global settings in the NC configurator and the axis is never visible in this position display, although it is globally parameterized like that.

By means of this feature, it is for example possible to hide individual axes in the position display of the operation mode "Manual" as opposed to the display in the operation mode "Automatic" of the same channel.

Select the checkbox "Axis hidden" for those axes that are never to be shown in the display.

Display of the actual position in the WCS

The display of the actual position in the WCS is CPU-intensive especially in case of an activated transformation on the NC-side (backward transformation required) and has thus to be activated directly via the NC configurator if required.

If a value > 0 is entered for the parameter `TRA/GuiDisp/UpdateTimePosDisp`, this axis position is calculated in the kernel in this IPO cycle. The position display indicates the real actual position of the drives in the WCS instead of "--".

Position display - Properties - Load: Velocity/acceleration

View Definition of the display of velocity and acceleration load

Name	Values	Description
Value	{Yes, No}	Specifies whether the current value is displayed as number.
Maximum value	{Yes, No}	Specifies whether the maximum value is displayed as number behind the current value and additionally as colored bar.

HMI commissioning

Name	Values	Description
Reset maximum value		The maximum value can always be reset via the F-key "Reset peak values" in the F-keypad of the focused display or via the action of the same name. This can be assigned to an M-key for example.
Full deflection corresponds to value [%]	10...1000	Determines the % value of the maximum full deflection. To represent overshooters (>100%), determine a value higher than 100. For a better resolution of smaller values, determine a value lower than 100.
Color gradient	{various color gradient types}	Specifies the color gradient type. Various color gradient types are available. Specify the color and the change in color in relation to a certain percentage.

Tab. 6-8: Position display - Properties

Variants [chapter "Position display - Properties - View" on page 78](#)



The control first provides the values for the velocity and acceleration axis load diagnostics if these diagnostics are activated via system data entry (/SysNcDgnCtrl/AxesLoadDgn/VelLoadDgn/DgnMode = 2, .../AccLoadDgn/DgnMode = 2).



Switch off the velocity and acceleration load diagnostics using the system data entry (.../VelLoadDgn/DgnMode = 0, .../AccLoadDgn/DgnMode = 0) if these diagnostics are not required anymore! This impedes unnecessary loads of the interpolator interrupt.

Technology display - NC configurator

NC configurator path	MACODA	Name
/GUI/TechDisp/FeedDisp/PrecFeedMetr	6020 00001	Number of decimal places for feed display (metric)
/GUI/TechDisp/FeedDisp/PrecFeedInch	6020 00002	Number of decimal places for feed display (inch)
GUI/TechDisp/SpeedDisp/PrecSpSpeed	6020 00005	Number of decimal places for spindle speed
/GUI/TechDisp/Ch[i]/SpSpGrDisp[j]/SpDisp/SpIndSpDisp	6005 00030	Number of the spindle to be displayed

Tab. 6-9: Technology display for NC configurator

Technology display - Properties - View

View

Name	Values	Description
Active variant	{1..10}	Specifies the active display variant. The variant can also be switched via the key <F2 Variant> in the menu of this display. Available variants are set in the variant editor in the "Properties" dialog.
Layout	{all, feed, spindle}	Specifies the layout of the display. In the display either only spindle or only feed or all technology data can be displayed.
Spindle name length	{4...16}	Specifies the maximum length of the spindle names that can be displayed. The parameter determines the width of the column of axis names.
Unit symbol	{Yes, No}	Specifies whether the unit symbol [mm] or [inch] is to be displayed in the title line of the display. The parameter is only effective if the display is not in the unit mode "automatic unit according to G70/G71". This mode is set in the NC configurator (measuring unit for the display).

Tab. 6-10: Technology display - Properties

Variants [chapter "Position display - Properties - View" on page 78](#)

Technology display - Properties - Spindle power/torque

View Definition of the visual design of the spindle power and torque display

Name	Values	Description
Value	{Yes, No}	Specifies whether the current value is displayed as number.
Maximum value	{Yes, No}	Specifies whether the maximum value is displayed as number behind the current value and additionally as colored bar.
Reset maximum value		The maximum value can always be reset via the F-key "Reset peak values" in the F-keypad of the focused display or via the action of the same name. This can be assigned to an M-key for example. A PLC variable of type "BOOLEAN" can also be specified. If the edge changes from "False" to "True", it is also reset.
Full deflection corresponds to value [%]	10...1000	Determines the % value of the maximum full deflection. To represent overshooters (>100%), determine a value higher than 100. For a better resolution of smaller values, determine a value lower than 100.
Color gradient	{various color gradient types}	Specifies the color gradient type. Various color gradient types are available. Specify the color and the change in color in relation to a certain percentage.

Tab. 6-11: Technology display - Properties

Variants [chapter "Position display - Properties - View" on page 78](#)

Program section display - NC configurator

NC configurator path	MACODA	Name
-	-	-

*Tab. 6-12: Display of program section - NC configurator***Program section display - Properties****View**

Name	Values	Description
Active program	{Yes, No}	Specifies whether the name of the active subroutine is to be displayed in the title line of the display.
History	{1...10}	Specifies the number of the NC blocks of the past (history). The parameter has to be smaller than the parameter "NC blocks".
Layout	{all, bar, icon}	Specifies the layout of the display. The active NC block is only optionally highlighted in color, marked with an icon or with both visualization options.
NC blocks	{4...40}	Number of the total NC blocks to be displayed (including the history).

*Tab. 6-13: Program section display - Properties***Program nesting display - NC configurator**

NC configurator path	MACODA	Name
-	-	-

*Tab. 6-14: Program nesting display - NC configurator***Program nesting display - Properties****View**

Name	Values	Description
Layout	{indented, list}	Specifies the layout of the display. The call sequence of the NC programs can either be a list or a tree with indentations.
Program number Position	{1...15}	Specifies the maximum number of decimal places for the NC program numbers. The higher the parameter is set, the less space remains to display the subroutine paths.
Nesting depth	{0...24}	Specifies the maximum nesting depth of the call sequence.

*Tab. 6-15: Program nesting display - Properties***Offset display - NC Configurator**

NC configurator path	MACODA	Name
/GUI/Zotdisp/Ch[i]/NofZot-Sets	6005 00061	Number of banks to be displayed for zero offsets

Tab. 6-16: Offset display - NC configurator

Offset display - Properties**View**

Name	Values	Description
ZO table name	{Yes, No}	Specifies whether the ZO table name is to be displayed
Line title	{Yes, No}	Specifies whether the line title is to be displayed

*Tab. 6-17: Offset display - Properties***Offset table display - NC configurator**

NC configurator path	MACODA	Name
/GUI/Zotdisp/Ch[i]/NofZot-Sets	6005 00061	Number of banks to be displayed for zero offsets
/GUI/TabDisp/PrecLinMetr	6020 00021	Number of decimal places for linear axes (metric)
/GUI/TabDisp/PrecRotMetr	6020 00021	Number of decimal places for rotary axes (metric)
/GUI/TabDisp/PrecLinInch	6020 00022	Number of decimal places for linear axes (inch)
/GUI/TabDisp/PrecRotInch	6020 00022	Number of decimal places for rotary axes (inch)

*Tab. 6-18: Offset table display - NC configurator***Offset table display - Properties****View**

Name	Values	Description
Number of characters in the table	{8...32}	Specifies the maximum character length of the ZO table. This parameter is not effective if the parameter "additional information" is set to "No".
Display unit	{Default, [mm], [inch]}	Measuring unit for the display
System axis names	{Yes, No}	Specifies whether the system axis names are to be displayed in brackets behind the channel axis names. The parameter is only effective if the channel axis name differs from the system axis name.
Additional information	{Yes, No}	Specifies whether additional information is to be displayed. An additional column with additional information is displayed.

*Tab. 6-19: Offset table display - Properties***Variants** [chapter "Position display - Properties - View" on page 78](#)**Placement display - NC Configurator**

NC configurator path	MACODA	Name
/GUI/Pmtdisp/Ch[i]/NofPmt-Sets	6005 00071	Number of banks to be displayed for placements

*Tab. 6-20: Placement display for NC configurator***Placement display - Properties****View**

Name	Values	Description
Placement table name	{Yes, No}	Specifies whether the placement table name is to be displayed
Line title	{Yes, No}	Specifies whether the line title is to be displayed

Tab. 6-21: Placement display - Properties

Placement table display - NC configurator

NC configurator path	MACODA	Name
/GUI/PmtDisp/Ch[i]/NofPmt-Sets	6005 00071	Number of banks to be displayed for placements
/GUI/TabDisp/PrecLinMetr	6020 00021	Number of decimal places for linear axes (metric)
/GUI/TabDisp/PrecRotMetr	6020 00021	Number of decimal places for rotary axes (metric)
/GUI/TabDisp/PrecLinInch	6020 00022	Number of decimal places for linear axes (inch)
/GUI/TabDisp/PrecRotInch	6020 00022	Number of decimal places for rotary axes (inch)

Tab. 6-22: Placement table display - NC configurator

Placement table display - Properties**View**

Name	Values	Description
Number of characters in the table	{8...32}	Specifies the maximum character length of the placement table. This parameter is not effective if the parameter "additional information" is set to "No".
Display unit	{Default, [mm], [inch]}	Measuring unit for the display
Additional information	{Yes, No}	Specifies whether additional information is to be displayed. An additional column with additional information is displayed.

Tab. 6-23: Placement table display - Properties

Variants [chapter "Position display - Properties - View" on page 78](#)

Tool display - NC configurator

NC configurator path	MACODA	Name
-	-	-

Tab. 6-24: Tool display - NC configurator

Tool display - Properties**View**

Name	Values	Description
Active variant	{1..10}	Specifies the active display variant. The variant can also be switched via the key <F2 Variant> in the menu of this display. Available variants are set in the variant editor in the "Properties" dialog.
D-correction table name	{Yes, No}	Specifies whether the NC configurator path of the D-correction table is displayed.

Name	Values	Description
Tool source	{CPL data, system data}	Specifies the data source to display the activated and preselected tool. System data (system data mode) The system data structure "SD.SysTool[channel]" is to be described in the channel with the PLC or NC program with location and sector of the active and preselected tool. Examples: SD.SysTool[1].ActTool.K2 = 2 SD.SysTool[1].ActTool.K1 = 1 SD.SysTool[1].PreTool.K2 = 5 SD.SysTool[1].PreTool.K1 = 1 CPL data (CPL mode) (not recommended) Displaying the tool number (TN): For each channel, one CPL variable of type INT is to be created for the active and preselected tool (e.g. in the "wmhperm.dat" file). Example: <code>DEF INT @ACTTOOL01;</code> It has to be written on the variable with the corresponding tool number (TN) from the PLC or NC. The "active tool" {ta} tool property is additionally to be set for the active tool. Important: If it is written on the CPL variable from an NC program, a WAIT instruction has to be programmed afterwards to update the display. Displaying the tool name (SKQ): For each channel, one CPL variable of type CHAR-ARRAY (string) is to be created for the active and preselected tool (e.g. in the "wmhperm.dat" file). Example: <code>DEF CHAR @ACTTOOL01(32);</code> It has to be written to the variable with the corresponding tool name (SKQ) from the PLC or NC. The "active tool" {ta} tool property is additionally to be set for the active tool. Important: If it is written on the CPL variable from an NC program, a WAIT instruction has to be programmed afterwards to update the display.
Edit tool source	{[profile file]}	Can only be edited if the "tool source" parameter is set to the value "CPL data". Opens the assignment file to define the CPL variables for the active and preselected tool. Changes in the file are only applied after an interface restart. If the CPL variable does not exist, nothing is displayed.
Tool changer	{chain magazine, turret}	Specifies the column labeling of the tool visualization. Chain magazine: Tool active, tool preselected Turret: Tool active, location active
Display unit	{Default, [mm], [inch]}	Measuring unit for the display

Tab. 6-25: Tool display - Properties

Variants [chapter "Position display - Properties - View" on page 78](#)

Tool correction display - NC configurator

NC configurator path	MACODA	Name
/GUI/TabDisp/PrecLinMetr	6020 00021	Number of decimal places for linear axes (metric)
/GUI/TabDisp/PrecRotMetr	6020 00021	Number of decimal places for rotary axes (metric)
/GUI/TabDisp/PrecLinInch	6020 00022	Number of decimal places for linear axes (inch)
/GUI/TabDisp/PrecRotInch	6020 00022	Number of decimal places for rotary axes (inch)

Tab. 6-26: Tool correction display - NC configurator

Tool correction display - Properties

View

Name	Values	Description
Active variant	{1..10}	Specifies the active display variant. The variant can also be switched via the key <F2 Variant> in the menu of this display. Available variants are set in the variant editor in the "Properties" dialog.
Number of characters for the tool name	{8...32}	Specifies the maximum character length of the tool name. This parameter is not effective if the parameter "additional information" is set to "No".
Tool source	{CPL data, system data}	Refer to chapter "Tool display - Properties" on page 84
Edit tool source	{[profile file]}	Refer to chapter "Tool display - Properties" on page 84
Additional information	{Yes, No}	Specifies whether additional information is to be displayed. An additional column with additional information is displayed.
Tool changer	{chain magazine, turret}	Specifies the line labeling of the tool visualization. Chain magazine: Tool active, tool preselected Turret: Tool active, location active
Display unit	{Default, [mm], [inch]}	Measuring unit for the display

Tab. 6-27: Tool correction display - Properties

Variants [chapter "Position display - Properties - View" on page 78](#)

Tool correction register display - NC configurator

NC configurator path	MACODA	Name
/GUI/TabDisp/PrecLinMetr	6020 00021	Number of decimal places for linear axes (metric)
/GUI/TabDisp/PrecRotMetr	6020 00021	Number of decimal places for rotary axes (metric)
/GUI/TabDisp/PrecLinInch	6020 00022	Number of decimal places for linear axes (inch)
/GUI/TabDisp/PrecRotInch	6020 00022	Number of decimal places for rotary axes (inch)

Tab. 6-28: Tool correction register display - NC configurator

Tool correction register display - Properties

View

Name	Values	Description
Active variant	{1..10}	Specifies the active display variant. The variant can also be switched via the key <F2 Variant> in the menu of this display. Available variants are set in the variant editor in the "Properties" dialog.
Display unit	{Default, [mm], [inch]}	Measuring unit for the display

Tab. 6-29: Tool correction register display - Properties

Variants [chapter "Position display - Properties - View" on page 78](#)

Input assistance display - NC configurator

NC configurator path	MACODA	Name
/GUI/TabDisp/PrecLinMetr	6020 00021	Number of decimal places for linear axes (metric)
/GUI/TabDisp/PrecRotMetr	6020 00021	Number of decimal places for rotary axes (metric)
/GUI/TabDisp/PrecLinInch	6020 00022	Number of decimal places for linear axes (inch)
/GUI/TabDisp/PrecRotInch	6020 00022	Number of decimal places for rotary axes (inch)

Tab. 6-30: Input tool display for NC configurator

Input assistance display - Properties

View

Name	Values	Description
System axis names	{Yes, No}	Specifies whether the system axis names are to be displayed in brackets behind the channel axis names. The parameter is only effective if the channel axis name differs from the system axis name.
Display unit	{Default, [mm], [inch]}	Measuring unit for the display

Tab. 6-31: Input tool display - Properties

Variants [chapter "Position display - Properties - View" on page 78](#)

Program coordinate display - NC configurator

NC configurator path	MACODA	Name
/GUI/TabDisp/PrecLinMetr	6020 00021	Number of decimal places for linear axes (metric)
/GUI/TabDisp/PrecRotMetr	6020 00021	Number of decimal places for rotary axes (metric)
/GUI/TabDisp/PrecLinInch	6020 00022	Number of decimal places for linear axes (inch)
/GUI/TabDisp/PrecRotInch	6020 00022	Number of decimal places for rotary axes (inch)

Tab. 6-32: Program coordinates display for NC configurator

Program coordinate display - Properties

View

Name	Values	Description
System axis names	{Yes, No}	Specifies whether the system axis names are to be displayed in brackets behind the channel axis names. The parameter is only effective if the channel axis name differs from the system axis name.
Display unit	{Default, [mm], [inch]}	Measuring unit for the display

Tab. 6-33: Program coordinate display - Properties

Variants [chapter "Position display - Properties - View" on page 78](#)

G-code display - NC configurator

NC configurator path	MACODA	Name
/GUI/NcCodeDisp/Ch[i]/NcCodeGr[j]	6005 00040	NC function for group

Tab. 6-34: G-code display - NC configurator

G-code display - Properties

View

Name	Values	Description
Groups per line	{2...15}	Specifies the maximum number of G-code groups per line. The parameter determines the number of lines.

Tab. 6-35: G-code display - Properties

M-code display - NC configurator

NC configurator path	MACODA	Name
/GUI/AuxFuncDisp/Ch[i]/AuxCodeGr[j]	6005 00040	NC function for group

Tab. 6-36: M-code display - NC configurator

M-code display - Properties

View

Name	Values	Description
Groups per line	{2...15}	Specifies the maximum number of M-code groups per line. The parameter determines the number of lines.

Tab. 6-37: M-code display - Properties

Precision correction display - NC configurator

NC configurator path	MACODA	Name
/GUI/TabDisp/PrecLinMetr	6020 00021	Number of decimal places for linear axes (metric)
/GUI/TabDisp/PrecRotMetr	6020 00021	Number of decimal places for rotary axes (metric)
/GUI/TabDisp/PrecLinInch	6020 00022	Number of decimal places for linear axes (inch)
/GUI/TabDisp/PrecRotInch	6020 00022	Number of decimal places for rotary axes (inch)

Tab. 6-38: Precision correction display - NC configurator

Precision correction display - Properties**View**

Name	Values	Description
System axis names	{Yes, No}	Specifies whether the system axis names are to be displayed in brackets behind the channel axis names. The parameter is only effective if the channel axis name differs from the system axis name.
Display unit	{Default, [mm], [inch]}	Measuring unit for the display

*Tab. 6-39: Precision correction display - Properties***Online correction display - NC configurator**

NC configurator path	MACODA	Name
/GUI/TabDisp/PrecLinMetr	6020 00021	Number of decimal places for linear axes (metric)
/GUI/TabDisp/PrecRotMetr	6020 00021	Number of decimal places for rotary axes (metric)
/GUI/TabDisp/PrecLinInch	6020 00022	Number of decimal places for linear axes (inch)
/GUI/TabDisp/PrecRotInch	6020 00022	Number of decimal places for rotary axes (inch)

*Tab. 6-40: Online correction display - NC configurator***Online correction display - Properties****View**

Name	Values	Description
System axis names	{Yes, No}	Specifies whether the system axis names are to be displayed in brackets behind the channel axis names. The parameter is only effective if the channel axis name differs from the system axis name.
Display unit	{Default, [mm], [inch]}	Measuring unit for the display

*Tab. 6-41: Online correction display - Properties***System message display - NC configurator**

NC configurator path	MACODA	Name
-	-	-

*Tab. 6-42: System message display - NC configurator***System message display - Properties****View**

Name	Values	Description
Diagnostic type "Setup diagnostics"	{Yes, No}	Specifies whether pending messages of the diagnostic type "Setup diagnostics" are to be displayed
Diagnostic type "Error"	{Yes, No}	Specifies whether pending messages of the diagnostic type "Errors" are to be displayed
Diagnostic type "Notes"	{Yes, No}	Specifies whether pending messages of the diagnostic type "Notes" are to be displayed

Name	Values	Description
Diagnostic type "Starting requirements"	{Yes, No}	Specifies whether pending messages of the diagnostic type "Starting requirements" are to be displayed
Diagnostic type "Warnings"	{Yes, No}	Specifies whether pending messages of the diagnostic type "Warnings" are to be displayed
Icon	{Yes, No}	Specifies whether the message is to be displayed with an icon. Refers to the diagnostic types "Errors", "Warnings" and "Notes"
Message type	{all, MTX, ProVi, SFC}	Specifies the message source or the message type ProVi Only ProVi messages are displayed MTX Only MTX messages are displayed (NC kernel, general diagnostics) SFCs Only SFC messages are displayed All All messages are displayed
Message exchange time	{500...5000}	Specifies the exchange time between the pending messages. If more messages are present than can be displayed, the messages are sorted according to their priority and then exchanged on a cyclic basis similar to a circular buffer. The parameter is not effective if the parameter "only latest message" is set to the value "yes"
Message lines	{1...20}	Specifies the maximum number of the messages displayed
Messages in color	{Yes, No}	Specifies whether the messages are to be highlighted in color when displayed
Only latest message	{Yes, No}	Specifies whether only the latest message is to be displayed. If the value of the parameter is set to "yes", the parameter has no effect on the "message exchange time"

Tab. 6-43: System message display - Properties

6.3.3 DBT1 screens (tool management)

General information

Screens providing different views on the tool stock in the tool database can be created under the "DBT1 Screens" node.

For the complete commissioning of the tool management including the creation of such screens, refer to [chapter 7.10 Configuring tool management, page 175](#).

Validating the "DBT1 screens (tool management)"

For a general description on the validation of visualization data in the HMI, refer to the HMI manual "DOK-IWORKS-HMI*Vxx****-APxx-EN-P"

Checked criteria *When validating the MTX tool visualization data, the following criteria are checked:*

- Specification and validity of the referenced screen elements
- Validity of the references F-/M-panel
- Validity of process variables
- Validity and existence of comments

Overview on error numbers **Each detected problem during the validation of the visualization data obtains an ID. Use this ID to find information on cause and remedy in the following table:**

ID	Cause	Recovery
MTX-00010301	List name is not unique	Rename the list name
MTX-00010302	The list name contains at least one of the following characters: \\ : * ! ? " ' < >	Remove the special characters from the name
MTX-00010303	The list is not used	The list can be used in a DBT screen or called in IndraWorks Operation if an F- or M-panel key is assigned with the list call "ToolList".
MTX-00010304	The tool list configuration file was not found	Open the editor and select a list name from the selection list
MTX-00010305	F-panel or M-panel invalid	Open the editor and select a panel from the selection list
MTX-00010306	The ID of the database query is invalid	Open the editor and select a database query from the selection list
MTX-00010311	The name of the tool list configuration file is not unique	Rename the configuration file
MTX-00010312	The name of the configuration file contains at least one of the following characters: \\ : * ! ? " ' < >	Remove the special characters from the name
MTX-00010313	The configuration file is not used	The configuration file in the tool list can either be used directly or it can be called via the TL_ChangeConfig command using an M-key or F-key
MTX-00010321	The database query was not specified	Open the "Query" editor and enter a query
MTX-00010322	The database query is not used	The query can be used in a DBT list or called in IndraWorks Operation if an F- or M-panel key is assigned with the list call "TL_ChangeQuery"
MTX-00010323	The specified token was not found in the language file	To ensure that a token exists in the language file, select a text in the dialog from the list of the provided texts.

MTX-00010324	The specified text list was not found	In the editor, it can be selected from the existing lists
MTX-00010325	The specified token was not found in the language file	To ensure that a token exists in the language file, select a text in the dialog from the list of the provided texts
MTX-00010331	The text "Token" is not available in the specified resource file	To ensure that a token exists in the language file, select a text in the dialog from the list of the provided texts
MTX-00010332	The specified process variable is invalid	Select a valid process variable using the selection dialog
MTX-00010333	The specified resource could not be loaded	Select a valid resource using the selection dialog
MTX-00010341	The name of the DBT screen is not unique	Rename the screen
MTX-00010342	The name of the screen contains at least one of the following characters: \ / : * ! ? " ' < >	Remove the special characters from the name
MTX-00010351	No complete schema data set was found to check process variables	Ensure that a control is assigned to the HMI and go online with this control

Tab. 6-44: Overview on the error IDs of the validation messages

6.3.4 SystemData screens (system data displays)

General information

Individual system data structures can be visualized in freely definable lists in IndraWorks Operation using "SystemData Screens". These lists are configured in IndraWorks Engineering under the "SystemData Screens" configuration node. Such a list is then embedded as screen call of type "MtxSystem-DataList" in one "screen" together with "M-panels" and "F-panels".

Configuring the system data display

A list configuration can be created in IndraWorks Engineering to visualize a system data structure via the "SystemData Screens" configuration node.



Only one SD variable can be visualized in one system data screen.

Commissioning tools

ULC configurator

See [chapter "ULC configurator" on page 183](#)

XML editor

See [chapter "XML file editor" on page 201](#)

Schema editor

See [chapter "Schema editor" on page 179](#)

System data configurator

To specify the system data definitions in IndraWorks Operation, a system data configurator is available.

Definition of terms

ULC	"ULC" stands for "Universal List Control" , which is the central element of the system data list configuration.
Sublist	"Sublist" is a central term in the schema definition for the ULC. A ULC is a table editor which can be able to multiply the configured line-column definition to present an array of system data structures according to the number of array elements. In this case, a sublist stands for the presentation of a system data structure. Thus, only one sublist has to be defined to display a multidimensional system data structure.

Handling instruction: Creating a system data screen

Creating a system data screen

None

1. Create an SD list configuration
New entry under "SystemData screens"
2. Create/edit the F-key panel of the SD visualization
3. Create a screen definition:
 1. <New screen>
 2. Select the "MtxSystemDataList" image type
 3. Configure the list definition
 4. It is not required to define the F-panel. The predefined F-panel is always used.
 5. Assign the screen to an operating area
4. Define the call of the SD.Screens in the operating area panel:
 1. Open the F-key configurator of the respective operating area panel
 2. Select "Image change" for the respective function key
 3. The newly created SD screen can be selected in the "Image name" box.
 4. Specify F-key labeling if required

Validating "SystemData Screens (system data displays)"

For a general description on the validation of visualization data in the HMI, refer to the HMI manual **"DOK-IWORKS-HMI*Vxx****-APxx-EN-P"**

Checked criteria	<i>When validating the visualization data of MTX system data screens, the following criteria are checked:</i> • Specification and validity of the referenced screen elements • Validity of process variables • Validity and existence of comments

Overview on error numbers

Each detected problem during the validation of the visualization data obtains an ID. Use this ID to find information on cause and remedy in the following table:

ID	Cause	Recovery
MTX-00010331	The text "Token" is not available in the specified resource file	To ensure that a token exists in the language file, select a text in the dialog from the list of the provided texts
MTX-00010332	The specified process variable is invalid	Select a valid process variable using the selection dialog
MTX-00010333	The specified resource could not be loaded	Select a valid resource using the selection dialog
MTX-00010401	List name is not unique	Rename the list name
MTX-00010402	The list name contains at least one of the following characters: \\ : * ! ? " ' < >	Remove the special characters from the name
MTX-00010403	The list is not used	The system data list can be used in a screen of type "MtxSystemDataList"

Tab. 6-45: Overview on the error IDs of the validation messages

6.3.5 List screens

Variable and parameter lists

General information

Variable and parameter lists can be used to display a list of different data of an MTX control in IndraWorks Operation. Supported are machine parameters, drive parameters, system data, CPL variables and PLC variables.



Variable and parameter lists are exclusively created and edited on the operating station in IndraWorks Operation!

The configuration of the variable and parameter lists is described in detail in the "Standard NC Operation" manual!



In IndraWorks Engineering, this data is archived, added to version control as well as exported and imported.

Data exchange between IndraWorks Operation and IndraWorks Engineering is started via the context menu of the HMI node.

- IndraWorks Operation --> IndraWorks Engineering: Visualization Data \ Loading changes from operating station
- IndraWorks Engineering --> IndraWorks Operation: Visualization Data \ Transfer and Activate

PLC list view (PLC variable lists)

General information

PLC variable lists show the contents of a global variable list from IndraLogic without including them in the symbol configuration! Additionally, they provide numerous sorting and filter options as well as detailed information on each

variable. Using the corresponding function block, it can be permanently written to values by forcing.



PLC variable lists are exclusively created and edited on the operating station in IndraWorks Operation!

The configuration of the PLC variable lists is described in detail in the "Standard NC Operation" manual!



In IndraWorks Engineering, this data is archived, added to version control as well as exported and imported.

Data exchange between IndraWorks Operation and IndraWorks Engineering is started via the context menu of the HMI node.

- IndraWorks Operation --> IndraWorks Engineering: Visualization Data \ Loading changes from operating station
- IndraWorks Engineering --> IndraWorks Operation: Visualization Data \ Transfer and Activate

Channel overview (channel dialog)

General information

The channel dialog shows the status of multiple channels in the configurable overviews and provides the option to switch channels. A channel filter is used to configure this overview.



The filter settings of the channel dialog are exclusively created and edited on the operating station in IndraWorks Operation!

The channel dialog is described in detail in the "Standard NC Operation" manual!



In IndraWorks Engineering, this data is archived, added to version control as well as exported and imported.

Data exchange between IndraWorks Operation and IndraWorks Engineering is started via the context menu of the HMI node.

- IndraWorks Operation --> IndraWorks Engineering: Visualization Data \ Loading changes from operating station
- IndraWorks Engineering --> IndraWorks Operation: Visualization Data \ Transfer and Activate

6.3.6 Programming screens

NC block sequences

The **NC block sequences** are a collection of templates. Each sequence can consist of one or multiple NC blocks up to a complete program. These can be directly be inserted into the NC program when programming.



NC block sequences can exclusively be executed on the operating station in IndraWorks Operation!

NC block sequences are described in detail in the "Standard NC Operation" manual!



In IndraWorks Engineering, this data is archived, added to version control as well as exported and imported.

Data exchange between IndraWorks Operation and IndraWorks Engineering is started via the context menu of the HMI node.

- IndraWorks Operation --> IndraWorks Engineering: Visualization Data \ Loading changes from operating station
- IndraWorks Engineering --> IndraWorks Operation: Visualization Data \ Transfer and Activate

NC cycle input mask

General information

Input masks allow a dialog-guided changing of selected parts of an NC program



Input masks can exclusively be created and changed on the input masks!



In IndraWorks Engineering, this data is archived, added to version control as well as exported and imported.

Data exchange between IndraWorks Operation and IndraWorks Engineering is started via the context menu of the HMI node.

- IndraWorks Operation --> IndraWorks Engineering: Visualization Data \ Loading changes from operating station
- IndraWorks Engineering --> IndraWorks Operation: Visualization Data \ Transfer and Activate

Input mask

Definition An "input mask" is a dialog with a determined layout. In such as dialog, values (parameters) are recorded in tabular form (insertion of new calls into an NC program) or provided for correction.

Input masks also provide information on an output format. The output format establishes a connection between the program text and a specific input mask and between syntactic units and the individual lines of the input table (see chapter [Syntax of mask definition - Output format, page 120](#)).

Objective Input masks can be used for a controlled editing of selected parts of the NC program text without using a user documentation. The programmer is supported by graphics, texts and integrated value validation mechanisms. The input masks can be directly accessed from the NC program editor (see "MTX Standard NC Operation", "Inserting functions ...").

NC blocks can be added and corrected using input masks. Moreover, comments initiated with a semicolon and functional comments on the NC Simulation can also be added and edited. However, editing CPL blocks is not possible.

Configuration After the installation, IndraWorks Operation is already provided with a high number of input masks for standard cycles and other instructions. For individual cycles or for instructions to be frequently programmed, it can be required to define input masks (using an external text editor) or to import existing input masks.

Setting up cycles on the NC

Due to the main purpose of input masks, there is a close relation to the cycles. Thus, it is noted here that it could be necessary to set up cycles in the control. Refer to the chapter [7.9 Setting Up NC Cycles, page 173](#).

The following chapters provide a definition of input masks. The syntax on which the definition is based, enables the user and machine manufacturer to create their individual masks without any knowledge of a higher programming language. The visible dialog is generated from the mask definition.

The following sections contain information on maintenance and error analysis.

Signature of calls and overlapping of input masks

Call components

A distinction is made between the following components of a DIN block when a call is compiled automatically:

- Unchangeable call names (subroutine names and DIN commands) followed by a parameter list
- Parameters (either in brackets as parameter list or without brackets as DIN words)
- Other unchangeable block components without a parameter list (outside the brackets)
- Comments (also comment lines, initiated with a semicolon)



The term "parameter" has a broader meaning with regard to input masks than with regard to the NC. As seen from the input masks, each block component with an input option is changeable and therefore a parameter. From the NC point of view, a parameter can also only be a command word, such as a parameter "Spindle command" with the values M3, M4, M5, etc.

Signature

The unchangeable block components, i.e., call names and constant DIN words outside the brackets, form the signature of a call. Special characters are not considered.

Example:

Signatures

- The first example includes the signature "G81".
`G81 (IX X, SL2, DT-30, RL15); Twist drilling`
- The second example includes the signature "T M6".
`T[4] M6`
- In the third example, the signature is "TOOL STANDARD".
`//TOOL/STANDARD,93,55,0.8,10,3\\ [""+NCF("G90")]`

Signature function for the correction

The signature makes a call unique in the NC editor. The default case for input masks provides that unchangeable block components are uniquely assigned to a mask. Therefore, it uses the specific feature of signatures, its uniqueness. That means that there is only one input mask for each signature. The NC program editor is responsible. However, the input mask engineer can also bypass the rule of unique assignment via the signature in multiple ways. Therefore, it is also possible to provide multiple masks for one signature.



If a call has a signature, only the input masks are considered for correction purposes, since their signature is at least partially in the program line on which the current cursor position indicates. If no signature part is in the current line, corrections via input mask cannot be started from there (comment line, blank line, line without call word and without unchangeable block components). In this case, go to a line containing at least one word from the signature.

Signature function in case of overlapping input masks

As guaranteed by the system by default that only one input mask exists for each known signature, the definition of an input mask with a signature already registered causes the known input mask to be overwritten. This principle is used to allow the machine manufacturer to replace the standard cycles

and the related input masks with his own ones. Moreover, the user also can replace Bosch Rexroth input masks or input masks provided by the machine manufacturer by self-defined input masks.

Overlapping concept

In the overlapping concept, the input masks are provided in a defined hierarchy according to their origin.

The following sources are distinguished:

- User-specific input masks
- Tool-specific input masks
- Default input masks (Bosch Rexroth)

In terms of overlapping, the user masks have the highest priority. The masks of the machine manufacturer have a higher priority than those of Bosch Rexroth. It overlaps only if the signature of the corresponding input masks is identical.

The overlapping concept is the basis for the input mask adaptation with regard to its content and partially availability.

Input masks without signature

The system also restrictively define the input masks, which are only provided with changeable components in their output format. These are calls only consisting of parameters and comments. During the correction, the assignment of an NC block to an input mask is no longer unique and is performed by taking the parameters into account. During the correction, the input mask most similar to the found parameters is provided.



1. Input masks without a signature are not subject to overlapping, i.e., each input mask without a signature continually increases the stock of input masks.
2. Only parameters with preceding addresses can be used in input masks without signature. The only parameter type allowed without address is the enumeration.
3. As only "soft" criteria are used when assigning NC blocks to input masks without signatures, the probability is significantly greater that several input masks with the same degree of concordance are considered for correction. For input masks without signature, always enter as many parameters possible to provide a unique mask assignment for the correction in case of input masks without signature.

Advanced systematization of calls

In addition to the differentiation criteria of calls based on the existence of a signature explained in the previous section, note the following:

Single line calls

All fixed block components and all parameters are in one line in case of a single line call. In addition, only comment lines initiated with a semicolon can be part of the call format.

Single line calls are the usual calls.

Multiple line calls

With multiple line calls, the unchangeable block components and parameters are distributed across several lines. The call can be supplemented by comment lines initiated with a semicolon.

Multiple line calls are primarily used for graphic NC programming (GNP), specifically with geometry definitions (see "MTX Workshop Programming Turning and Milling", chapter "Geometry definitions").

Parameter composition

Parameters can be specified as follows:

1. As DIN parameter list.

The list is enclosed by round brackets. Parameters are separated by a comma. The DIN syntax applies, which means, for example, that a character string without inverted commas is expected. For further information on parameter lists, refer to the "MTX Programming Manual ", chapter "Parameter transfer to subroutines". CPL expressions and the reference to CPL variables are only possible by enclosing the expression in square brackets (see "MTX Programming Manual ", chapter "Labeling CPL elements within a part program"). Such CPL elements within a DIN line are also referred to as **Inline CPLs** .

The functional comments on the NC simulation only initiated with "///", are to be considered in this context like a DIN parameter list, although the Inline CPL is not allowed. Refer to the "MTX Workshop Programming Turning and Milling ", chapter "Additional simulation information in the part program" for functional comments on the simulation.

2. As CPL parameter list

The entire parameter list is displayed as Inline CPL in square brackets. The parameters are separated by a comma. Each parameter value can be formulated as a CPL expression without additional brackets. A character string constant is to be enclosed with inverted commas.

3. As a sequence of DIN words.

The parameters are directly in the line and are not summarized in a list. The parameters are separated by blank spaces, special characters (";", "(", ")"), "[", "]") or with a letter following a character that is not a letter. These parameters are subject to the DIN syntax. The value assignment of addressed parameters can also be performed by CPL elements within a DIN block (Inline CPL).

Address parameters

The parameters can be provided with addresses. The significance of a value is defined by the preceding address name. Programming an address parameter is like the assignment of a value to a variable. Separator characters (spaces or assignment operators) are optionally inserted between the address and the value. Address parameters can occur in address parameter lists and in the line as DIN word (e.g., a coordinate).



Address names have to be unique within each parameter list and also outside a parameter list. Only for address names "G" and "M", duplicates of the address name are allowed if additional value range limits are defined. This is required to program different G-commands or M-commands belonging to different modal G-groups or M-groups in one line (using the parameter types for integers and real numbers).

Example:

Address parameters

G1(RND2.5, FL150) X200 Y-26

If all numbers of the example are seen as parameters and thus as changeable, "RND", "FL", "X" and "Y" act as parameter addresses.

Parameters without addresses

Parameters without addresses can be used in a parameter list. The significance of a parameter is generally defined by its position within the list (counting from left to right starting from the open bracket). Therefore, these parameters are also known as **position parameters**. If individual parameters are not assigned, several commas have to be written without parameter value to

maintain the assignment of subsequent parameters according to their position.

Parameters without addresses can also occur outside a parameter list. These are only DIN commands. The NC receives its significance from the word as usual.

Example:

Parameters without addresses

Position parameters: MYCYC["X", 2,,, -3.75]

The parameters 1, 2 and 5 are assigned. The parameters 3 and 4 are not assigned.

Parameters without address outside a parameter list:

G97 S2500 M3

If "G97" were considered as parameter and the G group would be assigned to it for spindle velocity, "G96" would also be added to the value range. As "G" is included in the value, there is no address left. This parameter is then without an address outside a parameter list. The same can be performed with spindle control command "M3". This parameter for the spindle command receives the value range M3, M4, M13 and M14 for example.

No value range listing of possible values can be specified for the S-value. The parameter for the cutting velocity or speed is therefore converted into an address parameter.

File structure - Overlapping concept in file structure

File structures are provided to implement the overlapping concept for input masks. These structure provide a storage location for each origin named.

User input masks

All files defining user masks and all further files which can be used as resources (data sources), are stored in the IndraWorks project in the folder <Visualization Device>\user\config\cycles.

Input masks of the machine manufacturer

The definition of the masks of the machine manufacturer and the resources are stored in the IndraWorks project in the folder <VisualizationDevice>\OEM\config\cycles.

Input masks of Bosch Rexroth

Bosch Rexroth input masks are provided by being installed in the Rexroth\IndraWorks\config\cycles folder. The text files to support multiple languages are stored in the Rexroth\IndraWorks\config folder.

Search strategy for resource files

Access to files, such as graphics, texts etc., is based on a specified search strategy. The file is initially searched for under

<VisualizationDevice>\user\config\cycles.

If the respective file is not found, it is searched in the

<VisualizationDevice>\OEM\config\cycles folder and finally in the particular folders for the installed input masks. This file search strategy allows selective overlapping of individual resources or re-using existing resources in user masks.

Online helps are searched in the language-dependent subfolders of the named folders

(<VisualizationDevice>\user\config\cycles\Help\<CountryCode>, etc.).

File structure - Mask definition files

Function of the mask definition file

The mask definition file specifies the visible content of one or several input masks (texts, graphics and helps), the behavior (checking type and value

	range of the parameters, default values, etc.) as well as the output format of the call. A mask definition file is an ASCII text file with the extension ".cyc" which has to be sufficient for the syntax for input masks described in the following.
Considered mask definition files	All files with the extension ".cyc" are considered in the folders reserved for the masks of the machine manufacturer and the user masks. The Bosch Rexroth input masks are taken from the following files of the Rexroth\IndraWorks\config\cycles folder: • MTX.Cycles.Canned.cyc: Masks for standard cycles and others • MTX.Cycles.GnpMill.cyc: Masks for GNP programming, parts milling and drilling • MTX.Cycles.GnpTurn.cyc: Masks for GNP programming, parts turning • MTX.Cycles.Simulation: Programming of functional comments on the NC simulation
Creating and editing mask definition files	The mask definition files can be edited using a simple external text editor such as WordPad.exe. Note that the editor can process ASCII files and that they use this file format only while saving. The NC text editor of IndraWorks Operation can also be used if the file was imported and then exported again (file handling functions in the Project Explorer).
How many masks should be combined in a definition file?	Any number of input masks can be combined in a .cyc file. If masks are to be transferred independently from others to another machine tool, the masks to be transferred together should be in one file. We therefore recommend to combine several masks with related topics in one file each.
File structure - Graphic files	
Graphic formats	<i>The following file formats are supported for image files:</i> • *.jpg, • *.bmp, • *.gif • *.tif. A ".gif" format is recommended for drawings and ".jpg" for photos. A standardized size of 223 x 262 pixels (width x height) is provided for an optimum display of the graphics. If images are large, sections of the images are cut off. The graphics can be created and edited using any graphic software which supports the mentioned file formats.
No graphics defined	If there is no graphic file or if a graphic file was not provided for an input mask, the IndraWorks\config\C0.jpg is displayed. This file can also be used as template for individual images.
File structure - Support of multiple national and regional languages	
	The texts for headings, parameter titles and explanatory information within the input mask can be provided in one or several languages, that is in multiple national or regional languages. In case of multilingualism, it can be switched between these languages. This takes place simultaneously when changing the set GUI language. In general, the online helps can be provided with several languages.

Accessing a language-dependent text	Monolingual texts are directly written into the mask definition file. The available languages are significantly limited as this file is in ASCII format. Thus, use preferably language-dependent texts.
	One text file can be assigned to each input masks. This file contains all texts of the input mask and all comments to be added. A text to be represented/added is referenced via a token. A token is the unique reference to a specific text in a language-dependent text file. In this case, it consists of a number greater than zero with one to four digits. In the mask definition file, it is accessed with "%TNxxxx%" placeholders. xxxx stands for the maximum four-digit token number.
Language variants	A text file can be present in multiple languages. A separate file has to be created for each language. The name structure is subject to the following rule: <langfile>_<countrycode>.txt with <langfile> constant part of the text file name (specified in the input mask definition) and <countrycode> comprises two letters for the language variant according to ISO 639-1 e.g.: "DE" for German, "EN" for English, "SV" for Swedish, "ZH" for Chinese, "KO" for Korean, etc. At least one text file has to be provided for English (<langfile>_EN.txt). English is the default language if there is no text file for the currently set interface language.
Text file not available in the current language	First, the text file is specified with regard to the current interface language according to the search strategy (see chapter File Structure - Overlapping Concept in the File Structure, chapter 101). If the text file is not available in the current language, the respective English text file and then the German text file is searched analogously.
File structure for language-dependent texts	Language-dependent texts have to be provided in an ASCII file or in an UNICODE file. Each token starts as follows: <pre> <tnr>\----\0000\</pre> with <tnr> is the token number (up to four digits, leading zeros permitted). The token numbers do not have to be arranged in ascending order and may have gaps. The respective text is to be provided in the following line. If it is a multiline text, one line in the text file is to be provided for each line of the text. At the beginning of the text file, the token number 0 is to be displayed and used in case of an error if the addressed token does not exist. <i>Example:</i> File for language-dependent texts (in this case the German language variant "MyText_DE.txt") <pre> 0000\----\0000\ No text defined! 0001\----\0000\ Retract height 0002\----\0000\ Line 1 of a multi-line text Line 2 of a multi-line text</pre>

Creating and editing text files	Language-dependent text files are created and edited by the user or by the machine manufacturer using a simple, external text editor (e.g., Word-Pad.exe). An independent provision mechanism exists for Bosch Rexroth text files.
How many text files should be provided?	Theoretically, a separate text file can be provided for every input mask. It is the recommended method since masks rarely allow re-using texts. Provide at least one text file for every cyc-file in order not to unnecessarily limit the option of transferring single input masks or input masks with related topics to other machines.
Help files in several languages	Help files for the individual languages are assigned with a mechanism similar as for the text files. The country codes are not contained in the file name. Instead, subfolders are provided which are named according to the country code (<VisualizationDevice>\user\config\cycles\Help\<CountryCode> and <VisualizationDevice>\OEM\config\cycles\Help\<CountryCode>). If the help file is not available in the respective language, first the English file is searched and then the German one.
Syntax of the mask definition - Structure of a mask definition	
Parentheses	Each input mask starts with a <pre>//%CHBEGIN%</pre> line and completes with a <pre>//%CHEND%</pre> line.
Mask elements	Mask elements (graphics, parameters, output formats, explanatory texts, titles) are to be defined between these two lines, wherein each of the lines relevant to the definition starts with "//%" and writes on one mask element. All other lines are not considered. Preferably initiate comments with a semicolon as in an NC program. The sequence of mask elements of a category determines the sequence in the input mask. Parameters are provided from top to bottom in the sequence as specified in the definition file for example. Mask elements of different categories can be listed in any order and mixed line by line with other categories.
Sequence of input masks	 The sequence of mask elements of the same type is decisive for the structure of the input mask, but the sequence of the masks in the definition file has no effect on the sequence provided in the selection list while inserting. The default sequence is defined by the alphabetically ascending sorting of the signatures, wherein G- and M- commands are sorted in a numerically ascending order. The sequence can be modified in a dialog to adjust the input support by shifting.
Basic information on the syntax	The syntax fundamentally consists of reserved language words and placeholders enclosed in percentage symbols and constants (values, names). The reserved language words in percentage symbols determine the interpretation of the subsequent words to a large extent. At least one separating space is required between individual words, although not before and after reserved words in percentage symbols. Names with spaces to be understood as a word (e.g. file names) can be provided with inverted commas.

Syntax of the mask definition - Reference to a language-dependent text file

If an input mask is to support several languages, a reference to a text file has to be included in the mask definition.

Syntax `//%LANG% <langfile>`
with
<langfile> Name of the text file without two-digit language code according to ISO 639-1, without extension and without path specification. Enclose the name with inverted commas if the name contains spaces.

Example:

Language-dependent text file

Reference to the MyText_EN.txt, MyText_DE.txt, MyText_RU.txt etc. text files.

`//%LANG% MyText`



There is also an older procedure to define language-dependent text files which are still supported due to compatibility reasons.

`//%ID% <langfile>`

Syntax of the mask definition - Title

Define a title in each mask definition.

Syntax `//%NAME% <title>{ %SIGN% <sign>{ <sign2>...}}`
with
<title> Title of a cycle.
The title is displayed for selection when adding and is therefore to be unambiguous for the NC programmer. A language-dependent title can be achieved by specifying a token instead of the text. A token is specified using `%TN<tno>%`, where `<tno>` is the token number of up to four-digits in the language file (e.g. `%TN0123%`, see [chapter File Structure - Support of multiple national and regional languages, page 102](#)).
<sign> <sign2>... Apart from the call signature specified in the output format, the signature can also be extended. `<sign>` is a word consisting of letters, digits and "_". Separate multiple additions from each other by spaces. For more information on the signature, refer to [Signature of Calls and Overlapping of Input Masks, page 98](#).

What is the purpose of the signature addition?

The unique assignment of input masks to a program text via the signature is the basis for mask overlapping. In exceptional cases, it can be desired that several input masks with the same signature have to be kept in parallel.

This is used as follows:

- If multiple masks with different parameters are to be declared to take different cases into account, but have identical call names (e.g., different masks for G1 for a straight point-to-point move and a straight three point move).
- For masks without signature. In case of cycles without signature, the alphabetical order is solely defined by the signature addition

The signature addition is subordinate to the call signature. Therefore, the alphabetical sorting via the additional signature can only be changed to a minimum extent for masks with signature.



Use only character strings not allowed as DIN words as additional signatures. Combine the signature with "_".

Example:

Title

```
//%NAME% G81 Twist drilling
```

Monolingual title

```
//%NAME% %TN0678%
```

Multilingual title with token specification

```
//%NAME% G1 - Three-point cycle %SIGN% 3_PNT_DRAFT
```

Signature addition

Syntax of mask definition - Properties of an input mask (grouping)

Objective Each input mask can be provided with properties used by the system when creating the list of the masks available for insertion (menu scope). The specification of the same property word in several masks declares all these masks to be elements of a specific subset, the group.

Properties are differentiated allowing a grouping with regard to the following criteria:

- Thematic (contentual) or technological points of view
- Origin (user, manufacturer, standard cycle or software option)
- Availability of the input mask when inserting and correcting

The property on the origin is automatically specified by the system and does not have to be programmed.

Syntax `//%GROUP% <property>{ <property2>{ <property...>}}`
with

<property> <property2>... One word consisting of letters, digits and "_", the group name standing for a specific property. According to the table below, there are multiple predefined group names in the system. The machine manufacturer and the user can introduce any number of new group names using %GROUP% instructions that are always understood as name of a thematic/technological group. Multiple properties are separated by spaces.

Properties table

List of predefined properties

Group name	Automatic assignment	Property
CANNED	Yes, when reading from IndraWorks\config\cycles\MTX.Cycles.Canned.cyc	Origin "Standard cycle" (Bosch Rexroth)
DIN	No	Topical grouping "DIN instructions" (masks supporting DIN programming do not represent cycle calls)
DRILL	No	Grouping for "drilling" technology
GNP	Yes, when reading from IndraWorks\config\cycles\MTX.Cycles.GnpMill.cyc and MTX.GnpTurn.cyc	Assignment to the software options of the graphic NC programming. Type code SWS-MTX***-RUN-NNVRS-D0-BAZ1 or SWS-MTX***-RUN-NNVRS-D0-TURN1
GRAVE	No	Grouping for "engraving" technology

Group name	Automatic assignment	Property
MILL	No	Grouping for "milling" technology (except engraving and contour milling)
MILLCONT	No	Technology grouping for "contour milling"
OEM	Yes, when reading from the project folder <VisualizationDevice>\OEM\config\cycles	Origin of "Machine manufacturer"
PATT	No	Thematic grouping for "bore pattern"
PROBE	No	Thematic grouping for "measurement"
SAMPLE	No	Thematic grouping for "examples"
SIM	Yes, when reading from IndraWorks\config\cycles\MTX.Cycles.Simulation.cyc	Input masks for functional simulation comments belonging to the NC simulation software options.
TURN	No	Grouping for "turning" technology (except contour turning)
TURNCONT	No	Technology grouping for "contour turning"
USER	Yes, when reading from the project folder <VisualizationDevice>\user\config\cycles	Origin of "User"
_HIDDEN	No	Hides an input mask. This mask is available neither for insertion nor for deletion
_NOCORR	No	Hides a mask for correction. This mask is only available for insertion
_NOINSERT	No	Hides a mask for insertion. This mask is only available for correction. (*)
_NOGROUP	Yes	This group gets each input mask automatically if it does not belong to any contentual/technological group. Never include it in a %GROUP% instruction
_OTHERGROUPS	No	This group name represents all groups not explicitly controlled via an F-key in the input support. This group name is used as filter for the function key "More cycles...". Never include it in a %GROUP% instruction

(*) Assign this property to all masks for which newer calls exist with other masks (e.g., in case of syntax change). This ensures that all newly inserted calls are based on the new pattern and the old masks are still available for correction.

Tab. 6-46: Composition of predefined properties

Example:

Properties (grouping)

The properties of a user cycle for drilling are to be defined.

```
//%GROUP% DRILL
```

As the cyc-file is saved in the <VisualizationDevice>\user\config\cycles project folder, the property "USER" for "User cycle" is automatically assigned to the mask.

*Example:***Hiding the input mask**

A machine manufacturer likes to hide G111 Hole circle rotary axis, since the machine is not provided with rotary axes. Therefore, the cycle mask of the standard cycle has to be hidden.

```
//%CHBEGIN%
//%GROUP% _HIDDEN
//%N% G111(%P%)
//%CHEND%
```

The machine manufacturer adds this mask definition to a cyc-file stored in the project folder <Visualization Device>\OEM\config\cycles.

Global mask elements - Graphics

Global mask elements are graphic dialog elements generally displayed as long as the assigned input mask is visible. Only the global graphics can be temporarily overlapped by a parameter-specific graphic.

Objective The graphic is particularly significant for the mask design, as it is a key to explain the function of a cycle and the parameters. In principle, it is not possible to incorporate long explanatory texts into an input mask. Therefore, all possible relevant information has to be provided graphically.

Multiple global graphics It is possible to assign multiple global graphics to an input mask. Use this option to display different machining cases (e.g., internal and external drilling) or dimensioning variants separately and thus more clearly.



Graphics are displayed independently of the set language. Thus, avoid texts in the graphics except for parameter names and, if necessary, an explanatory program code.

Syntax `//%GF% <graphicfile>{ <graphictitle>}`

with

<graphicfile> Name of a graphic file without path including the extension ".bmp", ".jpg", ".gif" or ".tif". Size of the images 223 x 262 pixels (width x height). Enclose file names with spaces in inverted commas.

<graphictitle> If several graphics are displayed, a title has to be assigned to each graphic. This title is displayed or can be selected in the list above the graphic. The title includes the entire text after <graphicfile> up to the end of the line. To support multilingualism, a token can also be specified ("%TNxxxx%", see [File Structure - Support of Multiple National and Regional Languages, page 102](#)).

If only one graphic is provided for a mask, it is recommended to omit the graphic title.

*Example:***Global graphics**

Assign two global graphics to one cycle - one graphic for the internal machining and one graphic for the external machining.

```
//%GF% TurnOutside.gif external machining
//%GF% "Turn Inside.gif" %TN0346%
```

Global mask elements - Explanatory texts

Explanatory texts can be displayed in the lower section of the input mask.

Objective	The explanations provide an overview on the syntax and on the effect of the cycle. Special features are also to be described there, such as for the default parameters.
Syntax	<code>//%C% <explanation></code> with
<explanation>	Content of an explanatory line or a token. A monolingual text starts at the first character different from the space following <code>//%C%</code> and extends to the end of the line. Thus, no indentation can be programmed. A language-dependent text referenced via token can also comprise several lines and indentations (also refer to File Structure - Support of Multiple National and Regional Languages, page 102).

Example:

Explanatory texts

```
//%C% %TN0672%
```

```
//%C% default for parameter <ax>: Drill axis for current  
interpolation plane
```

Global mask elements - Online help

Help file	Each input mask can be assigned to one help file only (preferably HTML help). The help can be provided in several language variants. For information on storage location and support of several language, refer to Help Files in Multiple Languages, page 104 .
Syntax	<code>//%HELPPFILE% <helpfile></code> with
<helpfile>	Name of a help file including extension and excluding path specification. The file name extends to the end of the line so that no inverted commas are required for spaces in the name.
Help token	A help token can optionally be specified for each input mask. A help token is the reference to a specific section in the online help.
Syntax	<code>//%HELPTOKEN% <helptoken></code> with
<helptoken>	Identifier of a reference to a specific section in the help file. This can either be a topic ID such as "G86Boring.htm" or a keyword (search term). If your help is translated into other languages, note that the selected keyword is identical for all languages and thus not translatable.

Example:

Online help

The help file "MyCycles.chm" is to be displayed in an input mask. The corresponding section is to be reachable via the keyword "CYCLE2".

```
//%HELPPFILE% MyCycles.chm
```

```
//%HELPTOKEN% CYCLE2
```

Syntax of mask definition - Default parameter types

Syntax	The following describes the parameter declarations as to be found in almost every input mask. There are also special types subject to extra syntax and therefore documented separately. <code>//%P%{%%<pname>}{ <type>{ [<range>] }}<ptitle></code>

```
{ %D%{<behaviour>}<default>}}{ %GF%<graphicfile>{ <graphic-
title>}}
{ %VALID%<validinfo>}
```

The syntax is explained in the following sections. Note that each parameter description has to be written into one line even if the syntax is displayed across several lines due to the space.



- Define the sequence of the parameter definition based on a reasonable procedure and not on the sequence of the position parameters. Parameters directly related or dependent on each other are defined successively.
- The method of defining parameters up to using "//%V%" is continuously supported for reasons of compatibility.

Title <ptitle>

A title is assigned to each parameter. This title has to include the parameter description in plain text along with any possible supplementary information on the unit of measurement etc. The parameter title can specify multiple languages as token for support purposes ("%TNxxxx%", see [File Structure - Support of Multiple National and Regional Languages, page 102](#)). No percentage sign can be used for a monolingual text. However, if it is required as measuring unit for example, enclose it with round or square brackets.

A constant title text extends to the end of the line or to the next percentage sign outside a bracket.

Parameter address <pname>

If the parameter is provided with an address, the address name is preferably to be specified directly behind "//%P%". The first word after "%@" is applied as address name. The address name is displayed in the first column of the input table. Otherwise, a continuous index is displayed.



Address names may only consist of letters, digits and "_", but not of special characters. If NC notes are to be programmed via an input mask ("character string" parameter type, see also [Character String STRING, page 113](#)), only the following syntax variants can be used:

- Channel-specific note (MSG<NoteText>):
Address name "MSG" or
- Channel-comprehensive note (GMSG<NoteText>):
Address name "GMSG".

For NC notes, refer to "MTX Programming Manual", chapter "Notes in the User Interface".

Variable address name (GNP)

If an output format begins with a parameter of the special type "DCS" (description coordinate system from graphic NC programming defining the coordinate axes active in the cycle) the significances of the axis names specified in the DCS can be used again in the subsequent parameters. The axis names become variable. That is achieved by the following placeholders instead of the constant address name:

%A1%

Name of the master axis of the plane (abscissa).

%A2%

Name of the slave axis of the plane (ordinate).

%A3%

Name of the drilling axis of the plane (applicator).

For more detailed information on the description coordinate system, refer to the "MTX Programming Manual", chapter "Geometry Definitions".

Example:

Parameter address and title

Position parameters with monolingual title

//%P% Contact width [%]

The coordinates of the first DCS master axis with a multilingual title are to be programmed as reference coordinate of a geometry definition.

//%P% %%A1% %TN3742%

Parameter types and value ranges

A data type **<type>** can be optionally assigned to each parameter. The specification of a data type can - and has to for the enumeration type - follow the specification of a value range **<range>** in square brackets.

The data type is to follow directly after the address name or after "%P%".

The following data types are possible:

Real number REAL

The "real number" parameter type is programmed using "REAL". This is also the default data type if a parameter declaration does not include any type specification.

The optional value range specification **<range>** consists of a lower value range limit followed by two points and the upper value range limit.



The specified value range limits are always within the value range. If a value range limit is not within the range, switch to a fraction number close to the real range limit. If the value range limit is an integer, enter as limitation a value either 0.01, 0.001 or 0.0001 smaller or greater than the integer already outside the value range.

If the lower or the upper value range limit is not required (value range limited on one side), change to a very small or very large number. The lower value range limit is canceled with -999, -9999 or -99999, the upper one with 999, 9999 or 99999.

Example:

Parameters of type "REAL"

A real parameter ANG should have the value range $0 \leq \text{ANG} \leq 360$.

//%P% %%ANG REAL[0..360] angle [deg]

A real position parameter should be greater than 0.

//%P% REAL[0.0001..99999] distance

A real parameter A should move in the range $90 < A < 180$.

//%P% REAL [90.001..179.999] angle [deg]

Integer INT

For the "integer" parameter type, "INT" is to be specified.

The optional value range specification **<range>** consists of a lower value range limit followed by two points and the upper value range limit.

Example:

Parameters of type "INT"

An integer parameter TEC should be provided with the value range $0 \leq \text{TEC} \leq 9$.

```
//%P% %%TEC INT[0..9] Edge position
```

An integer number position parameter should be greater than 0.

```
//%P% INT[1..9999] Number of bores
```

Binary type BOOL

The "binary type" represents a logic statement which can only apply two values:

"FALSE" or "TRUE".

This parameter type is to be declared with "BOOL". It can only be used in In-line CPL parameter lists (parameter list in square brackets) and applies the values "TRUE" and "FALSE" according to CPL syntax.

The optional value range specification **<range>** is used to display plain text for the meanings "FALSE" and "TRUE". First, the text is given for "FALSE" and then, separated by a comma, for "TRUE". The texts can be defined as token or monolingual text, although no comma and no "]" are permitted for the latter.

Without any range specification, "TRUE" and "FALSE" are displayed as possible values in the input table.

Example:

Parameters of type "BOOL"

For binary parameters, the text is to be displayed "on the right" if the state is "TRUE" and "on the left" if the state is "FALSE".

```
//%P% BOOL[let, right] direction of rotation
```

The same in the multilingual version:

```
//%P% BOOL[%TN0560%, %TN0561%] %TN0562%
```

ENUM enumeration type

If a finite number of words, numbers or strings belong to the value range of a parameter, the "ENUM" enumeration type is often used. In this case, each element of the value range is defined separately including a corresponding plain text. Indicate the value first for each element followed by a plain text separated by a space. The plain text can be declared by a token or can be monolingual, although no comma and no "]" may be included in the text. A comma is the separator between two elements.



- If the enumeration contains integers (see [Integer INT, page 111](#)), consider whether the type "INT" is not more suitable, since editing takes place in a selection list which is more difficult to operate without a mouse than a simple text field.

The following criteria are in favor of ENUM:

- The value range has interruptions
- or –
- The values are not assigned to memorable meanings (e.g. machining variants).

The following criteria are in favor of INT:

- The value range has no gaps
- The value meaning can be represented more graphically than with a short plain text
- The "ENUM" type is the only valid data type for parameters without address that are displayed outside a parameter list. In this case, each value has to start with a letter.
- The type "ENUM" may only be provided with one single element in the value range. Use this parameter definition to provide multiple input masks for one cycle. By specifying only one single element in the value range, the input mask is assigned to this parameter value. In this case, the corresponding parameter is not displayed in the input table.

Example:

Parameters of type "ENUM"

A position parameter (Inline CPL) has the character string type, wherein only two values, "CCW" and "CW" are accepted.

```
//%P% ENUM["CCW" counterclockwise, "CW" clockwise] Direction of rotation
```

A parameter without an address outside a parameter list is to be used to program spindle control commands M3, M4, M5 and M19.

```
//%P% ENUM[M3 right, M4 left, M5 stop, M19 orient] Spindle command
```

A position parameter has to apply the values 0, 1, 10 and 11 (similar to the binary representation of a number). It is not possible to provide the meaning as short plain text. Thus, the individual values are illustrated in a graphic. The element value is therefore repeated in plain text.

```
//%P% ENUM [0 0, 1 1, 10 10, 11 11] variant
```

Weekday codes (two characters) are to be assigned to a parameter with address D. The parameter is to be multilingual.

```
//%P% %%D [Mo %TN0501%, Tu %TN0502%, We %TN0503%, Th %TN0504%, Fr %TN0505%, Sa %TN0506%, Su %TN0507%] %TN0500%
```

Character string STRING

Character strings are declared using the type specification "STRING".

The range specification <range> is always omitted in case of character strings.

Example:

Parameters of type "STRING"

A position parameter has to apply a character string.

```
//%P% STRING engraving text
```

Default values and pre-assignment

Optional pre-assignments can be implemented for each parameter when inserting a call or a default parameter behavior. The syntax elements that follow "%D%" <behavior> and <default> are therefore provided.

Pre-assignment

Due to the pre-assignment, an initialization value is entered when a call is inserted and before the mask dialog becomes visible. Pre-assignment is not important for corrections.

During pre-assignment, the only value programmed at the position of <default> is the initialization value (rather than the substitute representation in plain text for the types and BOOL!). Only the first word after %D% is accepted as initialization value. Enclose the character string in inverted commas.

Use the pre-assignment option if the same value is used frequently, particularly if this is an obligatory parameter.

Example:

Parameter pre-assignment

The direction of rotation programmed in a spindle control command is almost always clockwise (to the right).

```
//%P% [M3 right, M4 left, M5 stop, M19 orient] Spindle  
command %D%M3
```

Default parameters

A "default parameter" is an optional cycle parameter, for which - if it is not programmed - a specific value from the value range of the parameter is used in the program (e.g., 60° thread edge angle if the parameter is not assigned). This differs from a similar case, in which an unassigned parameter leads to the omission of a certain function or process (e.g., no chip breaking). The latter is not a default parameter, since the default is "unassigned" and applies to each parameter in this case.

For default parameters, "unassigned" stands for a value. This value can constantly be changes (depending on options or via the context of the subroutine call depending on the previously activated interpolation plane).



Default parameters that depend on the call context cannot be implemented into the input masks. Such a condition can be the interpolation plane that is activated before the cycle for example. Document the behavior in an explanatory text (see [Global Mask Elements - Explanatory Texts, page 108](#)).

Constant default parameters

If the default value is constant, implement the default parameter behavior into the input mask. Enter "=" instead of <behavior> is followed by the default value (<default>).

With an unassigned default parameter, the default value always is displayed in the input mask when inserting and correcting. When transferring a call to the NC program, the respective parameter is omitted if its value corresponds to the default value.

*Example:***Default parameters**

In a thread turning cycle, the edge angle is to be transferred via a parameter. The default value of this parameter is 60°.

```
//%P% %%FA REAL[0..179.9999] %TN7321% %D%=60
```

Option-dependent default parameters (GNP)

There is an extension compared to the aforementioned default parameters for input masks of the graphic NC programming. The default parameter is not a constant, but it is taken from a current "Options" setting (see "MTX Workshop programming turning and milling", chapter "GNP options"). This "Options" setting is accessed via placeholders resulting from the address (enclosed in percentage symbols) of a value from the file the in project folder <VisualizationDevice>\user\ config\MTXGnpConfig_000_<channel>.ini, section "[Tech_Settings]". The file structure is documented in the manual "MTX Workshop Programming Turning and Milling", chapter "Options and user settings".

*Example:***Option-dependent default parameter**

An address parameter AT is to be provided with the default value set in the GNP options for the tangential approach length when turning.

```
//%P% %%AT REAL[0..9999] %TN7327% %D%=%Siabst_Tan%
```

Channel 1 is assigned to the program. The file MTXGnpConfig_000_001.ini contains:

```
[Tech_Settings]
..
Siabst_Tan = 2,
```

The default value of the parameter is 2.

Option-dependent default parameters with output condition (GNP)

Option-dependent default parameters leads to the risk that another program sequence results after modifying the options or after transferring a program to another machine. Thus, a parameter with the default value set in "Options" should not be transferred to the source code. Whether the default value is transferred can also be set using the "GNP Options" (see "MTX Workshop Programming Turning and Milling", section "Tabs - General" in the chapter "GNP options"). The following settings in file <VisualizationDevice>\user\config\ MTXGnpConfig_000_000.ini, Section "[View]", control the output (see also "MTX Workshop Programming Turning and Milling", chapter "Options and user settings"):

- "ForceAT"
Responsible for the approach when turning
- "ForceCV"
Responsible for the cutting variant when turning
- "ForceTR"
Responsible for the edge radius when turning
- "ForceTRV"
Responsible for the tool retract motion at the starting point before the cycle

If the value stored there is "1", each value is transferred to the NC program. If the value is "0", normal behavior applies, as it is the case with an option-de-

pendent default parameter. This means that the parameter is only displayed if it is different from the default parameter.

To formulate the output condition with an option-dependent default parameter, enter "`!<placeholder_outputcondition>`" for `<behavior>` and the placeholder "`<placeholder_tech_setting>`" for default. Enclose each placeholder with percentage signs.

Example:

Option-dependent default parameter with output condition

An address parameter TRV is to control the retract motion to the starting point of a cycle. The default parameter is to be taken from the "GNP Options", value address "Tool_Rev_T_Out". If the parameter value corresponds to the default value, it is to be transferred to the NC program depending on the options (output condition "ForceTRV").

```
//%P% %%TRV [0 without, 1 inclined, 2 X-Z, 3 Z-X] Re-
tract motion %D%!%ForceTRV%%Tool_Rev_T_Out%
```

Only if found in the file MTXGnpConfig_000_000.ini

```
[View]
```

```
..
```

```
ForceTRV = 1
```

the parameter is unconditionally applied to the NC program.

Parameter-related graphics

Using an addition introduced by the "`%GF%`", an own graphic can be assigned to every parameter in an input mask, which temporarily hides the global graphic ([Global Mask Elements - Graphics, page 108](#)). A parameter-related graphic is recommended if the effect of a parameter cannot be clearly represented in one or more global graphics.

<graphicfile> Name of a graphic file without path including the extension ".bmp", ".jpg", ".gif" or ".tif". Size of the images 223 x 262 pixels (width x height). Enclosed file names with spaces with inverted commas.

<graphictitle> As an input mask is automatically provided with multiple graphics if a graphic is parameter-related, assign a title to each graphic. These titles are displayed in the selection list above the graphic. The title includes the entire text after `<graphicfile>` to the end of the line or to the next reserved word enclosed by a percentage sign. To support multilingualism, a token can also be specified ("`%TNxxxx%`", see [File structure - Support of multiple national and regional languages, page 102](#)).



Behavior towards parameter-related graphics changed. It is now incompatible. For MTX 10VRS and higher, there is an overlapping of the global graphic by the parameter-related graphics. Up to now, it was only necessary to activate the global graphic again in the following parameter using `%GF%`.

Adapt your input masks from older versions accordingly.

Example:

Parameter-related graphics

A cycle consists of two global graphics and one parameter-related graphic (to the second parameter).

```
//%P% Parameter 1
```

```
//%P% Parameter 2 %GF%GraphPar3.gif Effect of parameter 2
//%P% Parameter 3
//%GF% GraphGlob1.gif first variant
//%GF% GraphGlob2.gif second variant
```

Validation information <validinfo>

The validation of parameter values is defined irrespective of other parameters by specifying a parameter type (see [Standard parameter types, page 109](#)). This section shows how to provide further information (initiated via "%VALID %") which describes dependencies between individual parameters and how to declare a parameter to an obligatory parameter.

The validation information can trigger system-internal tests and lead to desirable, error-avoiding input obligations. Other relations are simply visualized graphically to illustrate a programming rule about one or more parameters to the programmer at a glance.



Only simple standard cases can be dealt with using the validation information. More complex relations between individual parameters have to be presented in an explanatory text.

Obligatory parameters

The declaration as obligatory parameter initially results in the parameter being labeled in the input table using the symbol . There is no test when transferring a call whether an obligatory parameter is actually filled out.

From a functional point of view and in case there is an unassigned obligatory parameter occurs in an output line, this line is inserted, even if no single parameter in this line is assigned. Without obligatory parameter labeling, a line without an assigned parameter is omitted. If this is an obligatory parameter in a parameter list, the call name and the brackets around the parameter list are also displayed. With position parameters, empty parameters are optionally filled so that the parameter list has the corresponding length for the subsequent acceptance of all mandatory parameters.

Instead of <validinfo>, "MAND" is programmed for a mandatory parameter.

Example:

Validation information of mandatory parameters

The parameter DT is a mandatory parameter.

```
//%P% %@%DT %TN2266% %VALID% MAND
```

Parameter pair

There are optional parameters which can only be programmed with one other or several other parameters. Parameters to be programmed in pairs can be identified by a symbolic string in the input mask connecting the corresponding parameters to each other. It is thus required to define all parameters of a pair amongst each other.

Instead of <validinfo>, "PAIR" is written into a parameter pair to which a list of placeholders is connected indicating on other parameters of the pair. These placeholders look as follows: %P<pnr>%. <pnr> is the continuous parameter number from top to bottom beginning with 1. When defining a pair, it is freely selectable on which parameter of the pair the references to the parameters belonging to the pair are entered.

Example:

Validation information of parameter pairs

A cycle has six parameters. The parameters 2 to 5 are to be optional and are to transfer the four corner coordinates of a machining window.

```
//%P% %%ID INT[1..999] %TN0711%
//%P% %%WHT %TN0712% %VALID% PAIR%P3%%P5%
//%P% %%WHD %TN0713%
//%P% %%WVL %TN0714%
//%P% %%WVR %TN0715% %VALID% PAIR%P4%
//%P% %%F REAL[0.001..99999] %TN0716%
```

In the example above, the PAIR entries are integrated in different parameters on purpose to show multiple options for a pair definition.

Param.	Comment	Value
ID	Ident of contour	1
WHT	Machining window outer border	50
WHD	Machining window inner border	20
WVL	Machining window left border	-60
WVR	Machining window right border	-25
F	Feed	

Fig. 6-6: Exemplary layout of the input table for the parameter pair

Alternative parameters

Many parameters exclude each other and cannot be programmed at the same time. This situation occurs occasionally with optional parameters, but is also possible with obligatory parameters. The only case supported within the input masks is the one that ensures that - at the most - only one parameter from an arbitrary number of parameters can be assigned. Other conditions, for example, two parameters out of three, are not implemented. The input table prevents the assignment of a second parameter from the group of alternative parameters by disabling all other input fields if a parameter is provided with a value. While correcting, a multiple assignment is detected and a corresponding error message is displayed. No additional symbols are displayed.

Instead of <validinfo>, "ALT" is written for alternatively used parameters. A list of placeholders follows indicating the alternative parameters. These placeholders look as follows: %P<pnr>%. <pnr> is the continuous parameter number from top to bottom beginning with 1. Also when defining alternative parameters, the parameter of the alternative group at which the references to the other related parameters are entered is irrelevant.

Example:

Alternative parameters

There is a turning cycle for which chip breaking can optionally be programmed. If chip breaking is to be activated, specify the distance CBD after which the chip is to be broken. In addition, program either a dwell time DWT or the number of dwell revolutions (combination of alternatively programmed parameters or parameters programmed in pairs).

```
//%P% %%ZS %TN0801%
//%P% %%ZE %TN0802%
```



```
//%P% %@%CBD REAL[0.0001..9999] %TN0803% %VALID% PAIR%P4%
%P5%
//%P% %@%DWT REAL[0.0001..9999] %TN0804% %VALID%ALT%P5%
//%P% %@%DWR INT[1..9999] %TN0805%
```

Param.	Comment	Value
ZS	Start coordinate	-20
ZE	End coordinate	-30
CBD	Distance per chip break	8
DWT	Dwell time during chip break	
DWR	Dwell revolutions during chip break	2

Fig. 6-7: Layout of the input table of alternative parameters for example

Syntax of Mask Definition - Special Parameter Types

Description coordinate system "DCS"

For the graphic NC programming, the parameter type "description coordinate system" (DCS) was introduced. For more detailed information on the description coordinate system, refer to the "MTX Programming Manual", chapter "Geometry Definitions".

This parameter type does not have an address. It permits one of the planes declared in the GNP options to be selected for further programming or a new plane to be specified in DIN syntax. The DCS type implements the necessary default parameters and validation behavior. If this parameter type is used, the variable parameter names can be used in this mask for coordinate axes (%A1%, %A2% and %A3%, (see [chapter File structure - Multiple National and Regional Languages, page 102](#)) which depend on the DCS selected. The parameter type DCS cannot yet be used for the programming of an unwinding.

Syntax	<code>//%DCS% <ptitle> {%GF% <graphicfile>{ <graphictitle>}}</code> with
<ptitle>	Assign a parameter title to the DCS. Specify the title as token ("%TNxxxx%") or as monolingual text. A monolingual title text extends to the end of the line or to the next percentage sign outside a bracket.
<graphicfile>	Name of a graphic file without path including the extension ".bmp", ".jpg", ".gif" or ".tif". Size of the images 223 x 262 pixels (width x height). Enclose file names with spaces in inverted commas.
<graphictitle>	As several graphics are "automatically possible" with a parameter-related graphic, assign a title to each graphic. This title is displayed in the selection list above the graphic. The title includes the entire text after <graphicfile> to the end of the line or to the next reserved word enclosed by a percentage sign. To support multilingualism, a token can also be specified ("%TNxxxx%", see Chapter File Structure - Support of Multiple National and Regional Languages, page 102).

Example:

Using the DCS parameter type

An input mask is to be defined for the GNP geometry definition "Point Pattern on Pitch Circle" (PCI).

```
//%P% %@%ID INT[1..999] GeometryIdent %VALID% MAND
//%DCS% Description coordinate system
```

```
//%P% %%A1% Center point of main axis %VALID% MAND
//%P% %%A2% Center point of secondary axis %VALID% MAND
//%P% %%R REAL[0.0001..9999] Radius %VALID% MAND
//%P% %%NR INT[1..999] Number of points %VALID% MAND
//%P% %%SA Angle of first point
//%P% %%EA Angle of last point %VALID% ALT%P9%
//%P% %%A Angular pitch
```

Parameters with variable binding

In MTX 08VRS and lower, a special parameter type was supported. For that type, only CPL variables could be specified instead of a constant value. This data type whose definition started with `///%VAR%` cannot be used any longer.

Replace this type in older input masks by a default type. To assign a value via a CPL variable or a CPL expression, enclose the expression/variable with square brackets in the input field as for an Inline CPL.

Syntax of mask definition - Output format

Using the "output format", the part of the syntax of the call belonging to the input mask is defined. It includes more than the collection of the individual parameters. The output format includes the compilation of the parameters and their combination with other unchangeable block components and comments.

Objective The output format defines the syntax of a call, which, during correction, also permits the unique input mask identification which suits a piece of program text best. The format forms the link between the NC program and the input mask.

An output format can contain any number of characters.

Syntax `///%N% {<format>} {<comment>}`

with

<format> The format of the call(s) is stored under `<format>` in the form in which it is to be displayed in the NC program. Enter the call names and the brackets enclosing the parameter lists. Unchangeable block components are displayed in their final form. Placeholders are provided for the parameters:

`"%P%"` stands for all parameters appearing in the order of their definition (see also [chapter Syntax of mask definition - Standard parameter types, page 109](#)).

If the parameters are distributed across several parameter lists or if position parameters require a different order than given in the parameter definition, control each parameter individually. Placeholders looking as follows `"%P<pnr>%"` are again used. `<pnr>` is the continuous parameter number from top to bottom beginning with 1.

No comma has to be inserted between the placeholders in a parameter list. The commas are automatically inserted. The insertion of spaces cannot be controlled by the format specification.



For simple and therefore single line cycle calls (see [chapter Advanced systematization of calls, page 99](#)), the specification of the brackets enclosing the parameter list and the placeholders can be omitted in this exceptional case. Only the call name has to be specified. For parameters without address, a parameter list in square brackets and for parameters with address, in round brackets follows automatically.

<comment>

The comments initiated with a semicolon can either directly be specified or as a separate line after the call. In monolingual texts, the semicolon is written first and then the comment. If multiple languages are to be supported, a semicolon is not to be specified, but only a token as follows `%TNxxxx%` (see [chapter File structure - Support of multiple national and regional languages, page 102](#)). Comments in round brackets are not supported except for NC notes.



- If comment lines are to be added to multiline calls (via the output format or by subsequent editing), note that the number of comment lines between two DIN lines is limited to 4. Once this limit is exceeded, not all lines belonging to the format can be found during the correction. This can result in missing parameter values in the input table and double assignments when transferred to the program.
- Comments supported by multilingualism are only generated during insertion. During correction, comments remain untouched. There is also no translation into another language if a language setting - differing from the language setting during the insertion - is active when correcting the call.

Example:

Output formats

Program a cycle with position parameters. A monolingual comment is to be added to the call.

```
//%N% G84[%P%] ;Thread drilling
```

Program a cycle with address parameters. Add a multilingual comment to the call.

```
//%N% G722(%P%) %TN0722%
```

Write a functional comment to describe a tool for the simulation. The definition sequence of the parameters is to be different to the sequence in the parameter list. At the end of the line, add an unchangeable block component as Inline CPL.

```
//%N% //TOOL/STANDARD,%P2%%P3%%P4%%P5%%P1%\%  
[""+NCF("G90")]
```

A complete tool change (travel to the tool change point, tool change, a comment, an NC note and a master block) is to be provided for a lathe. The machine is provided with two spindles.

```
//%CHBEGIN%
```

```
//%NAME% Tool change
```

```
//%P% %%X Tool change point X (diameter) %D%200
```

```
//%P% %%Z Tool change point Z %D%250
```

```

//%P% ENUM[M105 Spindle 1, M205 Spindle 2] Spindle stop
before tool change
//%P% INT[1..12] T-number %VALID% MAND
//%P% %%MSG STRING NC note
//%P% ENUM[DIA Diameter, RAD Radius] X-scaling %D%DIA
//%P% ENUM[G17 G17, G18 G18, G19 G19] Interpolation plane
%D%G18
//%P% %%G REAL[54..59.6] Zero point offset %D%54.1
//%P% ENUM[G97 Speed, G96 Cutting velocity] Spindle pro-
gramming constant ... %D%G97
//%P% %%S1 REAL[0.001..99999] 1. Spindle value %VALID%
PAIR%P11%
//%P% ENUM[M103 right, M113 right KM, M104 left, M114
left KM] First spindle command %D%M114
//%P% %%S2 REAL[0.001..99999] Second spindle value %VAL-
ID% PAIR%P13%
//%P% ENUM[M203 right, M213 right KM, M204 left, M214
left KM] Second spindle command
//%N% G0 DIA G53 G90 G48 %P1% %P2% %P3%
//%N% ;----- Tool change -----
//%N% T[%P4%] M6
//%N% (%P5%)
//%N% %P6%%P7%G47%P8%%P9%%P10%%P11%%P12%%P13%
//%CHEND%

```

Configuring the NC Program Editor/Text Editors

Overview on the configurations steps

Editor types *To edit text files and NC programs, there are two editor types in IndraWorks Operation:*

- NC program editor
- Text editor

The NC program editor provides additional functions for the NC programming while the text editor is more simple and also suitable for bigger files.



The following steps can be made either in the NC program editor or in the text editor. Thus, it is simply called "editor" in the following description.

Configuration steps The same options to adjust and change are provided for both editor types. During installation, each editor is provided with all functionalities and a basic layout. Thus, no configuration steps are required when working with each editor. Special user or machine manufacturer requirements can be configured. The following configurations are distinguished:

Options The editor behavior and its layout can be set under **Tools ► Options ► Editor Options...**:

- Criteria when to use the editor types
- Tab positions
- Font

- Behavior when assigning block numbers

The options of the editor are modified in IndraWorks Operation. The settings are assigned to the Windows user currently logged in and may only be changed in exceptional cases during commissioning. Thus, these settings are not described in detail in the commissioning manual. Instead, it is referenced to the chapter "Options - Options Editor..." in the manual "MTX Standard NC Operation".

Data for input support

Both editors are provided with identical input assistances. These are integrated tools that insert individual words or complete sections and partially also correct them. This procedure is dialog-guided.

Data can be provided for the following parts of the input assistance:

- The **NC block sequences** are programming templates
- The **input masks** are input tables to edit parameter values to instructions

The input masks of the user and machine manufacturer, the information on hidden input masks as well as the NC block sequences are part of the IndraWorks project.

F-keys of the input support

The function keys to insert commands/calls via input mask can be changed in their allocation. These settings are made in IndraWorks Engineering.

Two configurable F-panels are available. The following section describes the particularities when adjusting these panels.

The F-keys of the editor are part of the IndraWorks project.

F-keys for input assistance

General information on F-panels

F-panels

In total, there are four F-panels to insert instructions via input masks:

1. For the configurable **"left part of the menu"**, also refer to legend (5) in the following figure. This panel named "InsertCycle1", also called left half of the menu, is the entry level when inserting cycle calls and is referred to as **"InsertCycle1"**. On this panel, group filters and different characteristics can be combined for each F-key. When pressing this key, the provided mask scope is reduced to those provided with at least one of the characteristics listed in the filter. Switching between different group filters is illustrated by the selection buttons for these keys. If the number of requested different filters does not fit on one panel, the key for switching to a subsequent level is provided
2. The subsequent level is called **"right half of the menu"** and named **"InsertCycle2"**. This panel can also be configured and has to be provided with a key for level switching back to the left menu level
3. The non-configurable panel with **instructions for simulation**
4. The non-configurable panel for **DIN instructions and contour drafts**

HMI commissioning



- (1) Main level
- (2) "Input support" panel
- (3) "Instruction for Simulation" panel
- (4) "DIN Code Contour Drafts" panel
- (5) "Left half of the menu" panel (name "InsertCycle1")
- (6) "Right half of the menu" panel (name "InsertCycle2")

Fig. 6-8: Overview on F-panels for insertion via input masks (default assignment shown)

F-keys Five special functions are allowed on the configurable F-panels "InsertCycle1" and "InsertCycle2":

- **"Set cycle group editor"** to define a group filter
- **"Level switching editor"** to switch between the left and the right half of the menu
- **"Channel selection editor"** for additional limitation of the menu to one channel
- **"Cancel entries editor"** to discard entries
- **"Apply parameters editor"** to add the call to the NC program

F-panel editor The panels are adjusted in the F-panel editor in IndraWorks Engineering.



The F-panels "InsertCycle1" and "InsertCycle2" may not be deleted.

When modifying the F-panel, the following rules are to be observed (IndraWorks Engineering does not verify whether they are complied with) apart from the exclusive use of the five named functions:

Function "Set Cycle Group Editor"

"Set cycle group editor"

The "Set Cycle Group Editor" function allows the key to enable a specified group filter. The "Group" combo box lists all predefined group characteristics. Select a group or enter multiple group names separated by spaces. Individual group names can also be used.

The chapter [Syntax of Mask Definition - Properties of an Input Mask \(Grouping\)](#), page 106 describes how the input masks are assigned to the individual groups and the significance of the predefined groups.

Rules

The following rules and particularities apply:

- F-keys with this function are automatically subject to a selection functionality
- Each key can be assigned to one graphic. Its position (preferred orientation on the bottom right, X=5 and Y=3) is used to set the selection button graphic
- F-keys, whose group filters specify an empty set, are hidden and do not have to be deleted
- Two group filters without an empty set are to be specified at least on each panel. Alternatively, all group filters may specify an empty set on the panel "InsertCycle2" or all keys with the "Set Cycle Group" function may be deleted. Both conditions cause that the "InsertCycle2" panel is not displayed in the dialog.
- A group name can be used an unlimited number of times on different F-keys.
- Always intend one F-key with the group filter "_OTHERGROUPS". This special filter is for groups that are not selected on other F-keys by a filter. If such unassigned input masks exist, these are hidden and cannot be inserted.
- Assign F-keys with group filters consecutively.

HMI commissioning

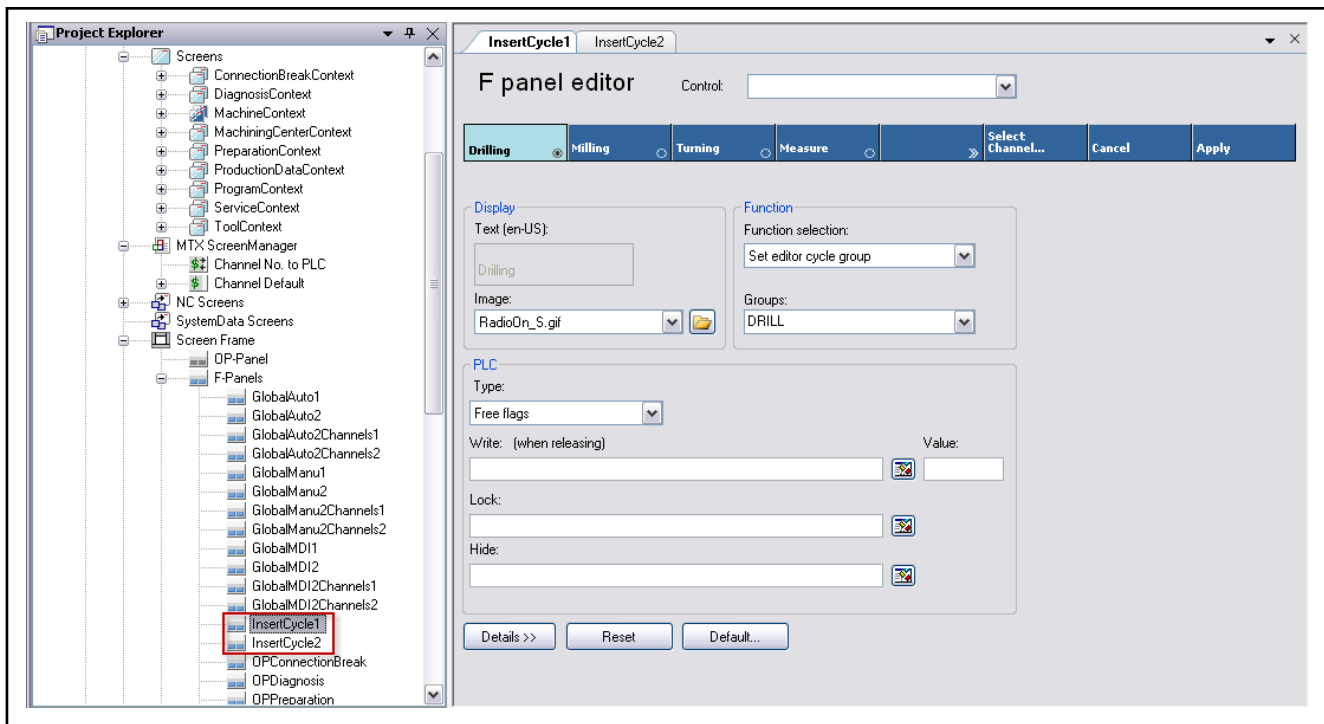


Fig. 6-9: Configuring F-keys of the input support "Set Cycle Group Editor"

Function "Level switching editor"

"Level switching editor"

The "Level Switching" function switches between the left and the right half of the menu. Enter the next F-panel name under "Level Name".

Rules

The following rules and particularities apply:

- The panel name always has to be the counterpart of "InsertCycle1" and "InsertCycle2" of the currently edited panel.
- If all group filters of the "InsertCycle2" label empty sets, the key to switch to the level "InsertCycle2" is hidden. In this case, it may be deleted or overwritten with the "Set Group Editor" function
- The "Level Switching Editor" function may be allocated at any position on the panel

Function "Channel selection editor"

"Channel selection editor"

The "Channel Selection" function enables a dialog in which a channel or the setting "channel-comprehensive" can be selected.

Rules

The following rules and particularities apply:

- If there is only one channel or the channel-related hiding of input masks is not used, the key with the "Channel Selection" function is hidden. In this case, it may be deleted or overwritten with the "Set Group Editor" function
- The "Channel Selection Editor" function may be allocated at any position on the panel. It is recommended to keep <F7>, since the channel selection is located on other non-configurable editor panels on <F7>

Function "Cancel entries NC editor"

"Cancel entries editor"

The "Cancel Entries" function quits the input mask and returns to the editor view.

Rules *The following rules and particularities apply:*

- This function has to be present on each panel.
- The function can be allocated at any position on the panel. It is recommended to keep <F8>, since "Cancel Entries NC Editor" is located on other non-configurable editor panels on <F8>

Function "Apply parameters NC editor"

"Apply parameters editor"

This function applies the edited instruction to the editor and returns to the editor view.

Rules *The following rules and particularities apply:*

- This function has to be present on each panel.
- The function can be allocated at any position on the panel. It is recommended to keep <F9>, since "Apply Parameters Editor" is located on other non-configurable editor panels on <F9>.

Handling instruction: Configuring the editor - Input assistance

Creating basic conditions



Configuring the input assistance of the editor is optional.

Basic conditions

Ensure or create the following basic conditions:

1. An IndraWorks project is created
2. A visualization device is created in the project
3. Visualization data is transmitted and activated

Applying input masks by importing cycles

Applying input masks by importing cycles



The cycle import comprises the input masks as well as the cycle subroutines stored in the NC file system. Thus, there is a partial overlapping with the chapter [173 Setting Up NC Cycles, page 173](#).

The configuration is made in IndraWorks Operation.

If input masks are present from another project, proceed as follows:

1. Switch to the "Program" operating area
2. Select an existing file with the extension ".npg" in the file list of the navigator Alternatively, a new NC program can also be created instead. Select a directory in the directory tree, press <F2 New> → <F3 NC program>, enter any short text and complete the editor with <F9>. In this case, also select the new NC program.
3. <F8 NC prog functions> → <F5 Import cycles ...>.
4. Select the import file (extension ".ncc") and press **Open**
5. Select the cycle package (origin) to be imported:
 - USER cycles are input masks and cycle files of the user
 - OEM cycles are input masks and cycle files of the machine manufacturer

Press **OK**.

Creating input masks

To create an input mask, external programs of the Windows operating system or similar programs such as the file explorer, text editor and graphic editor are used. The input masks are tested in IndraWorks Operation.

Examples as templates For MTX 12VRS or higher, examples of input masks are provided when creating a new project. These input masks are suitable for familiarizing with the mask definition technique. This is the most effective procedure to define individual masks from already existing ones.

Activating examples The familiarization with the examples shows the possibilities of input masks and helps to specify the requirements on the input masks.

The example input masks are normally hidden and can be shown as follows:

1. Switch to the "Program" operating area.
2. Open the editor To do so, select an existing file in the file list of the navigator and open it with <F6 Edit> or via the context menu. Alternatively, a new file can also be created instead. Select a directory in the directory tree and press <F2 New> → <F3 NC program> or <F4 Text file>. The context menu can also be used instead
3. <F8 Tools> → <F8 Options> → <F4 Adjust input support...>.
4. Select the "Examples" node on the "Availability" tab and press **OK**
5. <F9 Back>

Work with input masks as follows:

1. <F5 Input support>
2. <F2 NC cycle> → <F6 >>> → <F5 More cycles> (This key sequence is the standard key assignment for input help)
3. Select an example (SAMPLE1 ... SAMPLE4) and enter the parameter values. Use the online help as well with <F1> to find explanations on the examples.
4. Apply with <F9> and continue with the next example starting at point 2.

Testing examples Test the example input masks by editing all parameters and inserting calls into the NC program.

The following examples with increasing complexity are provided:

- Example 1: Simple example with position parameters without foreign language support
- Example 2: Simple example with pre-assigned address parameters, two global images and foreign language support
- Example 3: Advanced example with mandatory, standard and alternative programmed parameters or parameters programmed in pairs. Individual image per parameter
- Example 4: Multi-line example of DIN programming

Defining individual input mask *Define the first input mask as follows:*

1. Create a copy of the file C:\Documents and Settings\All Users\Application data\Rexroth\IndraWorks\Project\<VisualizationDevice>\user\config\cycles\SampleMasks.cyc in the same directory. The copy can have any name, but the extension ".cyc" has to be kept.
2. Open the .cyc file with a text editor. Select the most suitable example mask.
3. Edit the individual elements of the mask definition one after the other. First, do not use any tokens for multilingualism, but texts in your language. The line "//%LANG%" has to be converted to a comment using a semicolon in front.
4. Creates the images in a size of 223 x 263 pixels in the same directory using an external graphic editor and enters the name to the .cyc file (behind %GF%).

5. Assign the mask to one or several groups (//%GROUP%).
6. Delete the masks that are not required and save the .cyc file.
7. Test your input masks in the editor of IndraWorks Operation and correct them if required. The mask definition saved in the editor is always updated after closing all editors.
8. To support several languages, copy SampleMasks_EN.txt and create one text file each for your native language and at least one more for English. Fill out the token and write the token number %TNxxxx% to the .cyc file. Enable the //%LANG% instruction (remove preceding semicolon) and replace the name by your text file (without country code and extension). Test the multilingualism of the input mask in the editor. Note that the multilingual support of the modified texts is only displayed after an IndraWorks Operation restart.
9. If there are online helps, copy them to the subfolder Help\<countrycode> and adjust them to the instructions //%HELPPFILE% and //%HELPTOKEN%. Otherwise, the mentioned lines have to be deleted or converted to a comment using a semicolon.
10. One user mask is now defined. Any further number of input masks can be added to this or another .cyc file. As machine manufacturer, move all files belonging to the just defined masks to the folder C:\Documents and Settings\All Users\Application Data\Rexroth\IndraWorks\Project\<VisualizationDevice>\OEM\config\cycles.

Hiding input masks

Machine manufacturers often have to remove input masks of some standard cycles, since the machine is not suitable for certain technologies or there are no axes for certain motions. In principal, input masks can be hidden in the options of the editor (IndraWorks Operation). The option is available for machine manufacturers and users if the respective user permission is granted.

To remove certain input masks from this adjustment dialog, proceed as follows:

1. Create a copy of the file C:\Documents and Settings\All Users\Application Data\Rexroth\IndraWorks\Project\<VisualizationDevice>\user\config\cycles\SampleMasks.cyc in the directory C:\Documents and Settings\All Users\Application Data\Rexroth\IndraWorks\Project\<VisualizationDevice>\OEM\config\cycles\. The copy can have any name, but the extension ".cyc" has to be kept.
2. Open the .cyc file with a text editor. Delete all example masks except the last one (SAMPLE 5).
3. Open the file IndraWorks\config\cycles\MTX.Cycles.Canned.cyc with an editor in the installation directory.
4. Copy all lines of the input mask to be hidden beginning with //%N% to the clipboard.
5. Replace the existing //%N% line with the content of the clipboard.
6. If the input mask to be suppressed is located in the line //%NAME% of an instruction %SIGN%, also transmit this line to the new .cyc file.
7. Any further number of input masks to be suppressed can be added to this or another .cyc file.

Example:

Example of hiding an input mask with additional signature

This is the definition of the input mask that is to be hidden:

```
//%CHBEGIN%
//%LANG% MTXCyclesCanned
//%NAME% %TN801% %SIGN%_RptPnt
//%GROUP% DIN
//%GF% Contour_Chamfer_Round_Base.gif %TN809%
//%P% %%X REAL %TN802%
//%P% %%Y REAL %TN803%
//%P% %%Z REAL %TN804%
//%P% %%CHL REAL[0..9999] %TN805% %GF% Contour_Cham-
fer_Length.gif %TN809% %VALID%ALT%P5%P6%
//%P% %%CHS REAL[0..9999] %TN812% %GF% Contour_Cham-
fer.gif %TN809%
//%P% %%RND REAL[0..9999] %TN806% %GF% Contour_Round.gif
%TN809%
//%P% %%FL REAL[0.0001..99999] %TN807%
//%C% %TN811%
//%C% %TN1000%
//%C% %TN1001%
//%N% G0 (%P4%%P5%%P6%%P7%)%P1%%P2%%P3%
//%HELPPFILE% MTX_StandardNCCycles.chm
//%HELPTOKEN% ncc.ContourBezelCurve.htm
//%CHEND%
```

Apply the call signature from `//%N%` (parameters can be omitted) and set the group characteristic `"_HIDDEN"`. Since an additional signature was arranged with `%SIGN%`, add this together with the line `//%NAME%`.

```
//%CHBEGIN%
//%NAME% %TN801% %SIGN%_RptPnt
//%GROUP% _HIDDEN
//%N% G0
//%CHEND%
```

Loading data from operating station

To backup input masks or block sequences in the "IndraWorks Engineering" project, start IndraWorks Engineering, right click on the node of the visualization device and start the process under **Visualization Data ► Load changes from the operating station**.

F-keys of the input support

Menu design

At a new project, the IndraWorks Operation editor is provided with a menu structure to select groups of input masks that covers all existing and not permanently hidden input masks. There is no channel-specific restriction. If groups are irrelevant for the machine, it can be required to replace menu keys without function with menu keys with functions or to omit the "right half of the menu".

Preliminary considerations

Ask the following questions:

1. Which different group names are interesting for my project? Which groups should not be hidden?

Determine the group names known by installed and individual input masks as follows:

- Start the IndraWorks Operation desktop and switch to the "Program" operating area.
 - Open the editor To do so, select an existing file in the file list of the navigator and open it with <F6 Edit> or via the context menu. Alternatively, a new file can also be created instead. Select a directory in the directory tree and press <F2 New> → <F3 NC program> or <F4 Text file>. The context menu can also be used instead
 - <F8 Tools> → <F8 Options> → <F4 Adjust input support...>.
 - All groups are listed first on the "Availability" tab. The tooltips of the group nodes indicate the respective group names.
 - After pressing the key **F-Key View**, the current menu structure and the groups and input masks hidden below can be seen.
2. Is a channel-dependent reduction of the scope of the provided input masks required? "No" if there is only one existing channel.

Determining the menu design

Specify the menu structure for the two configurable panels "InsertCycle1" and "InsertCycle2". The two groups "DIN" (DIN programming, contour draft) and "SIM" (instructions for simulation) are located on panels that cannot be changed. These do not have to be considered.

The groups can be distributed on the F-keys as follows:

- No channel-related menu restriction is required.
 - If limited to one panel, six F-keys for group selection are available. Further keys are "Cancel" and "Apply".
 - In case of two panels, ten F-keys are available in total for group selection. Three keys each are reserved per panel for level switching, "Cancel" and "Apply".
- A channel-specific menu restriction is required. It applies that the scope of the provided masks can vary channel-specifically, but not the menu design.
 - If limited to one panel, five F-keys are available for group selection. The remaining three keys of the panel are required for channel selection, "Cancel" and "Apply".
 - In case of two panels, eight F-keys for group selection are available. The remaining four keys per panel are required for level switching, channel selection, "Cancel" and "Apply".

Think about the future menu structure without considering the groups to be hidden. Each function key can be assigned to several groups. Several F-keys can also be assigned to groups.



In the standard menu structure, the group filter "_OTHERGROUPS" is on the F5-key of the panel "InsertCycle2". This name stands for the remaining groups that are not selected as group filters by other F-keys. This filter should absolutely be assigned to a function key, since all new groups added at a later point in time appear under this filter. If no key is unassigned, assign this filter to a key already considered for another assignment.

Configuring function keys Transmitting and activating data

This configuration measure is listed in IndraWorks Engineering.

If IndraWorks Operation is still running, close this application now.

To activate the panels in the editor, right-click on the node of the visualization device and start the process under **Visualization Data ▶ Transmit and Activate**.

Configuring in the F-panel editor

The configurable panels of the input support in the editor can be adjusted as follows:

1. Select the node **Visualization device ▶ ScreenFrame ▶ F-panels ▶ InsertCycle1** in the Project Explorer. That is the panel of the left half of the menu. Open the node via the context menu or double-click.
2. First, edit the keys to set the group filters:
 - If a new labeling is required, delete the key first
 - If a key for level or channel selection should be used for group selection, delete this key before as well
 - Enter a new text if necessary
 - Select an image (e.g. RadioOff_S.gif) and correct its position if required. It has to be located on the bottom right, at the position X=5 and Y=3
 - Select the function "Set Cycle Group Editor"
 - Determine a group filter under "Groups". Several group names are separated by a space
 - Repeat step 2 for all other keys for group selection. Delete keys that are not required
3. If the key for level selection is required but different compared to the standard, delete the future key and provide it with the respective image (Next_Level_S.gif or Last_Level_S.gif). The function is set to "Level Switching Editor". Select the opposite of the currently edited panel under "Panel Name".
4. Delete the key for channel selection if still available and not required.



Keys without function are hidden. They do not have to be deleted.

5. To support multiple project languages, enable more languages under **Project ▶ Language ▶ Select Project Language...** and label the keys again.
6. Repeat the points 2 to 5 for the panel "InsertCycle2". If the right half of the menu is not required, delete all keys with the function "Set Cycle Group Editor" on this panel.

Hiding individual groups and input masks

The configuration step is made in IndraWorks Operation.

Hiding and sequence

Proceed as follows to hide specified input masks:

1. Switch to the "Program" operating area.
2. Open the editor To do so, select an existing file in the file list of the navigator and open it with <F6 Edit> or via the context menu. Alternatively, a new file can also be created instead. Select a directory in the directory tree and press <F2 New> → <F3 NC program> or <F4 Text file>. The context menu can also be used instead
3. <F8 Tools> → <F8 Options> → <F4 Adjust input support...>.
4. Deselect all nodes on groups and masks that are not required anymore. Go down to the channel node for a channel-specific menu restriction.
5. The key **F-Key View** can check the assignment to the keys

6. To adjust the sequence of the masks, enable the "Sequence" tab and move the masks accordingly.
7. Press **OK**.
8. Check the menu design via <F9 Return> → <F5 Input support> → <F2 NC cycle>.

Loading data from operating station

To backup menu adjustments in the project, start IndraWorks Engineering, right click on the node of the visualization device and start the process under **Visualization Data ► Load Changes from Operating Station**.

6.3.7 WinStudio

"WinStudio" is a high-performance collection of automation tools to create HMI applications.

WinStudio is described in detail in an individual manual.

6.3.8 Operating screens

Operating screens are used for the manual operation using M-keys. The current position of the moved axis as well as the states of freely configurable PLC variables can be displayed. The meaning of the individual states is determined when configuring the operating screen. It is additionally displayed whether an action can be executed.

Operating screens are described in detail in the "IndraWorks HMI" manual.

6.4 Screens



A general description to create and edit "screens" is described in the "IndraWorks HMI" manual.

Screen and operating areas can be created via the "Screens" node.

Each screen has to be assigned to one operating area.

A "screen" consists of a screen frame ("ScreenFrame") and a screen application ("ScreenApplication"). The screen frame consists of different panels and the machine status area ("header"). An application created with one of the tools below the "ScreenApplication" node is located in the center of the screen.

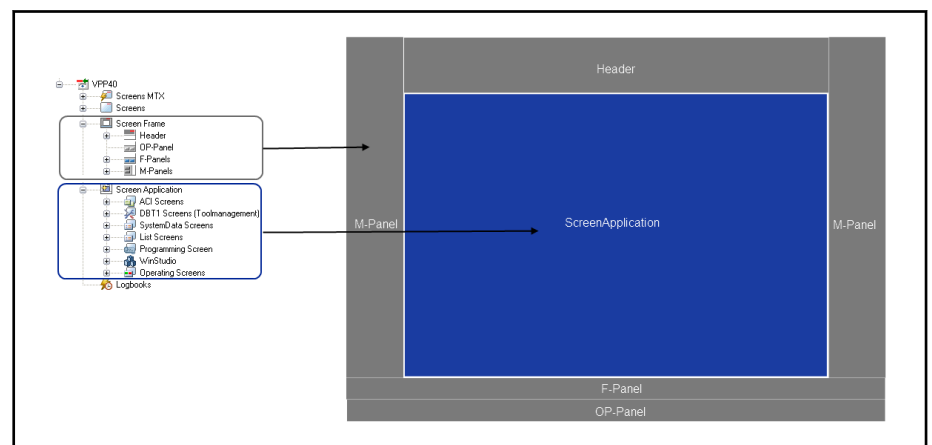


Fig. 6-10: Screen structure ("Screens")

Such a screen is composed below the "Screens" node using the "Screen - Configurator". This "Screen configurator" can be used to select the desired panel and application (out of the existing ones) and to position the screen accordingly.

"Screens" assigned to an operating area are displayed in the tree below it. The panels and applications assigned to the screen are also shown as link in the tree below the corresponding "Screens".

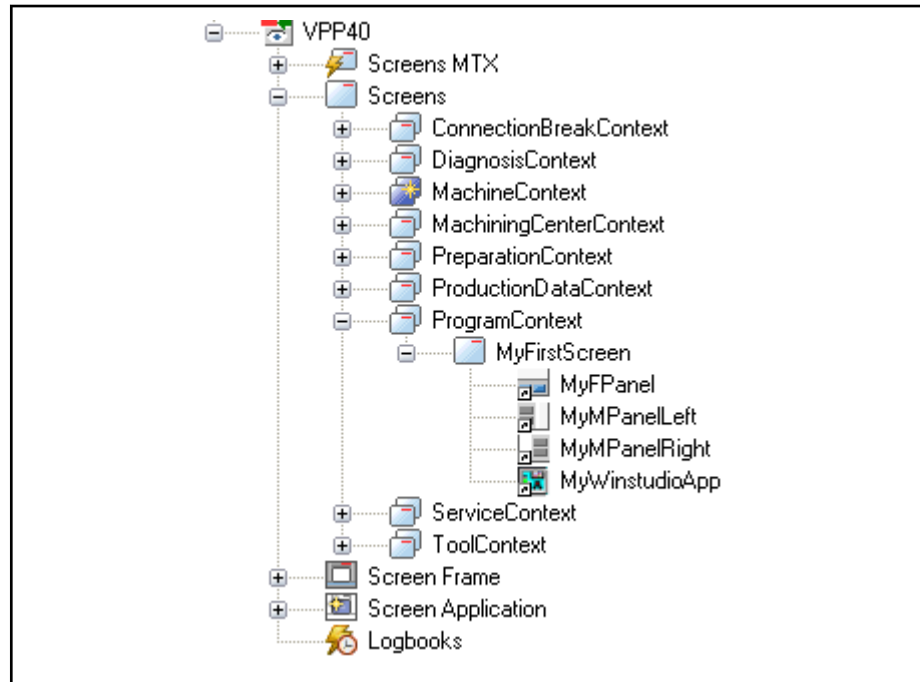


Fig. 6-11: Operating area (ProgramContext) with screen (MyFirstScreen) and assigned "panels" and application

6.5 MTX screens

6.5.1 General information

Events can be defined below the "Screens MTX" node. These events add screens defined here to IndraWorks Operation.

They are valid either for one channel or for all channels. Alternatively, the event can also be created so that it only applies if no individual events exist for one channel. For the definition, the validity range (that is the channel) is determined first. The corresponding event is then created below.

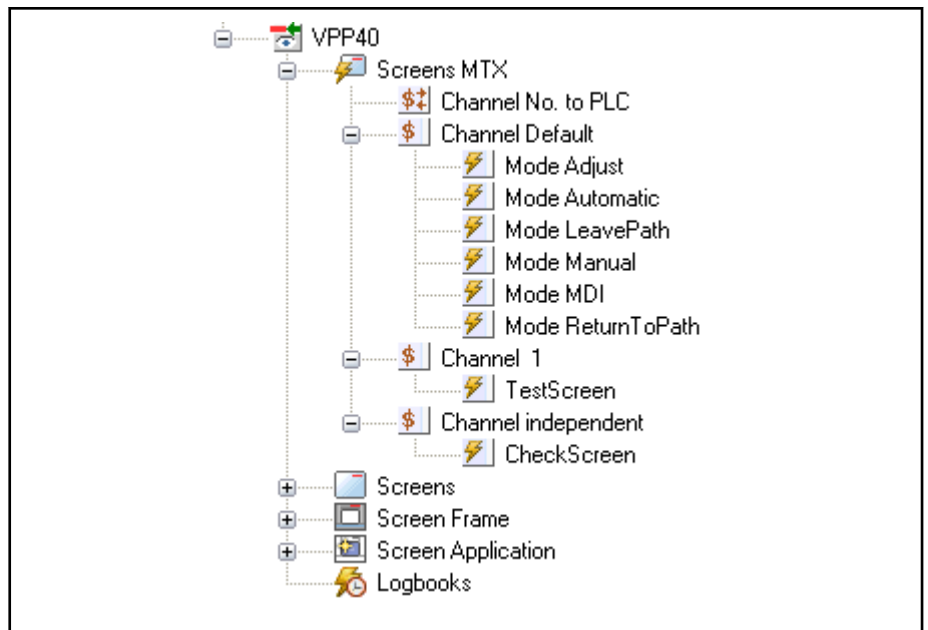


Fig. 6-12: Events

The event type can either be the state of a PLC variable (PLC) or of an SCP item (SCP). A specified operation mode of the control can additionally be selected as constant (CONST). Apart from "ACI Screens", simple "Screens" can also be assigned as screen type.

The "M-panels" assigned in "Screens" act locally. However, the "M-panels" assigned in "Screens MTX" act globally.

Local "M-panels" only remain active as long as the screen assigned to them is activated. Global "M-panels" always remain active. The only exception is if a screen with local "M-panels" is currently activated. If a screen without "M-panels" is activated, the latest activated global "M-panels" for this screen are displayed.

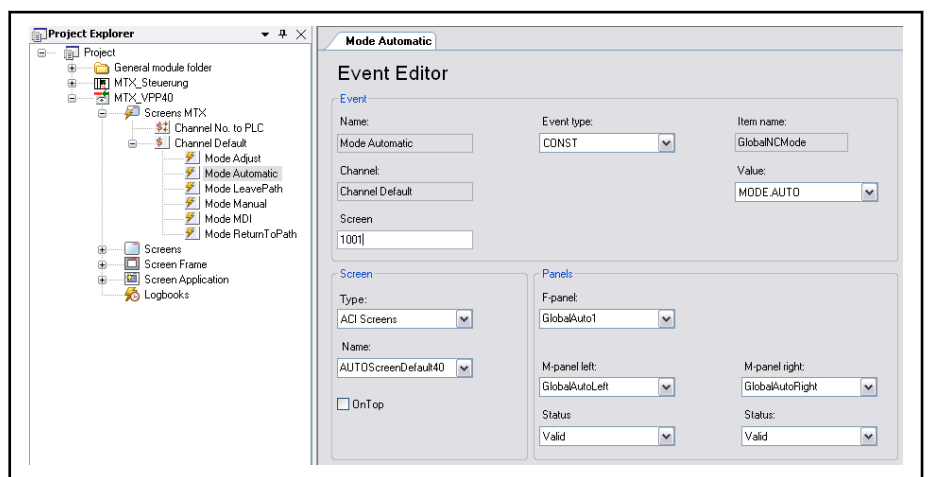


Fig. 6-13:

6.5.2 Validating MTX screens

For a general description on the validation of visualization data in the HMI, refer to the HMI manual "DOK-IWORKS-HMI*Vxx****-APxx-EN-P"

Checked criteria *When validating the "Screens MTX" visualization data, the following criteria are checked:*

- Uniqueness and validity of the event names
- Uniqueness and validity of the screen numbers
- Specification and validity of the referenced screen elements
- Validity of the references F-/M-panel
- Validity of PLC variables
- Specification of event type, item name and item value
- Uniqueness of PLC variables to handle the active channel number in the PLC

Overview on error numbers Each detected problem during the validation of the visualization data obtains an ID. Use this ID to find information on cause and remedy in the following table:

ID	Cause	Recovery
MTX-00010201	Event name is not unique	Rename the event
MTX-00010202	The event name contains at least one of the following characters: \ / : * ! ? " < >	Remove the special characters from the event name
MTX-00010210	PLC variables could not be checked	Assign an MTX control to the visualization device
MTX-00010211	Screen number is not unique	Open the event editor and enter a unique screen number
MTX-00010212	ASUP number invalid	Open the event editor and select an event type from the selection list
MTX-00010213	F-panel is invalid	Open the event editor and select an F-panel from the selection list
MTX-00010214	M-panel is invalid	Open the event editor and select an M-panel from the selection list
MTX-00010215	PLC variable is invalid	Include the variable in the symbol configuration or select a valid PLC variable
MTX-00010221	Screen name was not specified	Open the event editor and select an event type from the selection list
MTX-00010222	Event type was not specified	Open the event editor and select an event type from the selection list
MTX-00010223	Item name or PLC variable not specified	Open the event editor and select an item name or a PLC variable

MTX-00010224	Item value was not specified	Open the event editor and enter an item value
MTX-00010231	The PLC variables to handle the active channel number in the PLC are identical	Open the Properties dialog and enter different PLC variables for reading and writing

Tab. 6-47: Overview on the error IDs of the validation messages

6.5.3 Active channel number - Transferring to PLC/Changing via PLC

Only one channel is active at a time in "IndraWorks Operation". Thus, all displays refer to this channel.

To report the active channel to the PLC or to change it via the PLC, proceed as follows:

1. Declare an integer variable in IndraLogic.
2. Provide this variable in the symbol configuration.
3. Enter the instance path of this variable into the "ChannelNo. to PLC" dialog.

"IndraWorks Operation" writes the number of the active channel to this variable at runtime. Alternatively, changing this variable in the PLC causes that the active channel in "IndraWorks Operation" is switched.

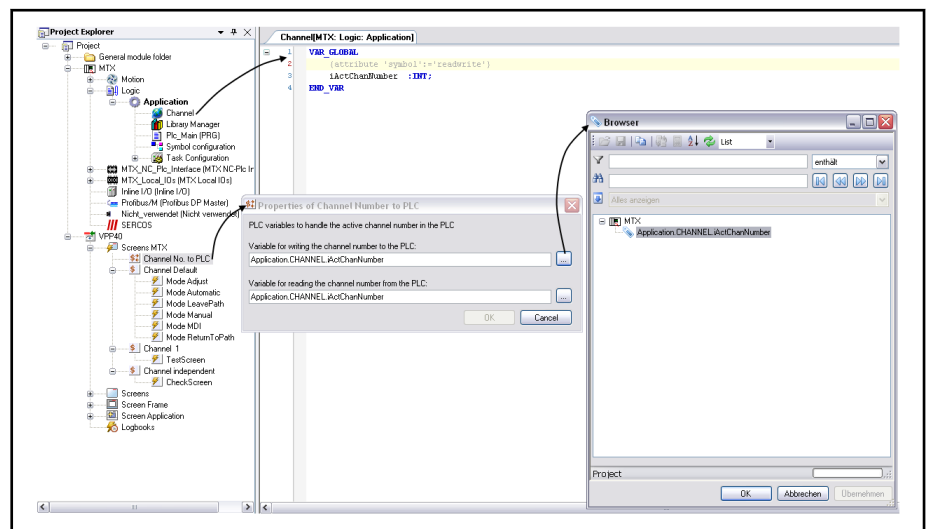


Fig. 6-14: Example of the channel number in the PLC

6.6 NC simulation

6.6.1 General information

The NC simulation is a graphic simulation for CNC-controlled machine tools with the following functions:

- Material removal and material application simulation
- Displaying the path of the programmed tool point as polyline (tool path)
- Simulation of the machine tool motions including collision detection

The simulation can be used in multiple configuration levels:

Basic license

The basic license facilitates the workpiece simulation as material removal simulation and as tool path graphic. The definition of the raw part geometry is limited to semi-finished products such as cuboids, sheets and cylinders. The

Simulation of the virtual machine	tool geometry is calculated based on their technology data (generically generated tool models). An accurate tool representation based CAD models as well as the representation of parts or the entire tool machine is not possible. A collision can only be detected between a workpiece and the currently displayed tool (its non-cutting part).
Virtual Machine Builder	The configuration level "Simulation virtual machine" extends the functionality by the use of CAD models. Using CAD models, workpieces of any form, tools or tool holders, fixtures up to an entire machine (machine simulation) can be represented in detail and then material removal and collision analysis can be executed under realistic conditions. The "Virtual Machine Builder" is an additional program for a user-guided and graphically supported creation or to edit machine models installed and used as stand-alone program (without IndraWorks context). The machine model consists of the following two components: • Machine definition This is a description of the machine structure on how it is composed of moving and stationary elements, where and in which direction the NC axes move (kinematic tree) and where the reference points for tools (tool nodes) and workpieces (stock nodes) are. • A CAD model set (optional) The files have to be in STL format (STereo Lithography). They define the physical design of machine parts, tools, clamping devices and workpieces.



The NC simulation can only be configured on the operating station in IndraWorks Operation.

For details on the NC simulation, refer to the documentation "MTX 15VRS NC Simulation Multitouch" (DOK-MTX***-NC*SIM*M*15-APRS-DE-P).

6.6.2 Simulation models

The machine model includes a description of the machine structure and how it is composed of moving and stationary elements. It includes information on where, in which direction and to which traversing ranges the NC axes are applied to. This part is also called **Kinematic model**. The references for tools ("Tool" nodes) and workpieces ("Stock" nodes) are also defined in the machine model.

The physical design of machine parts, tools, clamping devices and workpieces is optionally subject to CAD models in the STL format.

Simulation data is managed in IndraWorks Engineering in the Project Explorer below the "Models" node.

Managing simulation models in the Project Explorer

1. Creating the "Models" node in the Project Explorer

The "Models" node is dragged from the "Simulation" library into the Project Explorer.

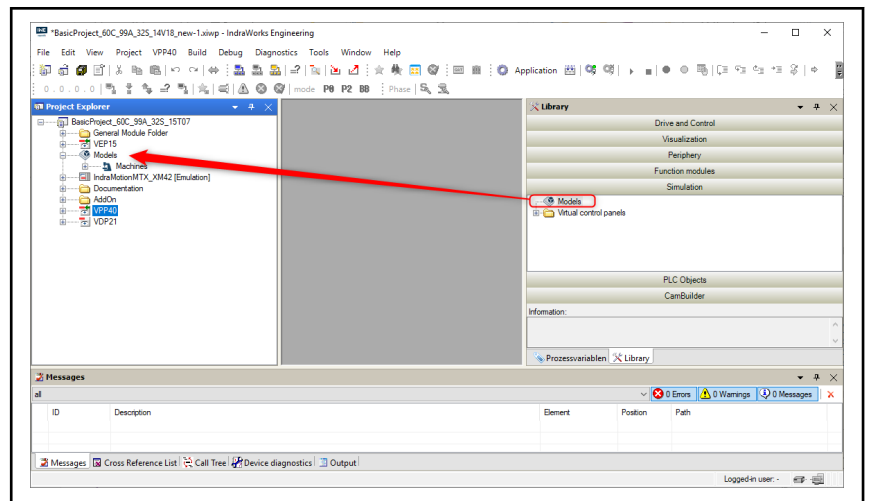


Fig. 6-15: Creating the "Models" node in the Project Explorer

The new "Models" node then includes a "Machines" child node. One or multiple machine models can be managed below this child node.

2. Adding a machine model

Select the menu function **Add** to add a model created with the Virtual Machine Builder. Select the storage folder of the machine model in the open dialog and open it with <Open>. The relevant XML file is selected and can also be opened with <Open>. The machine model is shown under the "Machine" node.

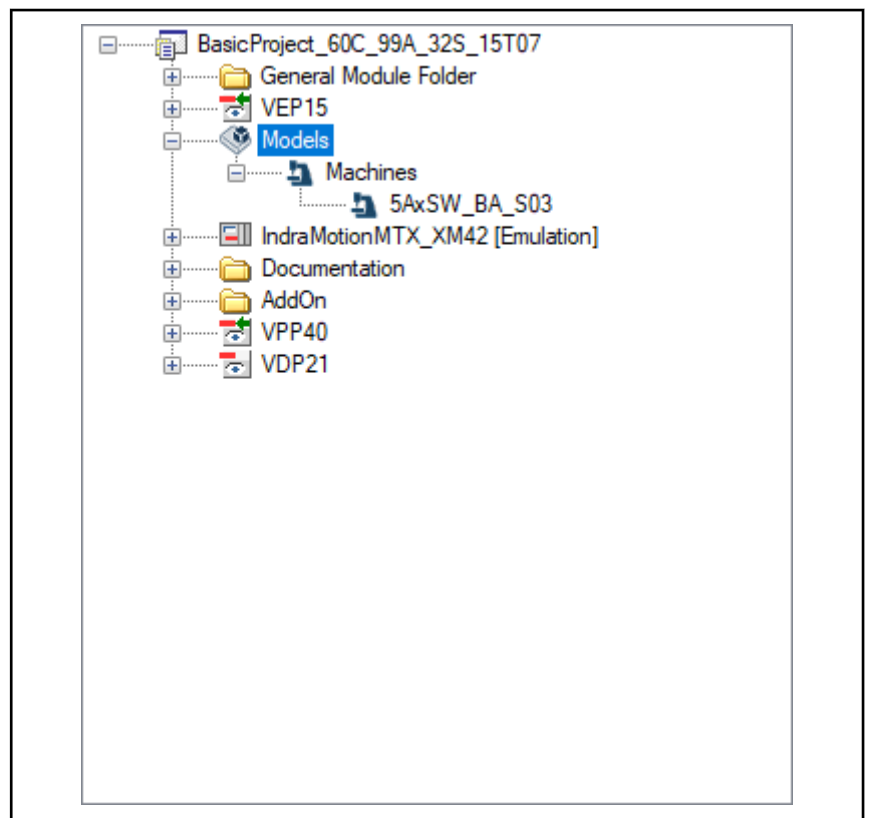


Fig. 6-16: "Models" node with a machine model in the Project Explorer

3. Enabling the machine model to be used in IndraWorks Operation

To enable the machine model in IndraWorks Operation, use the menu item **Active model**.

The model node is then shown a "enabled" (bold).

4. Applying the machine model to IndraWorks Operation

To apply the active machine model, go to **Visualization data ► Transmit and enable...** on the node of the visualization device.



Only one model can be used in IndraWorks Operation.



In IndraWorks Engineering, this data is archived, added to version control as well as exported and imported.

Data exchange between IndraWorks Operation and IndraWorks Engineering is started via the context menu of the HMI node.

- IndraWorks Operation --> IndraWorks Engineering: Visualization Data \ Loading changes from operating station
 - IndraWorks Engineering --> IndraWorks Operation: Visualization Data \ Transfer and Activate
-

7 Control commissioning

7.1 Introduction

The commissioning of an MTX control is basically determined by control data, drive data and the PLC program.

Control data Control data is exclusively available in the control, but it can be stored in an archive in the IndraWorks project. They consist of all data stored in the file system of the control and of the machine parameters determining the basic response of the control. Files on the control are either NC programs or configuration files for the tool management, variable lists, D-corrections, zero points, placements and system data.

Drive parameters Drive parameters determine the basic responses of a drive. They are stored on the drive, but can also be provided via the offline parameterization in IndraWorks Engineering.

PLC program The PLC program controls the response of the periphery. It is stored as compilation on the file system while the sources for the project planning and configuration are stored in IndraWorks Engineering.

7.2 MTX control, configuration

7.2.1 Configuring the MTX

General information



This configuration is available for the following actions:

- Wizard when creating an MTX
 - Menu item **Properties** in the MTX context menu
-

Fig. 7-1: Dialog: Configure IndraMotion MTX (General information)

Enter the general information on the device:

- Device name:
Name of the device displayed in the Project Explorer. The name has to comply with the IEC 61131-3 standard.
- Comment:
Comment on the device. This is also shown as tooltip in the Project Explorer on the device.
- Created by:
Author or editor of the device.
The Windows user is entered as presetting.

Configuration and communication



This configuration is available for the following actions:

- Wizard when creating an MTX
- Menu item **Properties** in the MTX context menu

Configure IndraMotionMTX

Configuration
Configure the control as well as the communication and PLC settings

Device configuration

Device type: IndraMotion MTX XM42

Version: MTX15VRS

Firmware release: FW*-MTX-XM42-23.1 (15T01)

Communication

☐ Emulation Normal NC program processing

☒ IP address: 192.168.1.1 Port: 10099

PLC gateway: localhost

☐ Transmit PLC sources together with boot application

☒ Secure online mode of PLC

Timeout (sec): 10

Ethernet interface of control

Configure

<< Back Next >> Finish Cancel Help

Fig. 7-2: Dialog: Configure IndraMotion MTX (configuration and communication)

Select the settings affecting the device version and the communication to the control.

The configuration is divided into four areas:

- **Device configuration**

Select which device variant and which firmware release is to be used.

 - **Device type**

Select the hardware of the control here.

Upon online switching it is verified if the connected hardware is compatible with the selected hardware.
 - **Version**

The version is used as filter to display the firmware release.

Only versions supported in the current compatibility mode can be selected.
 - **Firmware release**

The firmware release specifies the firmware version within the version.

The firmware can subsequently be modified. If the project has been loaded in compatibility mode, the compatibility mode has to be updated to the current or required IndraWorks version.
- **Communication**
 - **IP address:**

IP address to reach the control.

Default value: 192.168.1.1.

– **Port:**

Port to reach the control.

Default value: 10099.

Normally, this value does not have to be changed.

– **Timeout (s):**

Period in which the communication requests have to be answered to avoid an error report.

Default value: 10 s.

Normally, this value does not have to be changed.

- **Ethernet interface of the control**

- **Configuration**

Calling the Ethernet configuration under: [Ethernet configuration, page 146](#).

After changing the IP address, it is immediately applied to the project.



Mount drives have to be entered again. The entry is available on the control, but not enabled on the NFS server.

This function is only available if there is an online connection to the control.

- **IndraLogic**

- **Gateway IP address**

The gateway IP address specifies on which network participant the IndraLogic gateway is used.

Default value: localhost. The gateway runs on the Engineering PC.

- **Secure online mode**

A security prompt is sent in online mode before each action causing changes in the data of the PLC runtime system.

Configuring function modules



This configuration is available for the following actions:

- Wizard when creating an MTX
 - Menu item **Properties** in the MTX context menu
-



The options in the dialog to configure the function modules can be distinguished according to the device type.

Configure IndraMotionMTX

Interfaces
Select the onboard components and function modules to be used.

BRC:NETX_ETHXM: PROFINET IO Controller XM

Ethernet (XF5): Not used

Extensionmodule: Profibus (XFE01.1-FB-10)

Safety CPU (XFE01.1-SY-01)

Not used

<< Back Finish Cancel Help

Fig. 7-3: Dialog: Configure IndraMotion MTX (function modules)

The interface and optional function modules of the control can be managed here:

- **Realtime Ethernet configuration**

Current Profinet or Ethernet/IP configuration that can be selected when inserting the device. "Not used" or Profinet or Ethernet/IP interface can optionally be selected.

- **Ethernet configuration**

Current Ethernet/IP configuration that can be selected when inserting the device. Ethernet/IP-adapter and "Not used" are the available options.

- **Function modules**

Up to three function modules can be coupled to the "IndraMotion MTX". If required, select the function modules from the respective lists. The logic order of the created function modules has to correspond to the physical function module position at the control.



If a Safety function module (Safety CPU) is connected to the control, make sure that in the function module order, this Safety function module has to be positioned at the last or lowest position of the configured function modules.

Device data



This configuration is available for the following actions:

- Menu item **Properties** in the MTX context menu
- An existing online connection to the MTX is a prerequisite.

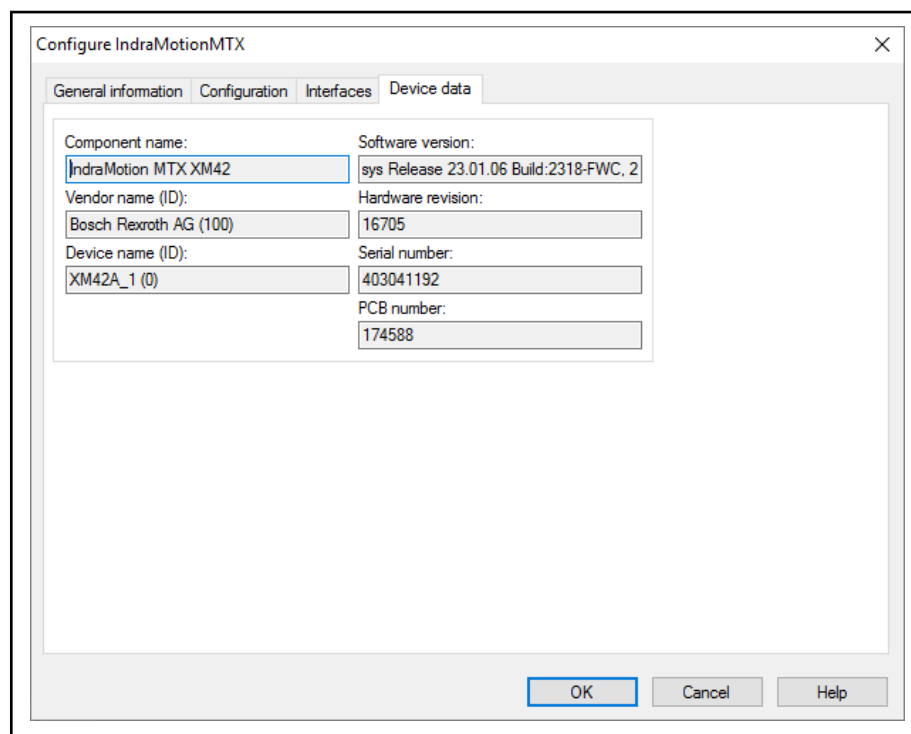


Fig. 7-4: View of device data in the MTX properties

The information on the "Electronic Label" of the control is shown here. This information is used to identify the device.

Configuring the Ethernet interface

Open it either via the context menu or in the "Properties" dialog of the MTX on the "Communication Settings" tab.

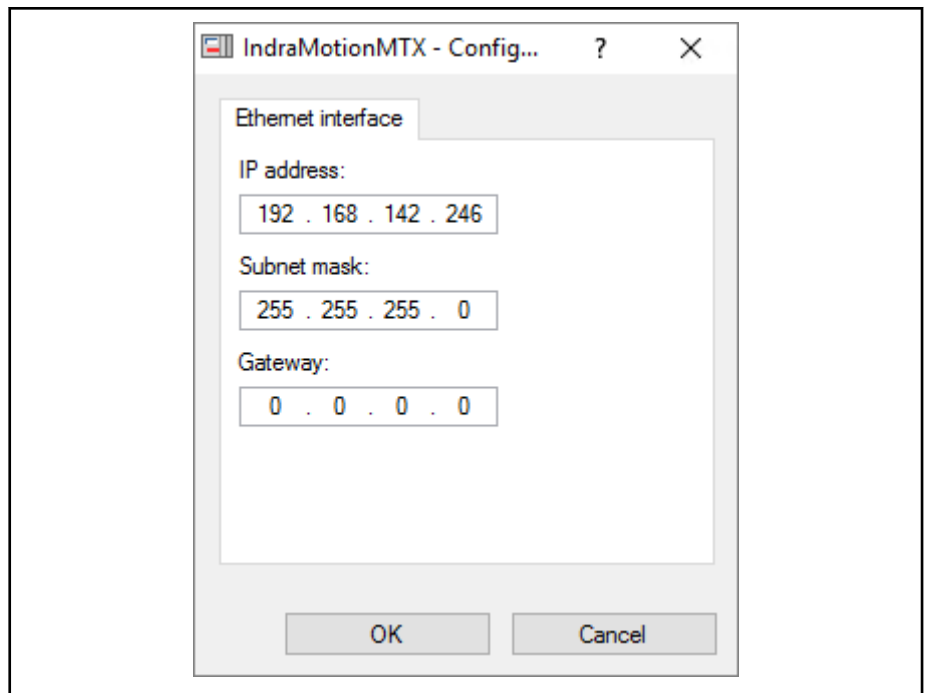


Fig. 7-5: Configuring the Ethernet interface of the MTX control

The parameters of the network interface of the control can be adjusted here:

- **IP address:**
IP address used by the control to report in the network or under which it can be reached.
- **Gateway:**
The control establishes a connection to the network via the gateway. Gateway and control have to be in the same subnet.



Changes of the Ethernet interface are applied after a control restart. When the IP address changes, the setting in the project is additionally adjusted. After a successful control startup, the project can be immediately reconnected.

IP addresses on the NFS server are not modified upon an IP address change. The mount drives have to be configured or the changes have to be made directly into the "export.us." file.

7.3 Adding the MTX control to the project

7.3.1 Creating a project

To create a new project, click in the start screen on "Create Empty Project".

A new project can also be created via **File ► New ► Project...**

The following settings can be made in the wizard to create a new IndraWorks project:

- **Name**
Freely selectable project name
- **Directory**
Directory in which a new project is created.
The default setting is the directory "My Documents".

- **Project language**

By selecting the project language, the language of the user texts is determined.

The newly created project is displayed in the Project Explorer with the previously defined project name.

7.3.2 Adding MTX to the project

There are the following options to create a control in the project:

- **Inserting from the library**

Drag the device (e.g. **IndraMotion MTX ► IndraMotion MTX XM42 performance**) from the "Drive and Control" library to the project in the Project Explorer.

- **Inserting via context menu**

Select the "Add" context menu item at the project node. The device can be selected from the opened list.

The wizard to configure the MTX starts subsequently (see [chapter 7.2 "MTX control, configuration" on page 141](#)).

7.4 MTX control, export and import

7.4.1 Exporting MTX nodes

Starting the process

Select **Export...** in the context menu of the MTX node.

User input

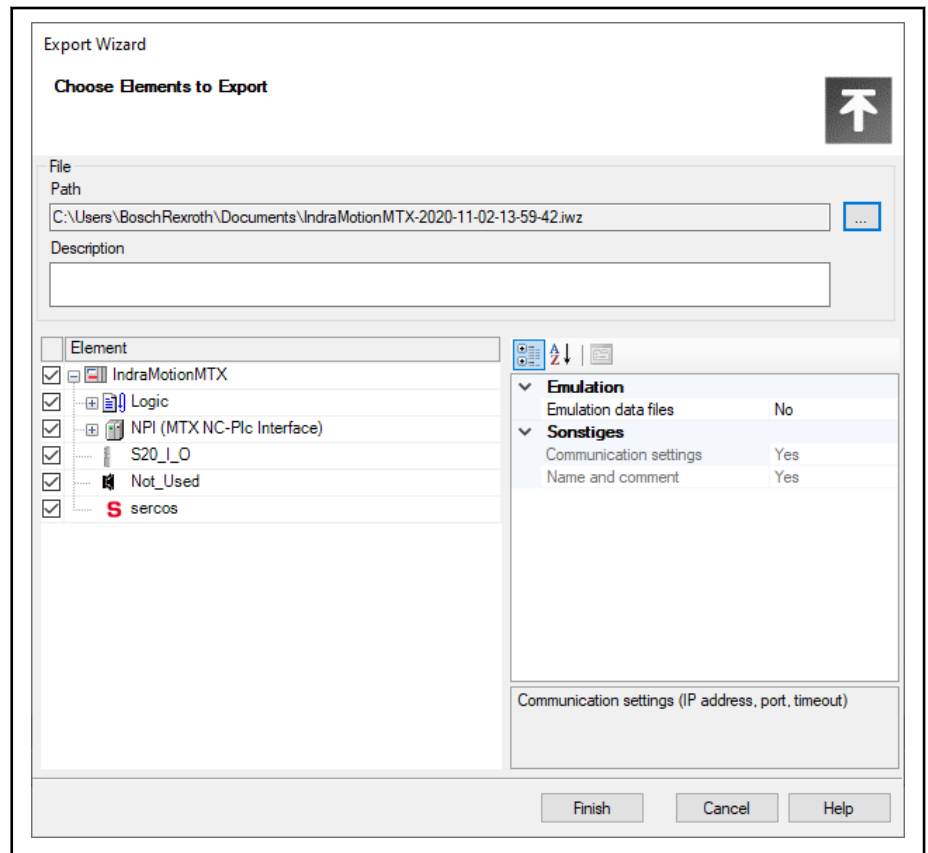


Fig. 7-6: Dialog: Export wizard (exporting an IndraMotion MTX)

The storage location and the name of the export file are specified in the export wizard.

The project elements to be exported can be selected in the tree.

Some project elements provide an extended selection. Individual archives stored in the project can be selected and deselected for the MTX. All archives are selected by default.

The following archive types are available:

- Archives of the drive data
- Archives of the control data
- Archives of the extended data

General information such as communication settings and name are always exported and can thus not be deselected.

After the selection, press "Finish".

7.4.2 Importing MTX nodes

Starting the process

Select **Import...** in the context menu of the MTX node and then the file to be imported.

User input

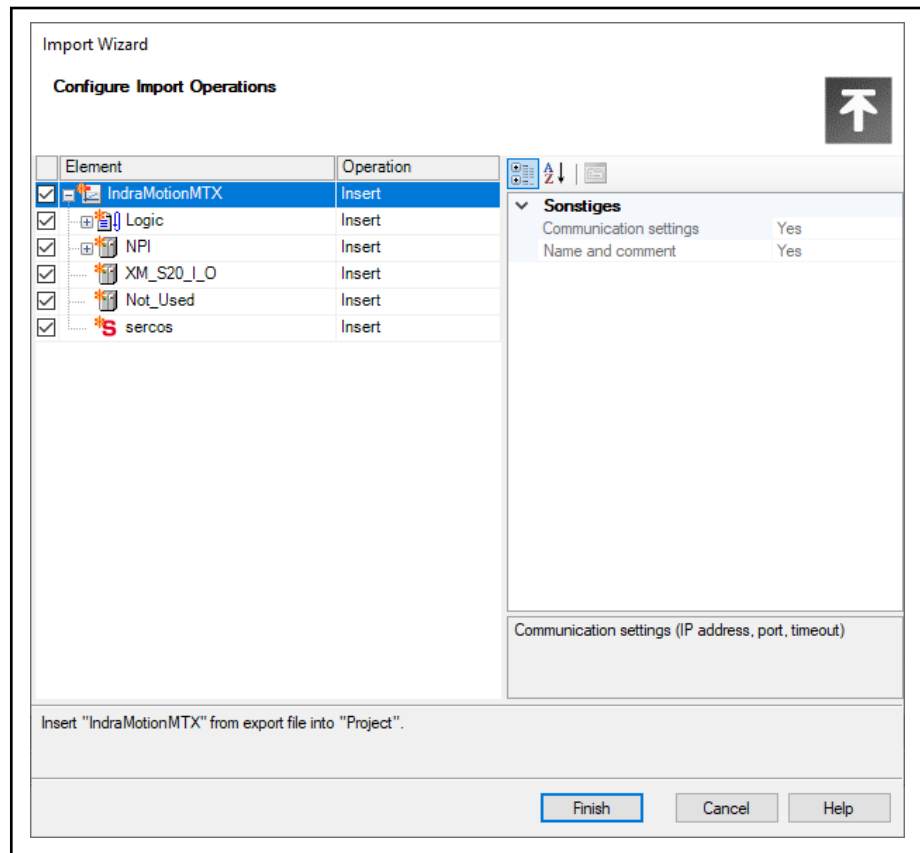


Fig. 7-7: Dialog: Configure import operations (import of an IndraMotion MTX)

All existing elements are displayed in the tree.

It can be selected for each element whether it is to be imported.

The MTX provides the following extended selection:

- Archives**
 All archives in the export file are listed. By default, archives are only applied to the project if not an archive exists with the same name.
- Communication settings**
 While overwriting, communication settings are only applied if selected explicitly before.
- Name and comment**
 While overwriting, device name and comment are only applied if selected explicitly before.

User input when importing the old control generation

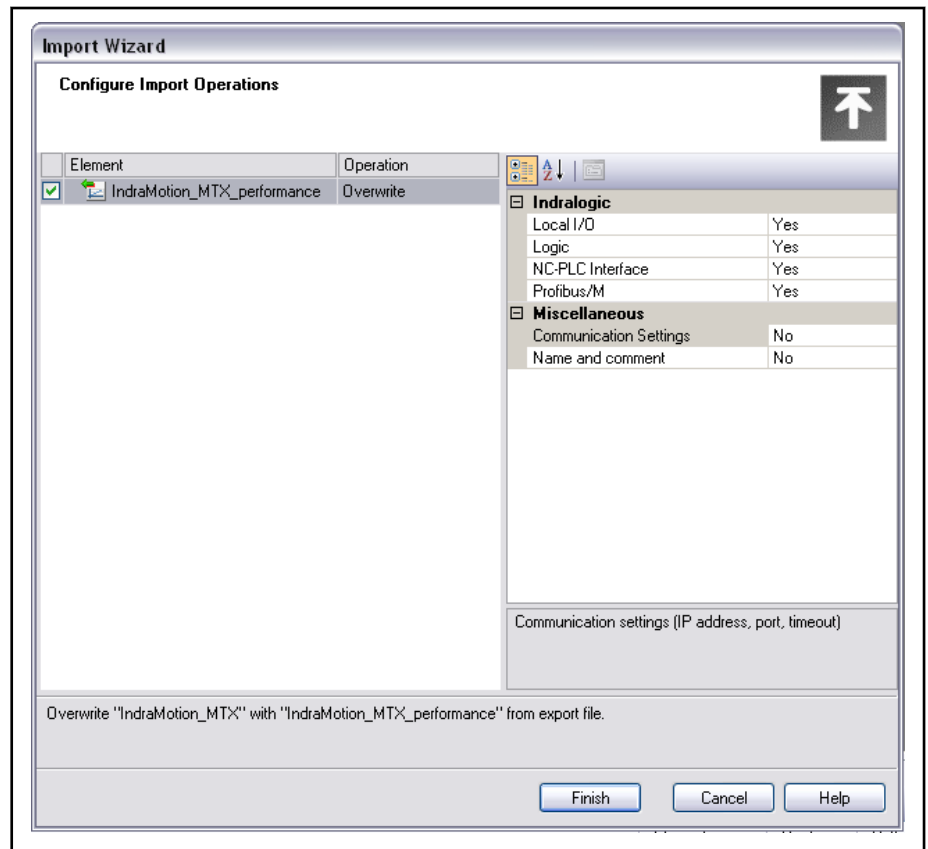


Fig. 7-8: Dialog: Configure import operations (import of an IndraMotion MTX)

Only the MTX is displayed in the tree.

The PLC parts imported by the 1G can be selected in the advanced settings.

When selecting the "Logic", the contained elements (e.g. POU's, variable lists, libraries) are subsequently selected as in case of the Logic data transfer.

7.5 MTX Emulation

7.5.1 General information

The "MTX Emulation" maps the response of an MTX control. Depending on the set control type (e.g. IndraMotion MTX XM42), the compatible MTX-Emulation is always executed.

Thus, it is possible develop and commission a significant part without hardware, i.e. without a real MTX control.

7.5.2 Starting and completing

The "MTX Emulation" is activated and deactivated using the context menu of the control node.

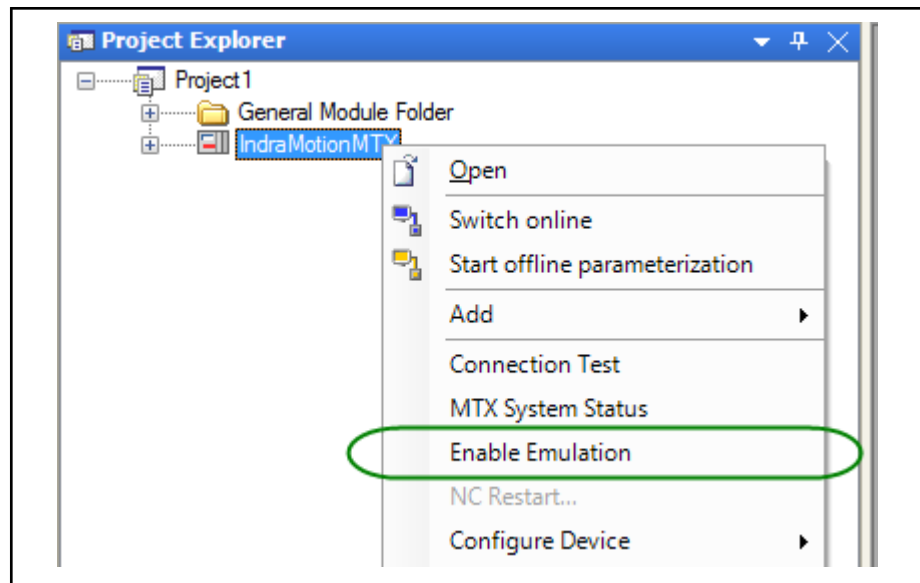


Fig. 7-9: Activating the emulation

At the control node and its context menu, it is displayed that the MTX-Emulation is active.

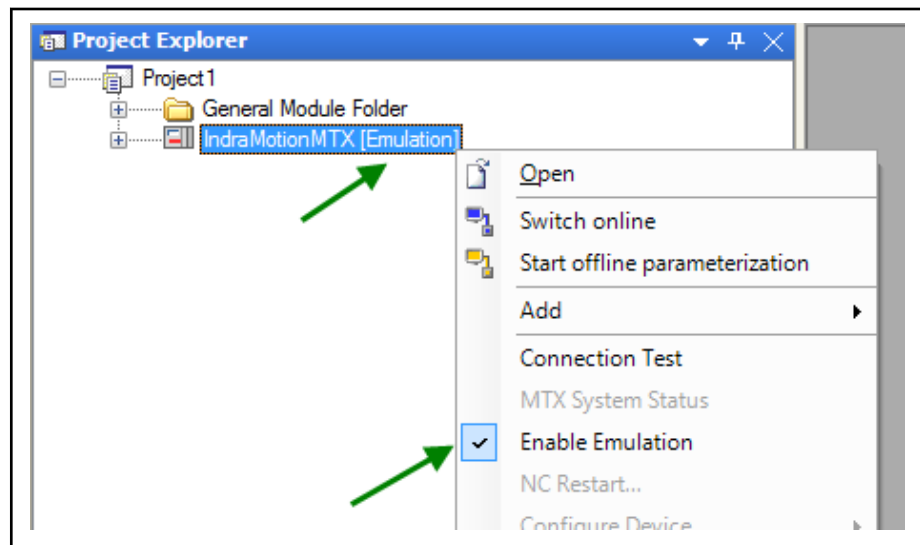


Fig. 7-10: Emulation active

As long as the MTX-Emulation is active, all control-specific actions (e.g. on-line, NC configuration etc) act on the MTX-Emulation and not on the configured control address (IP address).

A maximum of 10 emulations can be active simultaneously!

7.5.3 Data storage

All data (e.g. machine parameters, system data, etc.) of an MTX emulation are stored related to the project or the control but are not part of the (*.xiwp) project file and cannot be added to version control (VCS). The emulation data is contained in the export of the control. If there are several controls in a project, the data is available separately for each control.

7.5.4 Compatibility

Depending on the selected device version, the compatible emulation version is automatically started. Thus, it is ensured that the emulation data always match the selected version.

If no compatible emulation version is installed, a message is output upon start. Subsequently, a newer emulation version can be used. In this case, ensure that only functions are used that are available in the selected version.



Changes made in a version or in a release delete the emulation content. Thus, always create a backup before upgrading to another version or another release.

7.6 MTX system status

7.6.1 Introduction

The MTX system status is a commissioning and diagnostic application for MTX controls.

This application controls, monitors and visualizes the state of the following component groups:

- MTX standard XM42
- MTX performance XM42
- MTX advanced XM42

This application can be accessed via the device node context menu in IndraWorks Engineering.



The control does not have to go online for this application!

After opening the application, it is displayed in IndraWorks Engineering.

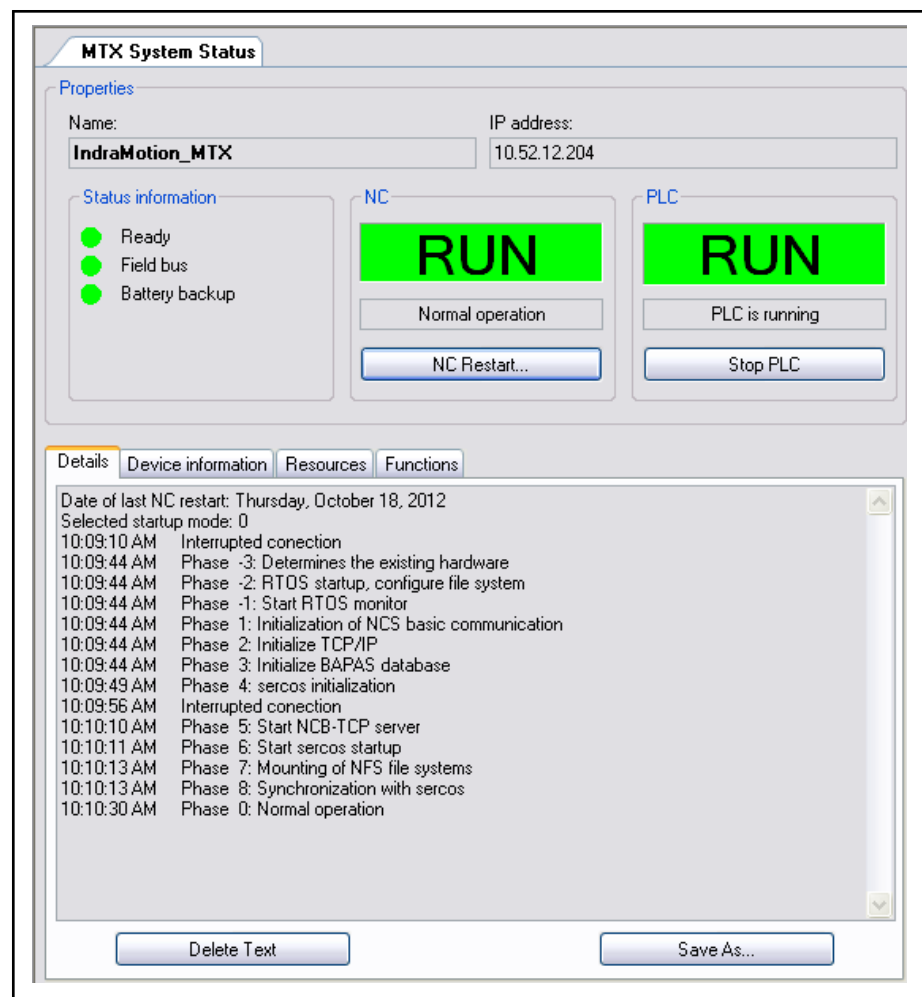


Fig. 7-11: MTX system status

The MTX system status is divided into two sections.

The general information which is divided into several sections and which is always visible to the user can be found in the upper section.

Three tabs which can be changed by the user contain various information and can be found in the lower section:

- "Details" tab
- "Device Information" tab
- "Resources" tab

7.6.2 General information

"Name and IP address" field

This is an area of the upper application section in which the following information is displayed:

- **Name**
The name to be configured for the respective control in the project tree is displayed. A name change in the project tree becomes immediately visible in the application.
- **IP address**
The IP address entered into the network configuration is visualized.

Field "Status info"

The status information of the control is displayed using red and green LEDs:

- **Ready**
Green LED = Ready signal is 1
Red LED = Ready signal is 0
Gray LED = Ready signal is not supported by the control
- **Field bus**
Green LED = Field bus function ok
Red LED = Problem with field bus
- **Backup battery**
Green LED = Correct buffer battery state
Red LED = Buffer battery discharged or buffer battery not available

"NC" field

In this area, the NC information is displayed:

- **Status field of NC**
This field visualizes the current NC status which is displayed in color and short form. The used abbreviations are described in [chapter "Start-up" on page 158](#).
This field is flashing during control startup.
- **Info field with long status name**
The abbreviations in the status field are explained to the user in this field.
- **Button for NC restart**
By means of this button, the user can execute an NC restart. After actuating this button, the following dialog is opened in which the user can specify the Startup mode (see [chapter "Startup mode" on page 159](#)) for the restart:

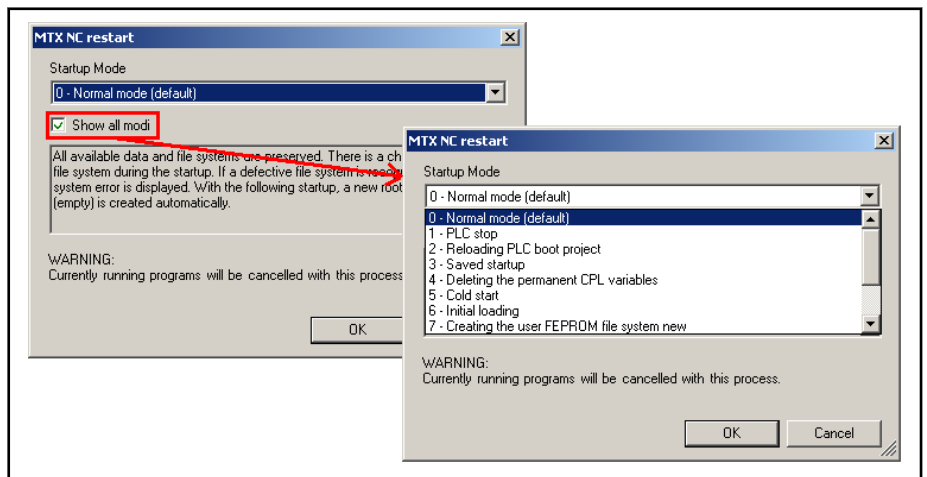


Fig. 7-12: MTX NC restart

If the checkbox "Show all modes" is not selected, the user can only choose between 3 modes.

- 0 - Normal mode (standard)
- 6 - Bootstrapping
- 7 - Creating the user FEPRM file system

"PLC" field

In this area, the PLC information is displayed:

- **Status field of PLC**
This field visualizes the current PLC status. This status is displayed in color and as a short form.
 - **RUN** is displayed on green background during normal mode in the display.
 - If the PLC is at standstill, **STOP** is displayed on red background.
- **Info field with long status name**
The abbreviations in the status field are explained to the user in this field. The following long names exist:
 - RUN = PLC is running
 - STOP = PLC is at standstill
- **Button to start/stop the PLC**
If the PLC is running, it can be started or stopped using this button.

7.6.3 Tabs

"Details" tab

In this tab, the results of the last control restart **executed using this application** are visualized. Date, time, phase and their meanings are displayed. The display is refreshed after each restart.

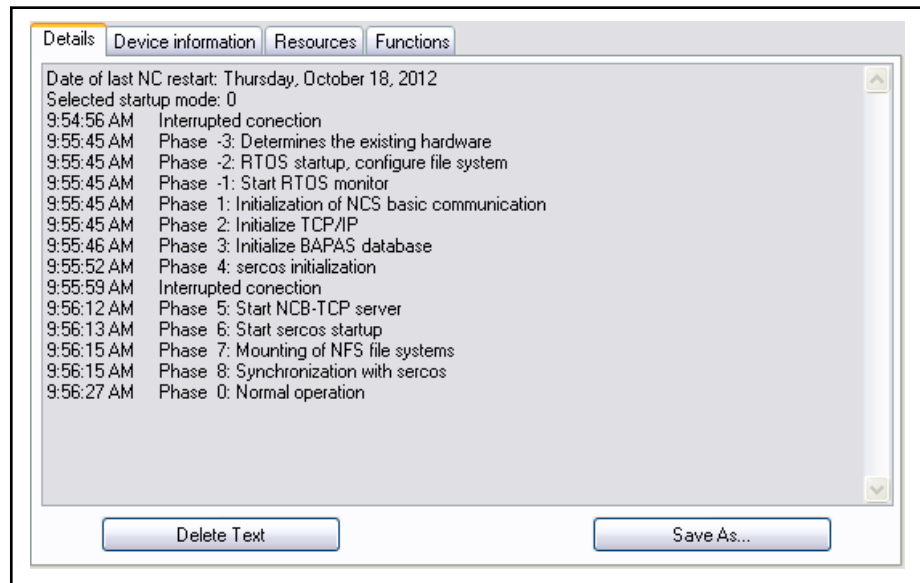


Fig. 7-13: "Details" tab

If required, this information can be saved in a *.log file. The path under which the file is to be saved is arbitrary. The application saves this information under the default name "Details.log".

"Device information" tab

The following information on the control is displayed on this tab:

- Firmware version
- Circuit board number
- Hardware version

- Serial number

For more details on the firmware version, press "Details":

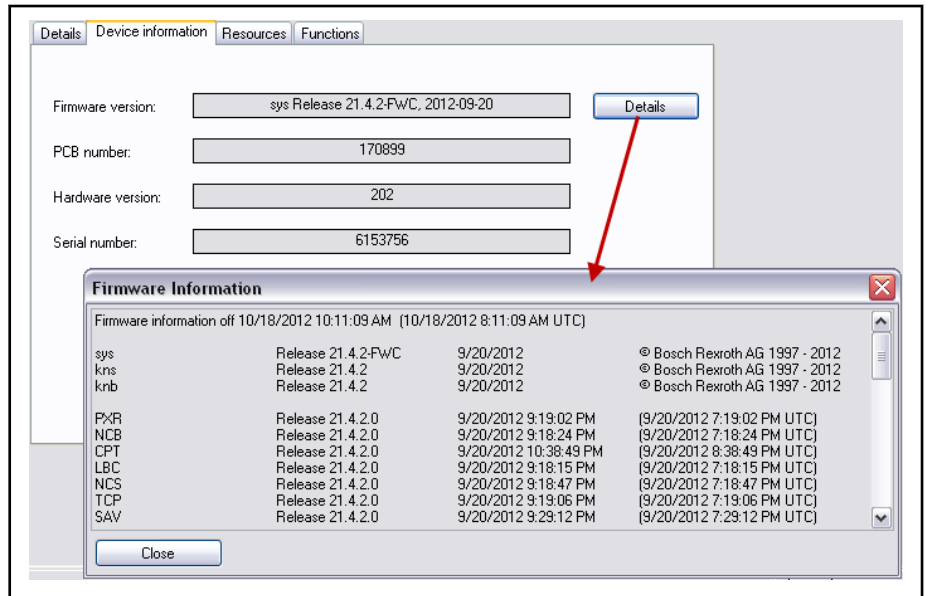


Fig. 7-14: "Device information" tab

If no connection is established to the control, the fields on this tab are blank.

"Resources" tab

The current memory usage is displayed in this tab.



The system of the MTX control requires memory of the basic functions. The memory available reduces by this value.

The following information is visualized:

- Memory of the permanent CPL variables (approx. 1% of the system is assigned)
- Memory of the persistent SD data (approx. 1% of the system is assigned)
- Memory of the RAM file system (approx. 30% of the system is assigned)

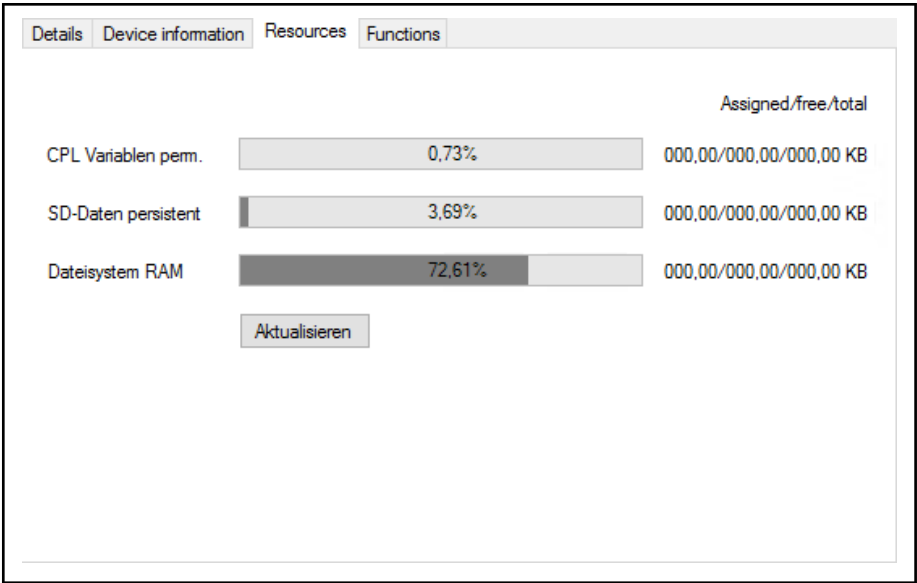


Fig. 7-15: "Resources" tab

The bar visualizes the memory consumption in percent. For detailed information, go to the tooltip of the corresponding bar.

"Functions"

A core dump for the control can be created in this tab.



Create a core dump only in case of error or upon request. If a core dump is created, the control is restarted without further prompts.

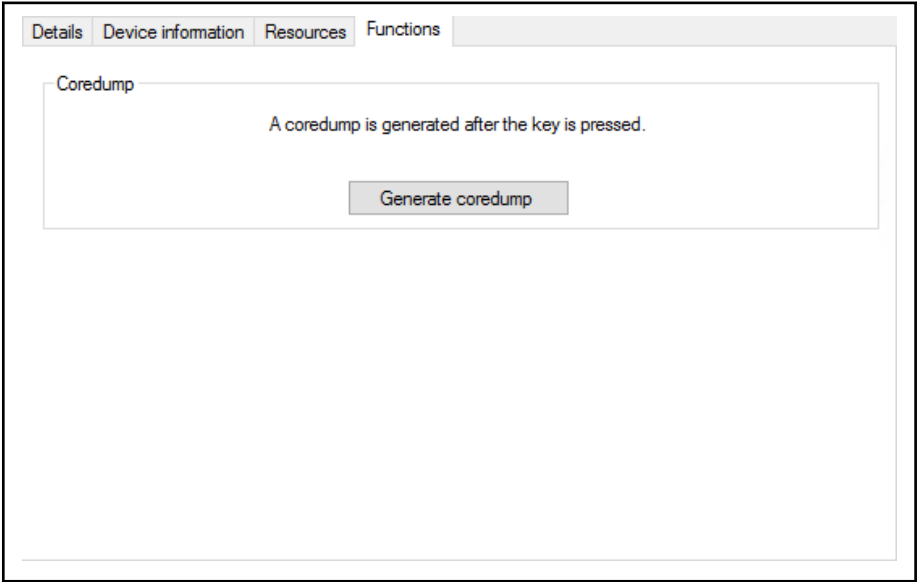


Fig. 7-16: "Functions" tab

7.6.4 Startup configuration

Startup

MTX starts when switching on the control or while triggering an NC restart.

The startup occurs synchronized in 12 phases displayed in the MTX system status:

P: -3	Determining the existing hardware
P: -2	RTOS startup, configuring file systems
P: -1	Starting RTOS monitor
P: 1	Initializing the basic NCS communication
P: 2	Initializing TCP/IP
P: 3	Initializing BAPAS database
P: 4	Sercos initialization
P: 5	Starting NCB-TCP server
P: 6	Starting Sercos startup
P: 7	Mounting NFS file systems
P: 8	Synchronization with Sercos
P: 9	Sharing NCB-TCP server (communication with user interface)
RUN	Normal operation

Tab. 7-1: Display of startup phases

For critical system errors, boot panic errors or active MTX shutdown, the display changes to:

SF	A critical system fault is pending
BP	A boot panic error is pending
SD	Shutdown active

Tab. 7-2: Error status display

Startup mode

The startup mode determines the behavior of the MTX at startup.

The startup mode changes are applied at next startup.

Startup mode	Description
0	Normal operation All existing data and file systems are retained. The root file system is checked at startup. If a defective file system is detected, a critical system error is displayed. A new (empty) root file system is automatically created during the next startup.
1	PLC stop The behavior corresponds to Startup mode 0 with the difference that the PLC remains in the STOP state and the PLC user program is not processed.
2	Reloading the PLC boot project The PLC boot project is loaded from the user FEPROM. Any PLC boot project that exists in the root file system is discarded. Otherwise, the behavior corresponds to the startup mode 0.

Startup mode	Description
3	Save startup In rare cases, due to faulty machine parameter specifications, a control startup can be impossible. Startup mode 3 carries out a startup in this error situation irrespective of the set machine parameters. A startup with the minimum configuration is executed. The machine parameters set are ignored. After the startup, the invalid machine parameter settings can be corrected and a new startup with Startup mode 0 can be carried out.
4	Deleting the permanent CPL variables The permanent CPL variables are deleted. Otherwise, the behavior would correspond to Startup mode 0.
5	Cold start The power-up management logic is not run through. Otherwise, the behavior would correspond to Startup mode 0.
6	Bootstrap A new root file system is created. As a result, all old file system data is lost. If an intact user FEPROM file system is available, the PLC boot project and the configuration data are loaded and copied to the root file system.
7	Creating the user FEPROM file system again The user FEPROM is created again. As a result, all the old file system data is lost. This is required, for example, if a user FEPROM file system is defective. The root file system is retained. The permanent CPL variables are deleted.
9	Debug mode This is normally for debugging if the control does not automatically boot after a reset. After the basic monitor has been initialized, the boot loader is enabled and the subsystems are automatically loaded.
10	Debug mode (without automatic loading) After the basic monitor has been initialized, the boot loader is enabled. Further loading can take place via TCP/IP.
11	Debug mode (without activating the boot loader) The basic monitor is initialized. Further loading can take place via TCP/IP.
15	Debug mode (basic monitor start) Only the basic monitor is enabled.

Tab. 7-3: Startup mode

7.7 MTX - Setting the time and date

General information For a correct chronological assignment of diagnostic messages, date and time of all devices in the machine network have to be identical. The adjustment of date and time of an MTX control is described in the following.

Setting date and time 1. Going online with the control

- Opening the context menu of the control and open the configuration dialog "Set time and date..." via "Configure device..."

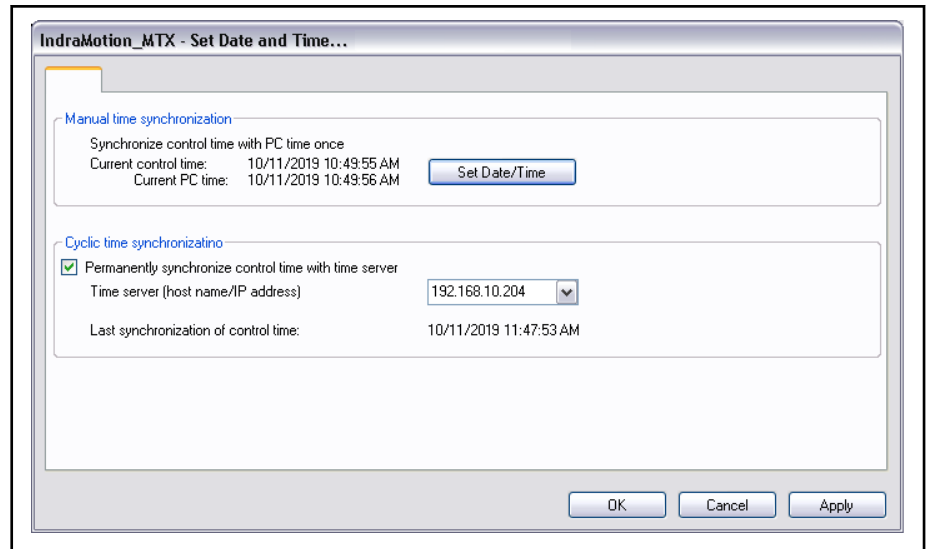


Fig. 7-17: Dialog "Set time and date"

To assign the current date and time to the control, use the options "Manual time synchronization" and "Cyclic time synchronization":

Manual time synchronization

The "Manual time synchronization" is used to apply the time setting of the visualization computer to the control using the **Set Date/Time** button.

The display indicates the current control time and the current PLC time.

There is no further adjustment between the control time and the PC time at runtime.

Cyclic time synchronization

The "Cyclic time synchronization" can be enabled via a checkbox.

Enter the IP address of the time server in the input field. Each Windows PC in the network can be used as time server if it is configured as synchronized time source.

"Apply" saves all settings and checks the connection between the time server and the control.

If the synchronization was successful, the last synchronization point in time is displayed and the time of the control is regularly compared with the time server.

If the configuration is incorrect (e.g. time server is not configured or cannot be reached), a corresponding diagnostic message is output.



The configured time server always has to be a synchronized time source. Otherwise, the control does not apply its time setting and the synchronization fails.



Refer to the following sections for the basic procedure to set up a time server for the Windows operating system. Only qualified personnel may make changes!

Windows 7 Proceed as follows to configure the Windows 7 computer as external time source:

1. Click on **Start**, then **Execute**, enter **regedit** and click on **OK**.
2. Select the following registration key and click on it:
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\W32Time\Config\AnnounceFlags
3. In the screen area to the right, right-click on **AnnounceFlags** and then on **Change**.
4. Enter the value **5** in the **Value** field under **Edit DWORD value** and click on **OK**.
5. For more information, refer to **Microsoft-Support** <http://support.microsoft.com/kb/314054/de>.

Windows 10 Proceed as follows to configure the Windows 10 computer as external time source:

1. Click on **Start**, then **Execute**, enter **regedit** and click on **OK**.
2. Select the following registration key and click on it:
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\services\W32Time\TimeProviders\NtpServer\Enable
3. In the screen area to the right, right-click on **EnableArea** and click on **Change**.
4. Enter **"1"** for "Enable".
5. Select the following registration key and click on it:
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\W32Time\Config\AnnounceFlags
6. In the screen area to the right, right-click on **AnnounceFlags** and click on **Change**.
7. Enter the value **5** in the **Value** field under **Edit DWORD value** and click on **OK**.
8. Open the Windows firewall. Select the "Advanced settings"
Add a new rule to the input rules.
Enable the UDP port 123.
9. Open the Control Panel.
Select the "Service administration tool"
Select the "Windows Time" service in the "Service administration tool".
10. Select the start mode "Automatic".
11. Restart the PC.

For more information, refer to **Microsoft support**.



Only personnel with knowledge of the Windows operating systems may make changes in the registry and firewall.

7.8 MTX diagnostics (MZA, ProVi, VCPs)

7.8.1 Machine fault and status display (MZA)

What is MZA?

Definition In order to carry out trouble-shooting quickly in case of interruptions in the sequence of operations of machine tools, a reporting system that issues the corresponding messages to the operator in plain text is required.

The Machine fault and Status Display, abbreviated MZA in the following, permits a total of 8096 messages to be displayed in the NC user interface. These can be displayed as faults, warnings or notes.

The messages are defined in plain text in a file (MZA file) and activated when the respective marker is set by the PLC sequential program.

All MZA messages can be saved as plain text in a logbook with the corresponding timestamps.

Displaying MZA messages

Description MZA messages are displayed in message lines in the header of the user interface. By default, fault messages and warnings are shown alternatively in the upper line and notes are shown in the lower line. In addition, a corresponding bitmap is displayed if messages appear in the header. If fault messages appear, the bitmap for the fault display flashes.

Control commissioning

Single Step **Single Block** **Progr. Block** **Contin. Block** **Skip Block** **Option. Stop** **DryRun Options** **Adjust Options**

\$1 Channel1 **Automatic Contin. Block** **inactive** **Rexroth MTX** **5/22/2006 | 2:36:34 PM**

Warning 2 **Error 1**

WCS		Command	End position	Dist. to go	Program
X	mm	0.000	0.000	0.000	0.000
Y	mm	0.000	0.000	0.000	0.000
Z	mm	0.000	0.000	0.000	0.000

	Program	Command	Actual	Override	Max	Gear
F	mm/min	0.0	0.0	8%	120%	
SSP01	1/min	0.0	0.0	45%	120%	1

No program selected

G80	G01	G17	G90	G94	G8
G62	G48	G40	G43	G45	G71
OVE					
G53.1	G53.2	G53.3	G53.4	G53.5	
/database/ZO1.zot					

Tool-active	Curr. edge	D-Number
ED0		D0
/database/DC1.dct		

NC Screen **EN**

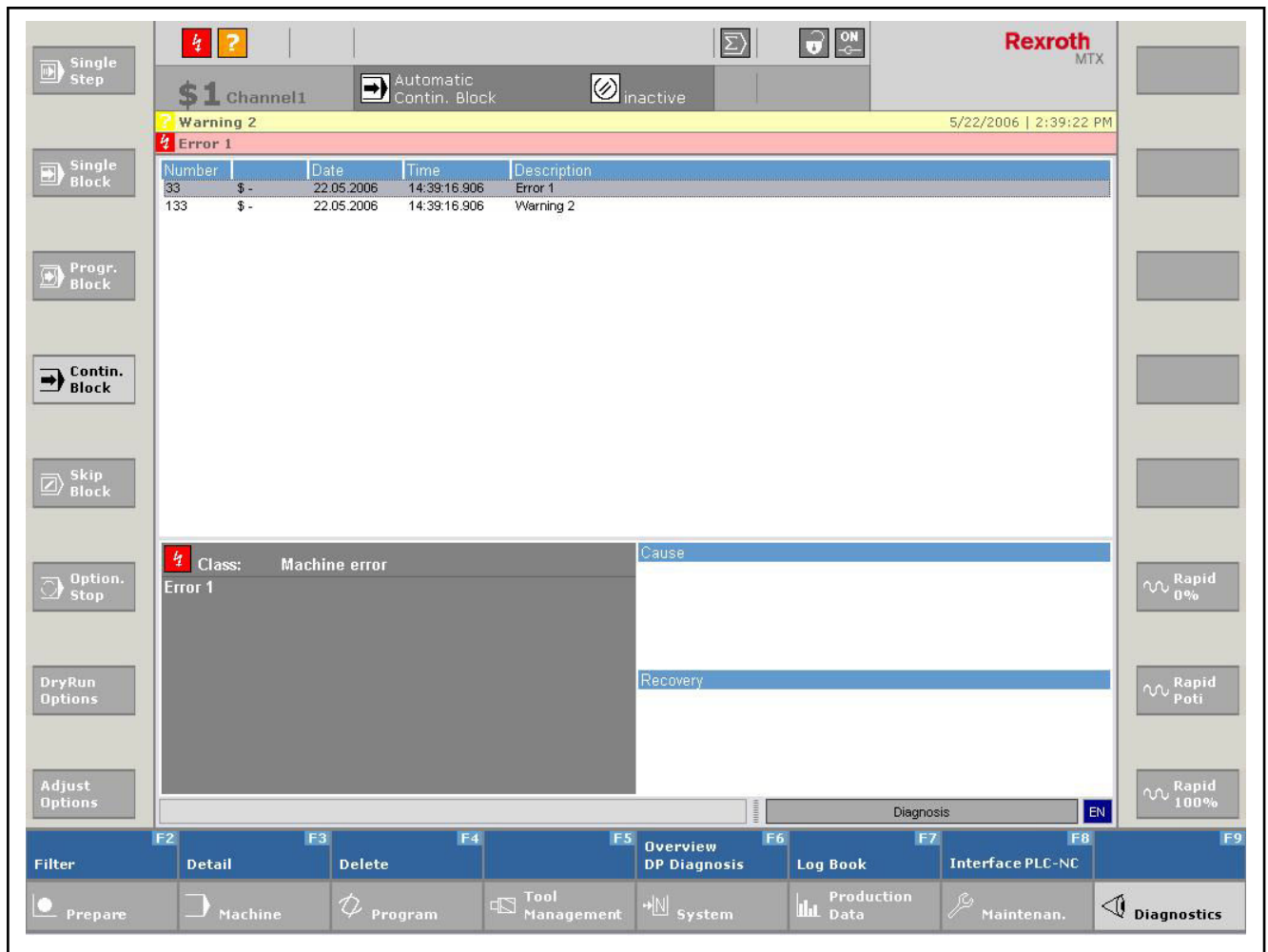
Program Selection **F2** **Active Program** **F3** **Overview Act. Corrections** **F4** **Active ZP Offsets** **F5** **Active D-Corrections** **F6** **Active Variables** **F7** **F8** **F9**

Prepare **Machine** **Program** **Tool Management** **System** **Production Data** **Maintenan.** **Diagnostics**

Rapid 0% **Rapid Poti** **Rapid 100%**

Fig. 7-18: Display of MZA messages in the header

All MZA messages that occur are displayed in the "Diagnostics" (OP9) operating area. The message that was activated last is located at the top of the list. The MZA message number (1 - 8096), the date and time of occurrence and the message text (description) are displayed for each message. In addition, cause and remedy texts can be displayed for the message.



- 1 Bitmap for fault message
- 2 Bitmap for warning
- 3 Bitmap for note

Fig. 7-19: Display of MZA messages in the "Diagnostics" operating area

If a message is selected using the cursor keys and <F7> Detail is then pressed, additional details regarding the selected message are displayed.

Further information about the "Diagnostics" screen of IndraWorks HMI can be found in the IndraWorks HMI documentation.

Commissioning procedure

The following steps are required to commission the MZA message system:

- Setting the parameters of the NC
- Generating the MZA text file(s)
- Configuring the PLC interface
- Programming the PLC interface

Parameterizing NC

The MZA message system is configured using the parameter **"Cycle Time"**. This parameter specifies the updating time of the message system in milliseconds. In general, an updating time of 500 ms is sufficient.

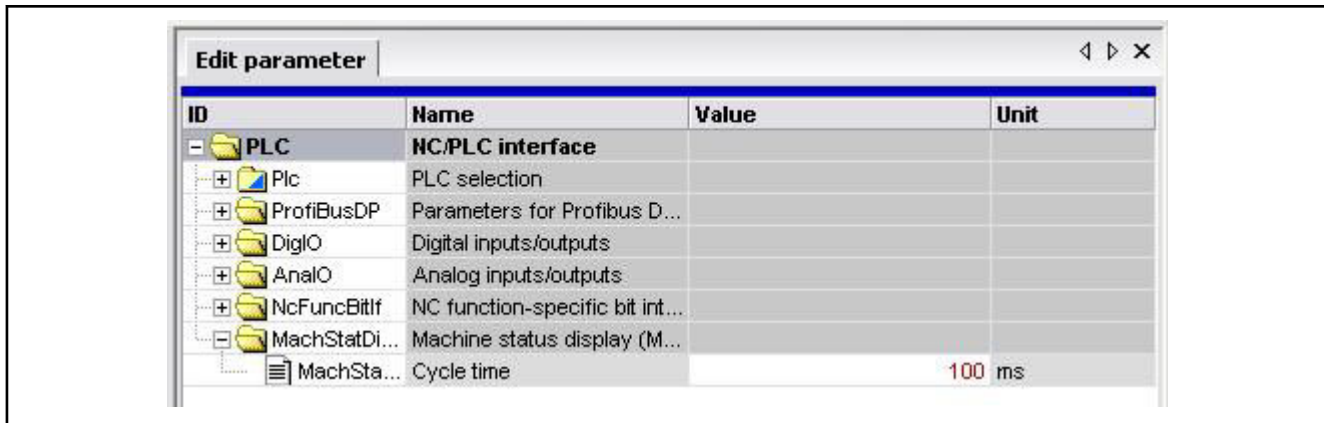


Fig. 7-20: Setting MZA message system parameters



If 0 is entered under "**Cycle Time**", no MZA messages are displayed or entered in the logbook!

Structure of MZA files

The MZA texts are entered in so-called MZA files. These are stored in the MTX file system in subdirectory **/usrfep**. The files can be saved in ANSI or UTF-8 format. The UTF-8 format is required to display Asian characters, for example.



Currently, only MZA files in ANSI format can be edited using the NC Editor. Files in UTF-8 format have to be edited externally (e.g. with the Windows editor).

A separate file is generated for each language in which the MZA messages are displayed. The file name is always **mzatexte.xxx**; the file extension **xxx** has been specified as follows for each language:



Directory **/usrfep** can be accessed only using the MTX user interface. Files can (currently) not be edited directly in this directory. Therefore, the files should be generated/saved in the **/mnt** directory (c:\mnt). Then they can be copied from **/mnt** to **/usrfep** within the MTX interface.

Language	File extension
German	.049
English	.044
French	.033
Italian	.039
Czech	.420

Tab. 7-4: File extensions



The languages German and English are always included in the scope of delivery. A corresponding language extension must be purchased as an option for every additional language.

Definition A total of 8096 messages can be defined. Each message is assigned a serial number from 1 to 8096. The number list can contain gaps.

The messages are distinguished as follows:

- Machine errors have an ID of E
- Machine warnings have an ID of W
- Machine notes have an ID of M

Precisely one ID can be assigned to each message number.

There is a message text for every message; optionally, a cause-and-remedy text can also be defined. The length of the message text is limited to 128 characters. A line break can be forced within the texts by inserting "\n". The file can contain comments in the form (<Comment>).

Structure of a message line

Program:

(<u>E,M,W</u> <1..2048>,<message text>{##<cause text>##<remedy text>})				
				remedy text max. 128 characters
			cause text max. 128 characters	
		message text max. 128 characters		
	message number			
message type				

Example:

```
(E200,error message number 200##cause of error 200##remedy of error 200)
(W201,warning number 201\nsecond line of warning 201)
```

Example MZA File (mzatexte.049)

Program:

```
(Fault in function group 1)
(E4, (FG1-3) +S2-A40/P100.1; AS-i Master 1: Configuration is inactive)
(E5, (FG1-4) +S2-A40/P100.1; AS-i Master 1: AS-i Power Fault)
(E0006, (FG1-5) +S2-A40/P100.1; AS-i Master 2: Configuration is inactive)
(E7, (FG1-6) +S2-A40/P100.1; AS-i Master 2: AS-i Power fault)

(Warnings for Function Group 7)
(W289, (FG7-0) +S2-I32.7/P151.8; Circuit breaker not switched on)
(W290, (FG7-1) +S2-I3.1/P144.2; Machine not switched on)
(W0291, (FG7-2) +M-S96.0/P251.6; Safety door 1 of the workplace not locked)
(W292, (FG7-3) +M-S97.0/P252.2; Safety door 2 of the workplace not locked)

(Fault of the function group 11)
(E482, (FG11-1) Drive Lock X,Y,Z1,Z2-Axis; Spindle 1 HDK pressure achieved missing)
(E483, (FG11-2) Drive Lock X,Y,Z1,Z2-Axis; Spindle 2 HDK pressure achieved missing)
(E484, (FG11-3) Drive Lock X,Y,Z1,Z2-Axis; Spindle 1 speed achieved missing)
(E485, (FG11-4) Drive Lock X,Y,Z1,Z2-Axis; Spindle 2 speed achieved missing)

(Notes on the function group 10)
(M1159, (FG10-6) control reset (M25 channel 1) missing)
(M1160, (FG10-7) control reset (M25 channel 2) missing)
(M1161, (FG10-8) control reset (M25 channel 3) missing)
(M1162, (FG10-9) control reset (M25 channel 4) missing)
(M1163, (FG10-10) control reset (M25 channel 5) missing)
```



Modifications to MZA files go into effect by switching the active language or by restarting the system (soft reset).

PLC interface

The PLC interface to the MZA consists of a data structure that is configured within the PLC/CNC interface. You can choose between 2048 or 8096 messages for the width of the MZA interface.

Configuration of the PLC interface

The interface is executed in the hardware configuration of the PLC/CNC interface (CyclicProcessData). The MZA interface can be inserted by clicking the right mouse button on node "CyclicProcessData". Choose between "New qMZA" for 2048 messages and "New qMZA_Ext" for 8096 messages.

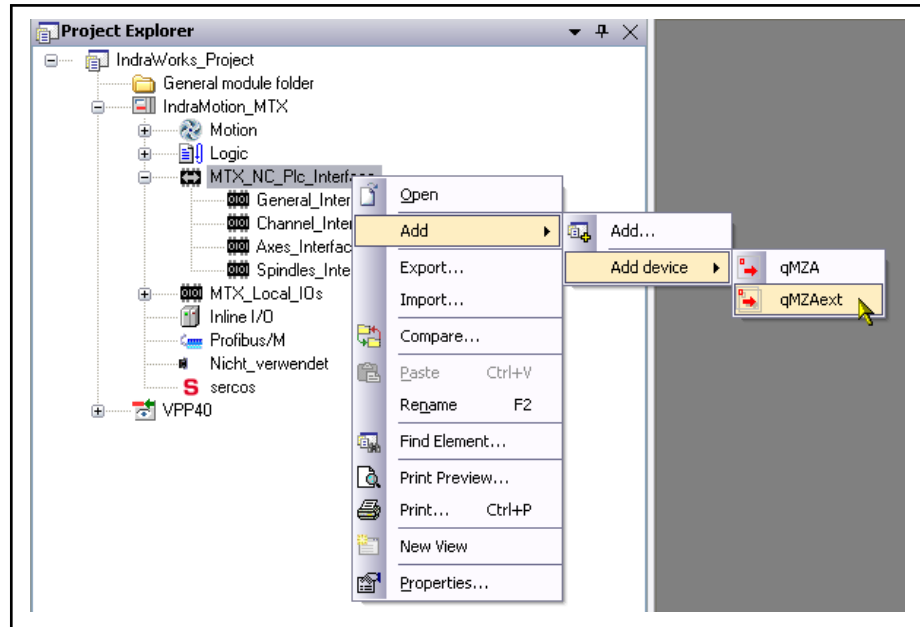


Fig. 7-21: Inserting the MZA interface

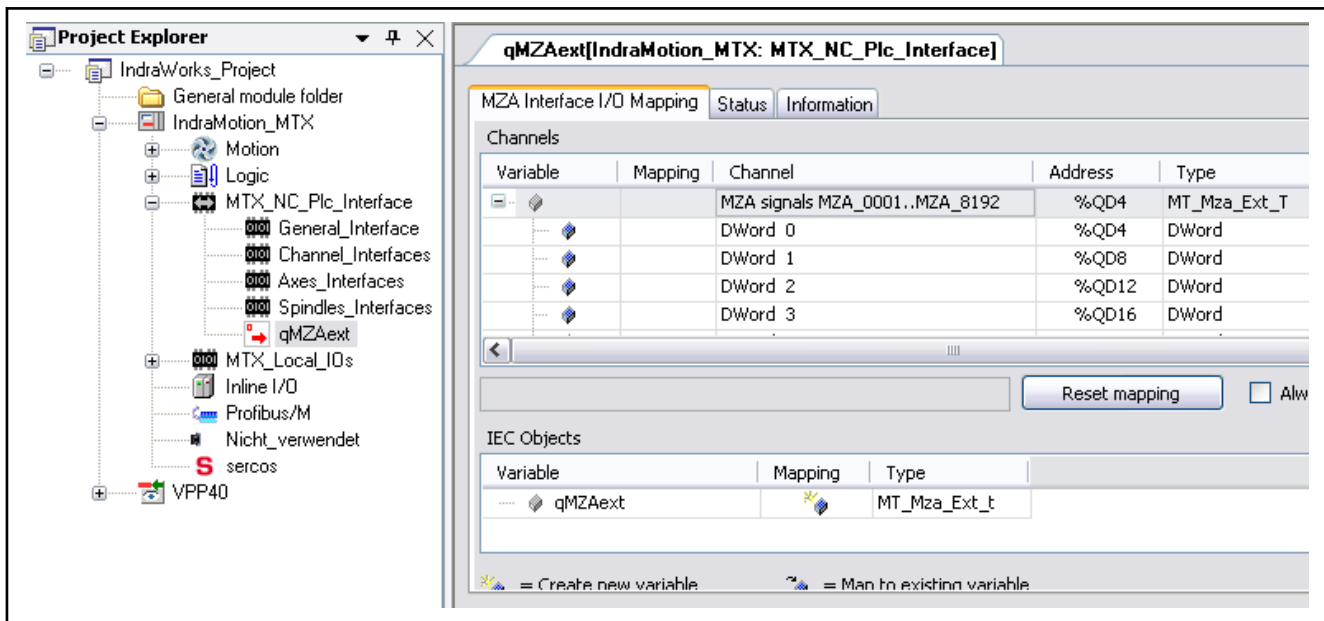


Fig. 7-22: Configuring the MZA interface

The symbolic name of the structure should be qMZA. The default address is at %QB1000.

Programming the PLC interface

General information

The "qMZA" data structure provides a separate bit for each message of the MZA. The bits are addressed using their symbolic names. The message is

shown in the diagnostics as long as the relevant bit for a message is TRUE. The time of the rising flank for the bit is entered in the logbook as "Message arrives" and the time of the falling flank is entered as "Message departs".

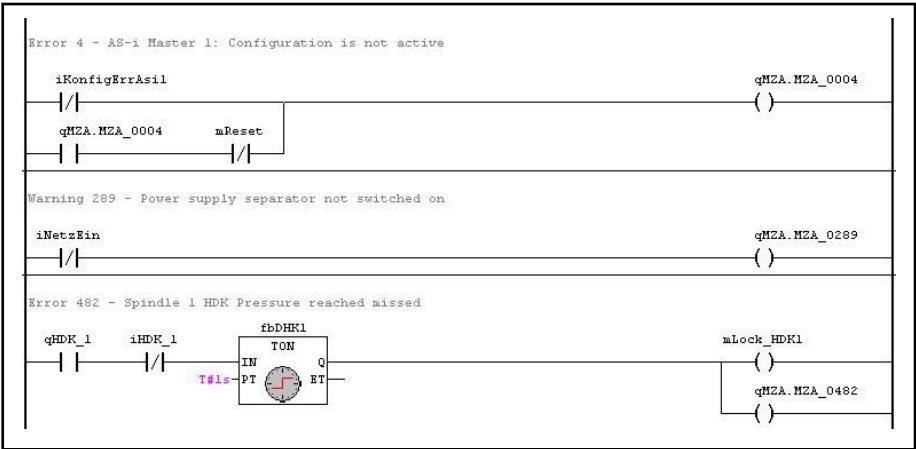


Fig. 7-23: Example of MZA programming

Handling instruction: MZA interface

This chapter describes the use of the MZA interface.

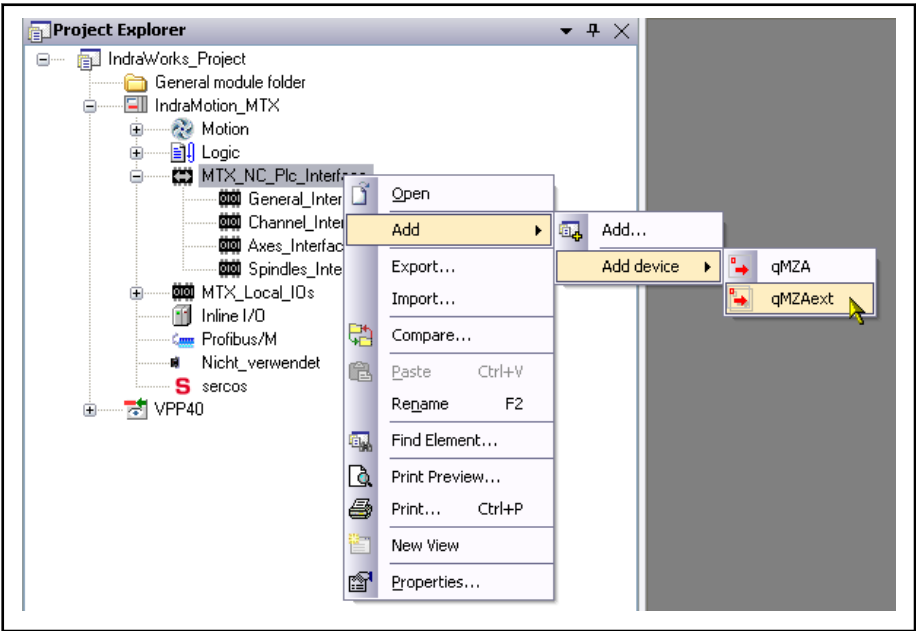


Fig. 7-24: Creating an MZA interface

IW Engineering / CyclicProcessData: Create a new MZA interface

1. Click with the right mouse button on the "CyclicProcessData" node.
2. Select the MZA interface.
Choose between "New qMZA" for 2048 messages and "New qMZA_Ext" for 8096 messages.

		Documentation chapter 7.8.1 "Machine fault and status display (MZA)" on page 163
Documentation:	IndraWorks Commissioning	MZA diagnostics

IW Engineering / CyclicProcessData: Delete an MZA interface

1. Right-click on the "qMZA" or on the "qMZA_Ext" node.
2. Left-click and select "Delete".

		Documentation chapter 7.8.1 "Machine fault and status display (MZA)" on page 163
Documentation:	IndraWorks Commissioning	MZA diagnostics

IW Engineering / CyclicProcessData: Make changes

1. Double-click on the node "qMZA" or "qMZA_Ext"

Additional information and configuration possibilities for the corresponding MZA interface are shown. This "Properties" dialog can change symbolic names, addresses and comments.

The modifications are applied after exiting the dialog.

		Documentation chapter 7.8.1 "Machine fault and status display (MZA)" on page 163
Documentation:	IndraWorks Commissioning	MZA diagnostics

Handling instruction: MZA Configuration

This chapter describes how to configure the MZA.

IW Engineering/configuration: Enable MZA

The MZA message system is configured using the parameter "Cycle Time". This parameter specifies the updating time of the message system in milliseconds. In general, an updating time of 500 ms is sufficient.

		Documentation chapter 7.8.1 "Machine fault and status display (MZA)" on page 163
Documentation:	IndraWorks Commissioning	MZA diagnostics

IW Engineering/configuration: Disable MZA

If 0 is entered under "**Cycle Time**", no MZA messages are displayed or entered in the logbook.

		Documentation chapter 7.8.1 "Machine fault and status display (MZA)" on page 163
Documentation:	IndraWorks Commissioning	MZA diagnostics

7.8.2 ProVi

Commissioning and programming ProVi messages

General information

ProVi messages are issued by the PLC. They can be displayed in the HMI interface. ProVi messages can also be logged in a logbook.

They can be grouped into five message types:

- Errors
- Note
- Warning
- Startup requirements
- Setup diagnostics

All message types can be grouped into different modules.

A fault category and a message group can also be assigned to each message. ProVi messages can be programmed to be set, i.e. the message is displayed until it is reset by calling a FB (function block). The message texts can be entered in multiple languages, directly in IndraLogic.

Handling instruction: Commissioning and programming ProVi messages

There is a separate documentation for ProVi which answers all questions in detail.

The chapter "First Steps" helps the user to become familiar with ProVi.

IW Engineering: Working with ProVi

		Documentation chapter 7.8.2 "ProVi" on page 171
Documentation:	PLC program development with Rexroth IndraLogic	ProVi Diagnostics - First steps

7.8.3 Diagnostic messages on small operator panels

General information

To implement the NC diagnostics for small operator panels of the VEP, VCP and VCH series, the diagnostics texts have to be transferred from the source path "...\\Rexroth\\IndraWorks\\mtx\\text" to control to provide them in the PLC. A download mechanism in the IndraWorks Engineering is used for the transfer.

Downloading language files

Calling the download dialog

The download dialog is started either via the context menu of a device or via the device-specific entries in the main menu. Therefore, select **Load Diagnostic Text in Control**

Control commissioning

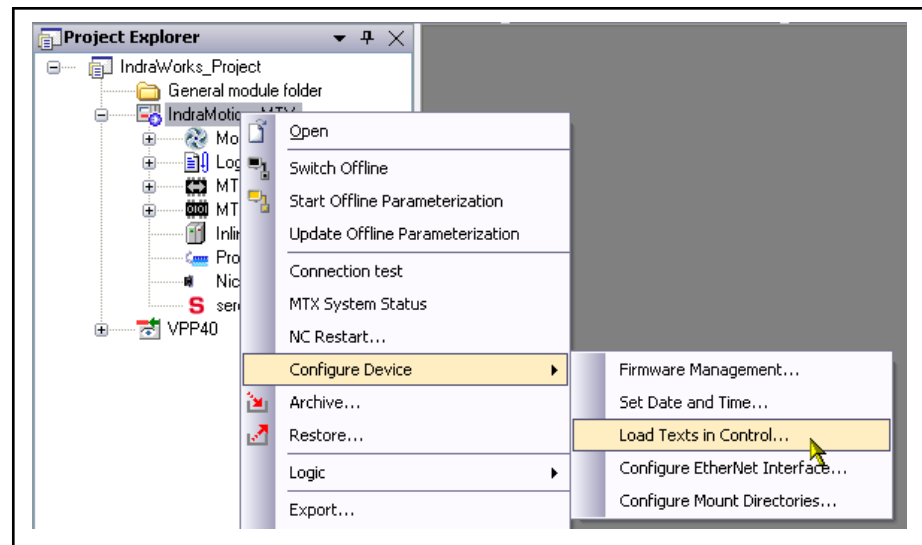


Fig. 7-25: Calling the download dialog

The download dialog provides the following functionalities:

- Loading diagnostic texts
- Displaying available diagnostic texts
- Displaying the diagnostics texts currently located in the control
- Deleting diagnostic texts in the control

Describing the dialog elements

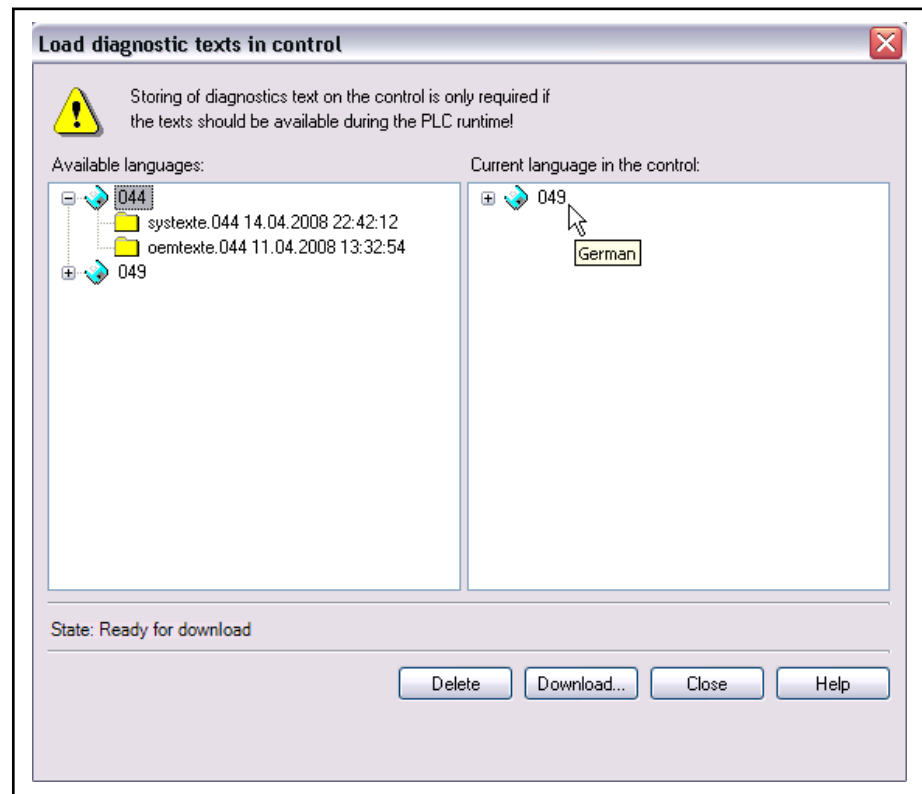


Fig. 7-26: Dialog "Load Diagnostic Text in Control"

The left window of the dialog shows the languages of the diagnostic texts available as country code for a download (a tool tip shows the language as plain text). If you open the language version tree, the corresponding language files are shown with time stamp.

The language versions in the control are shown in the right window. By comparing times stamps, as possible required update can be identified.

Downloading single files is not possible. The language selected is always completely loaded with all its files available in the installation directory. Additionally, index files providing a faster access on the language files are created in the control. The progress bars informs on the download status.

Storage location of the language and index files in the control is the "**root/usrfep/text**" directory. If any files exist, they are overwritten.

The "Delete" function removes all language and index files for the selected language version in the control.



If the are desired texts in the control, the **MT_DiagText** function block can be used in the PLC later on to provide an image of the NC error and the message database. Further information, the description of the function block as well as a configuration example can be found in the documentation "PLC Interface".

7.9 Setting up NC cycles

7.9.1 Input masks

To facilitate the programming of cycle calls, the input support of the editor is often used. Input masks have to be defined for cycles requiring this support. This is for the interface configuration and is not related to the function of cycles during program processing.

For how to create input masks, refer to [chapter "NC cycle input mask" on page 96](#).

7.9.2 Parameter settings for cycle calls

The non-modal standard cycles have already been entered. Non-modal user cycles can be added here.

The modal standard cycles have already been entered. Modal user cycles can be added here.

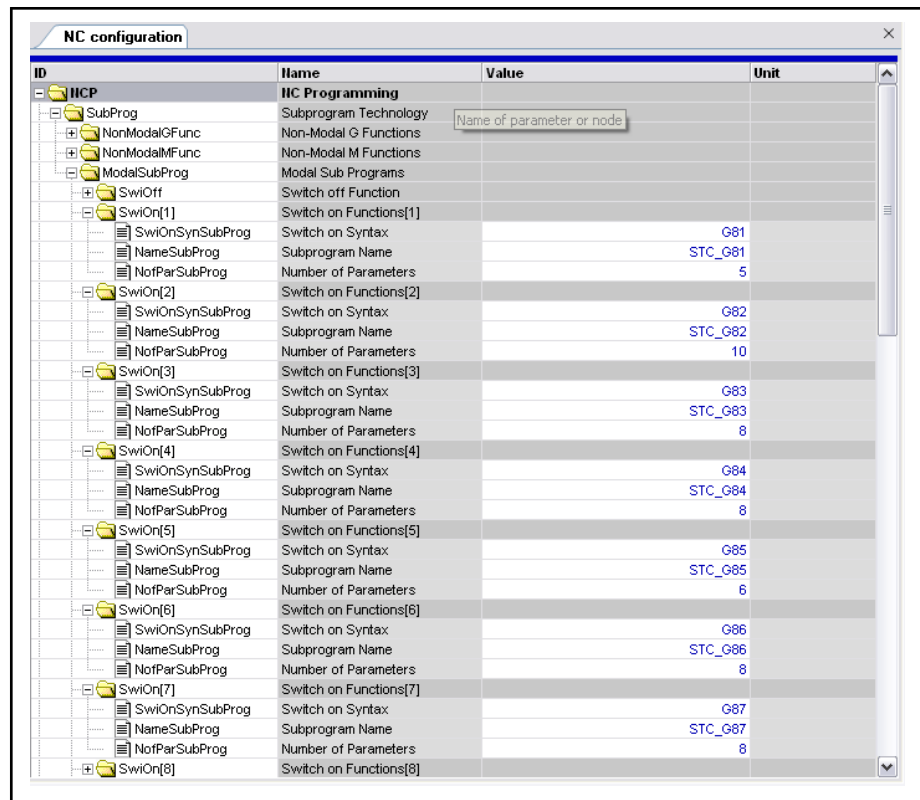


Fig. 7-27: Editing IndraWorks Engineering parameters

7.9.3 Subroutines

The subroutines (cycles) of the machine manufacturer are saved in the control file system under "root\usr\mtb\cycles".

The subroutines (cycles) of the end user are saved in the file system of the control under "root\usr\user\cycles".

7.9.4 SD variables

Some cycles can optionally operate with permanent channel-dependent SD variables. The assignment is listed in the respective cycle description. The SD variables used for the standard cycles are available. If user cycles are to be created and implemented this way, the SD variables used have to be defined.

7.9.5 Using existing projects

This section relates only to projects created with versions < MTX09V06.

There is not complete compatibility. Therefore, the user has to decide whether to continue using the existing projects in their "old" form or whether to work with standard cycles in future. **A combination of these two variants should not be used!**

The usage of individual cycles from an old version does not require any further measures. However, it is not possible to access the installed standard cycles.

The following modifications are required to use the standard cycles in existing projects (project version ≥ MTC09V06):

- Adapt the search path for the cycles in the machine parameters. "NC optimization (NCO) -- FileOrg -- SrchPathSubProg". Add a new entry "/feprom/cycles" in front of the entry "/feprom".

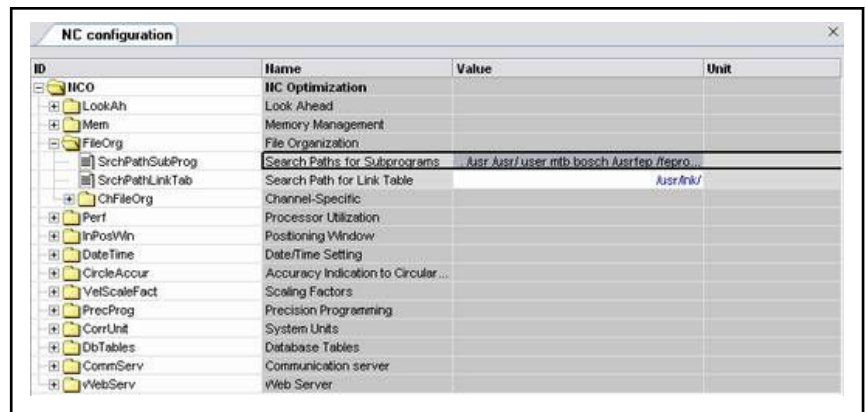


Fig. 7-28: Extending the search path

- Make standard cycle entries in the machine parameters. "NC Programming(NCP) -- SubProg -- ModalSubProg -- SwiOn[n]". Refer to [chapter 7.9.2 "Parameter settings for cycle calls" on page 173](#). The required entries can be taken from the default values.

If, after conversion to the standard cycles, the existing user cycles have to be used as well, these should either be adapted to the addressed notation or their names should be changed in such a way that they match the intended range for user cycles (G390 - G499). In doing so, cycle subroutines, cycle headers and definitions in the machine data also have to be adapted. It should also be noted that the cycle calls in the NC programs also have to be adapted to these changes.

7.10 Configuring the tool management

7.10.1 Basics

Tool corrections

The tool data management is of highest importance for controlling machine tools.

The tool data comprises, for example, data on:

- Tool identification
- Description of tool geometry (tool correction data)
- Tool life management
- Description of tool location (place)
- Description of tool type
- Status information
- etc.

In addition to these so-called external tool corrections, D-corrections can be used.

The following figure explains the mode of operation of external tool corrections and D-corrections.

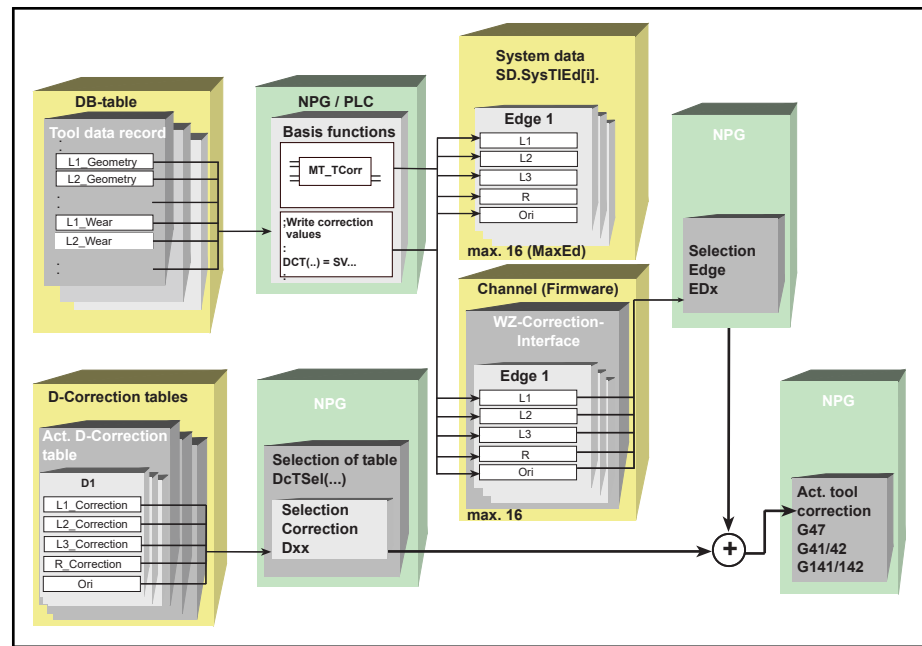


Fig. 7-29: Principle of D-corrections and external Tool corrections

The block preparation of the MTX takes the tool corrections stored in the tool correction interface into account. The tool correction interface can be written via NC (via CPL) or PLC. Within this interface, corrections for up to 16 tool edges can be stored. A correction block includes the correction values L1, L2, L3, R as well as the edge position Ori and orientation angles Phi, Theta and Psi. Use the orientation angle to position the tool at an inclined angle in space. G47 activates the MTX tool length correction and G48 deactivates it. Use ED (cyclic duration) to switch the edge or the correction data block within the tool correction interface.

The tool length corrections L1, L2 and L3 to the axes or coordinates to which the length corrections are to be applied are assigned via the configuration data. Irrespective of this setting, the effect of the tool length corrections can be changed at runtime via G47. G47 simultaneously activates the tool correction for the latest preselected edge using ED.

Data management

The data management of the external tool corrections is managed in a database. The database is subdivided into database tables. Two database tables are available. By default, database table 1 is used to manage data of the external tool corrections. Accordingly, database table 2 is available to manage pallet data for example.

The following figure provides an overview on the database structures and its exemplary use:

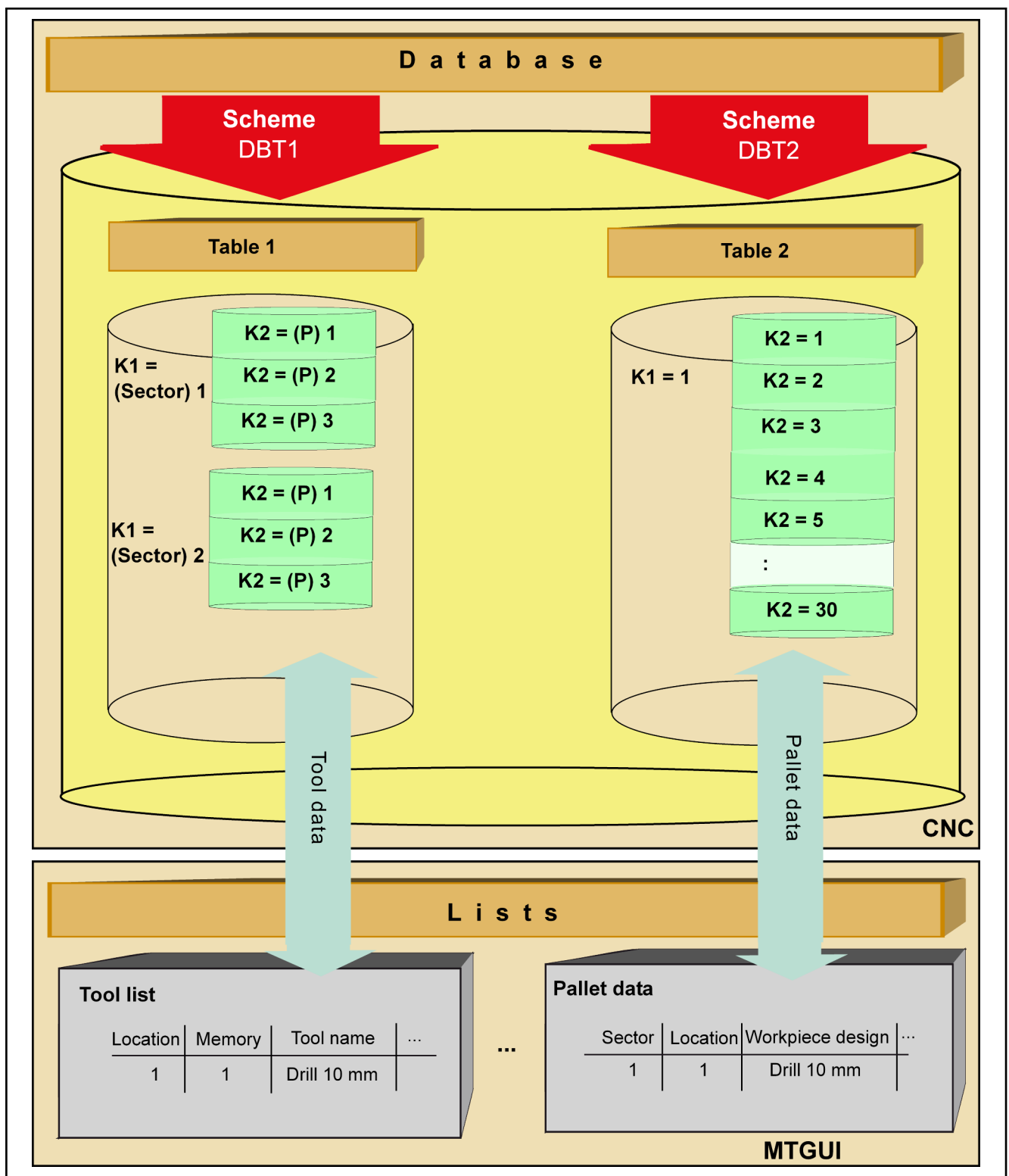


Fig. 7-30: Database structure

The tool database can be compared to a container structurally storing tool data.

The database consists of individual data blocks. Their structure can be configured user-specifically.

Array structure of a data block:

Array/ element	Identifier	Type	Description (for DBT1)	Comment (for DBT1)
1	K1	Integer	Sector	DS key
2	K2	Integer	Place	
3	SKQ	String	ID	Tool identification
4	IKQ1	Integer	Duplo no.	
5	IKQ2	Integer	Type	Type identification
6	IKQ3	Integer	T. No.	Tool identification
7	IQ1	Integer	Reserve	-
8	IQ2	Integer	Reserve	
9	IQ3	Integer	Reserve	
10	BQ1	Bit array (32)	P-status	Place status
11	BQ2	Bit array (32)	T-status	Tool status
12	BQ3	Bit array (32)	Technology	Type identification
13	aaa	1. freely configurable data element		
:	:	:		
n	zzz	nth freely configurable data element		

Tab. 7-5: Data block structure

A data block is uniquely identified by the two arrays "Storage"/"K1" and "Place"/"K2".

For a tool database, a single data block always corresponds to a place which can accept a tool. This can be specific places in the machine (e.g. spindle, gripper, charging place, discharging place) or in a magazine.

When tools are engaged into places or transferred to other places, relevant tool data in the corresponding data blocks is copied.

The number of data blocks available after re-initializing the database can be configured as well as the display of tool data in tool lists and tool editors.



The size of a data block is limited to 4 kB.

Communication with the PLC

Different tool management functions can communicate with the PLC.

There is the option to inform the PLC on changes in the database such as "Move tool" and to wait for a PLC reaction and to respond to it.

The function can also be transferred directly to the PLC for execution purposes. The operating interface provides only parameters such as the source or target place number.

In the first case, the data element is provided with a respective data attribute when defined (e.g. PLCCheck, PLCInfo or PLCWrite). The interface variables are defined in the file "ToolManagementConfig.xml". Go to the menu item **Properties...** on the node "DBT1 screens (tool management)" in the project tree in IndraWorks Engineering.

For a further description on the configuration steps, refer to ["Checking input value via PLC" on page 227](#).

In the second case, a key function is used to start the function and the interface variables are defined in the F- or M-key configurator.

For a further description on the configuration steps, refer to [chapter "F-key panel definition" on page 264](#).

7.10.2 General configuration tools

General information

To configure tool data management, follow the configuration steps described below in the specified sequence:

1. Specification of the database size or the number of data blocks ([chapter 7.10.4 "Configuring the database" on page 213](#))
2. Definition of the database structure (sector/place) ([chapter "Defining sector and place distribution of database table" on page 213](#))
3. Adjustment of the database schema ([chapter "Configuring data blocks" on page 214](#))
4. Optional extension of the tool catalog ([chapter 7.10.5 "Tool catalog" on page 242](#))
5. Configuration of the user interface ([chapter 7.10.6 "User interface" on page 259](#))

Definition of terms

ULC "ULC" stands for "Universal List Control", which is the central element of tool list and container configuration.

Sublist "Sublist" is a central term in the schema definition for the ULC. A ULC is a table editor which may be able to multiply the configured line-column definition for the presentation of a (tool) data block according to the number of data blocks. In this case, a sublist stands for the presentation of a data block. Therefore, only one sublist needs to be defined for a list of several tool data blocks.

This principle can also be applied to certain parts of the (tool) data block (e.g. tool edge data, etc.).

Schema editor

General information

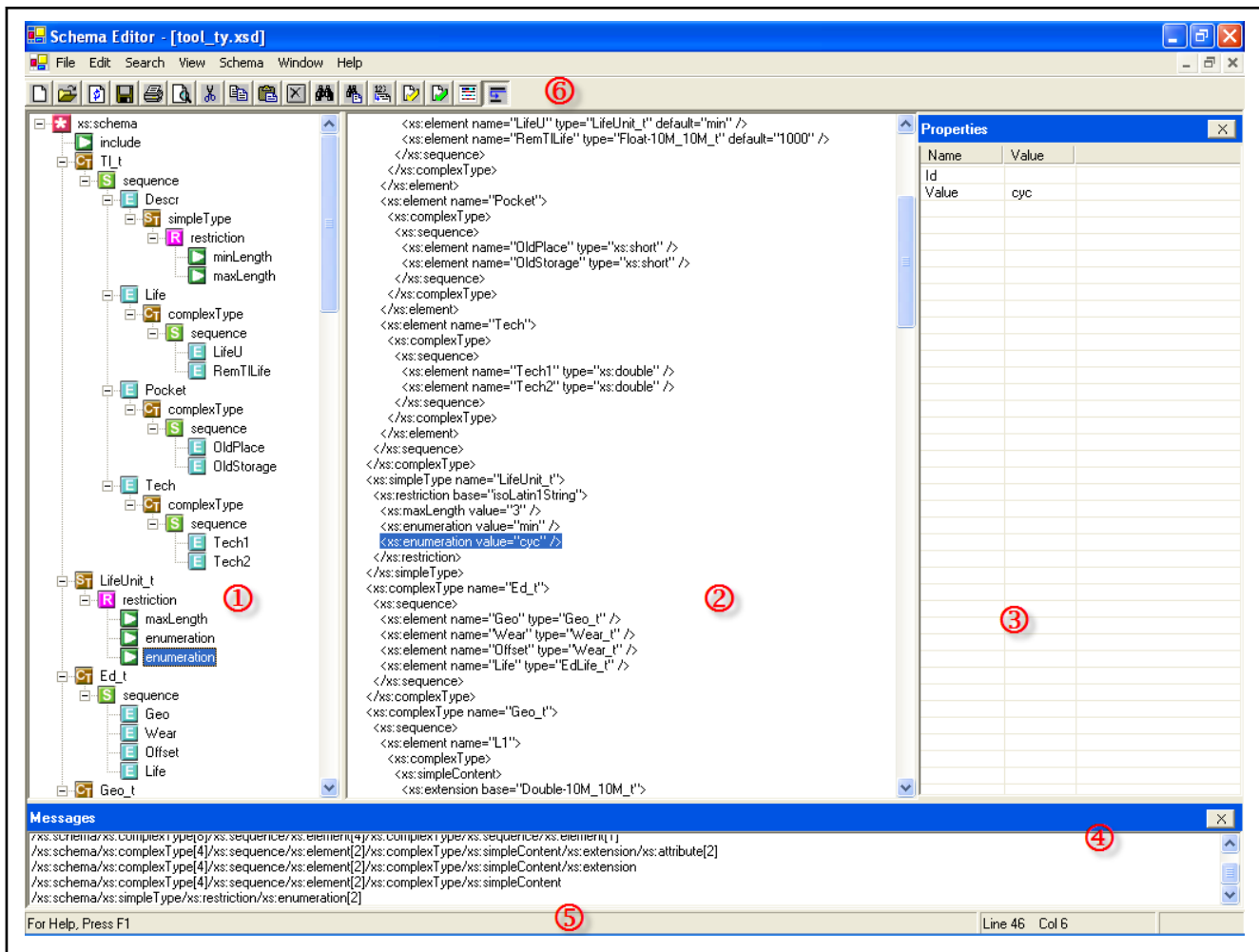
A schema editor is provided especially so that the user can see and change data block schemas.

In addition to the possibilities of schema validation and of the "style" test, this schema editor permits schema files to be clearly displayed and modified within various windows.

The figure below shows the various areas of the schema editor:

- Tree display (always present)
- Text display (always present)
- Properties window (can be optionally hidden)
- Message window (can be optionally hidden)
- Status and toolbar (can be optionally hidden)

Control commissioning



- 1 Tree representation
- 2 Text display
- 3 Properties window
- 4 Message window
- 5 Status bar
- 6 Toolbar

Fig. 7-31: MTX schema editor

Graphical Element Description



Root node in .xsd files



Simple type



Complex type



Element



Attributes



AnyAttribute



AttributeGroup



All

Graphical Element	Description
	Any
	Choice
	Group
	List
	Restriction
	Sequence
	Annotation
	Documentation
	Union
	<i>General symbol of:</i> • minLength • maxLength • include • #text • simple Content • extension • enumeration • ...

Editing in the tree view

- **Delete:**
Select the node and use the <DEL> key or right click on **Delete**
- **Paste:**
Right mouse key, popup **Add Child**. This menu item provides a selection of all schema elements that can be pasted under the currently selected element.

The pasted elements are immediately visible in the Text Editor. In order to change the elements, the Properties windows or the Text Editor can now be used.

Editing in the Properties window

The properties listed in this window always belong to the node selected in the tree representation. Different properties are possible from node to node. If only a limited range exists for the value of a property, this range is provided in combo boxes.

After the node has been changed, the other two views (tree and text) are updated.

Editing in the text editor

Here, anything can be entered without any limitations. By means of the button <F5> "Synchronize Tree", the tree is updated. Any errors are detected during the update and are shown in the "Messages" window. Furthermore,

you can check your text for "Good style" <F7> and for compliance with the schema conventions <F8>.

Menu items

File menu	New <Ctrl>+<N>:	Create new file; A selection can be made between XML and schema and between various Unicode character sets
	Open <Ctrl>+<O>:	Open an existing file
	Close:	Close the currently open file
	Close all:	Close all open files
	Save <Ctrl>+<S>:	Save the current file
	Save as:	Save the current file under a different name
	Print <Ctrl>+<P>:	Directly print the current file
	Print preview:	Display a print preview
	Print setup:	Set up the page to be printed
	Recent files:	List of the files opened last
	Exit:	Exit the schema editor
Edit Menu	Cut <Ctrl>+<X>:	Cut the highlighted characters in the text editor
	Copy <Ctrl>+<C>:	Copy the highlighted characters in the text editor
	Paste <Ctrl>+<V>:	Paste the highlighted characters in the text editor
	Delete text:	Delete the highlighted characters in the text editor
	Delete node:	Delete the highlighted node in the tree view
	Delete messages:	Delete the messages in the messages window
	Select all <Ctrl>+<A>:	If the text editor is active, mark the entire text from top to bottom.
	Go to line/char <Ctrl>+<G>:	Open an input window to enter the line and character number and to focus the specified character in the text editor.
Search menu	Find <Ctrl>+<F>:	Open the "find" window
	Replace <Ctrl>+<Shift>+<H>:	Open the "find and replace" window
View menu	Highlight text <Ctrl>+<H>:	Specifically switch "highlight key words" on or off. Documents up to a certain size are highlighted when opened. For performance reasons, larger files are not highlighted when opened
	WordWrap:	Switch the automatic line break on or off
	Expand tree view <Ctrl>+<Shift>+<E>:	Collapse/expand the tree and all the nodes
	Collapse tree view <Ctrl>+<Shift>+<C>:	Collapse the tree and all the nodes

	Schema view <Ctrl>+<Shift>+<S>:	Switch on or off schema view. The schema view hides certain structure nodes, e.g. the nodes of the type "simpleContent" and "complexContent" are hidden
	Synchronize tree <F5>:	Scan text and update the tree view. Scanning errors are shown in the message box
	Schema <F4>:	The schema that describes the permitted elements of the schema that is currently being processed is displayed
	Customize...:	Show/hide toolbar buttons
	Properties <Ctrl>+<Shift>+<P>:	Show/hide properties window
	Messages <Ctrl>+<Shift>+<P>:	Show/hide messages window
	Tool bar <Ctrl>+<Shift>+<P>:	Show/hide toolbar
	Status bar <Ctrl>+<Shift>+<P>:	Show/hide status bar
Schema menu	Check style <F7>:	Checks the "style" of the document. For example, missing parentheses are found with this function
	Validate <F8>:	Checks the document against the schema on which it is based
Window menu	Cascade:	Cascade the windows
	Tile horizontally:	Arrange the windows under one another
	Tile vertically:	Arrange the windows next to one another
Help menu	Help topics <F1>:	Show the "Help" dialog
	About the schema editor:	

ULC configurator

General information

The ULC configurator has 2 types of visual presentation which, however, use the same database (configuration file), i.e. Changes in one of the configurators have the same effect on the other configurator and are also displayed.

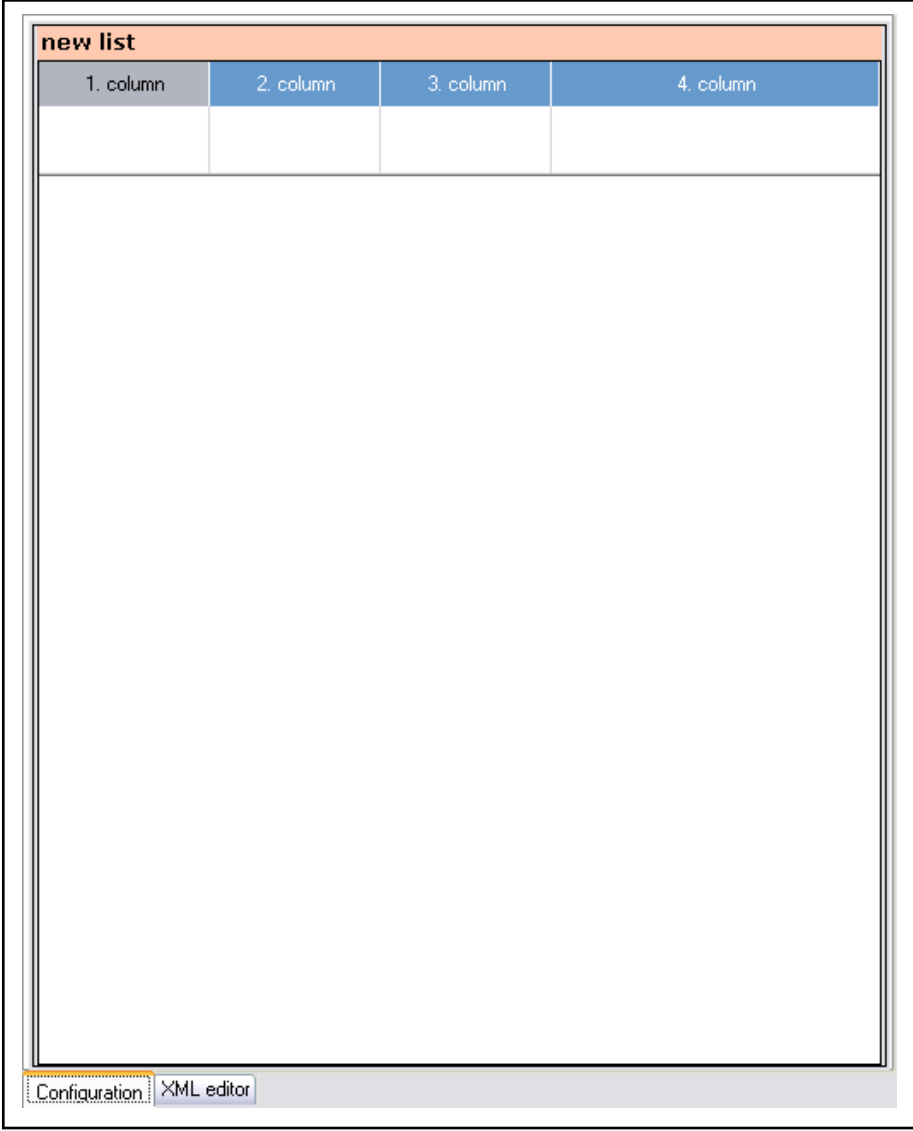
1. Configurator for common configuration steps.
2. XML editor for special settings which cannot be selected by means of the configurator (see [chapter "XML file editor" on page 201](#)).

This chapter especially deals with the ULC configurator.

Opening or creating a configuration

Opening an existing configuration	The ULC configurator is either called by the list configurator via the configuration key  or the function "Open" on the relevant entry in the branch "Form configuration".
Creating a new configuration	Using the function New under the node List controls or Container controls , either a new list configuration or a new container configuration can be created for the tool editor (see fig. 7-33 "Template for a new container configuration" on page 185). For this purpose, the relevant basic configuration is referred to for initial definition. Another option for creating a new configuration file is to create a configuration which is as similar as possible to basic configuration

using the function **Duplicate (copy & paste)**. After having created such a configuration, load it into the editor as described above.



The image shows a software interface for creating a new list configuration. It features a window titled "new list" with a header bar. Below the header is a table with four columns labeled "1. column", "2. column", "3. column", and "4. column". The first row of the table is highlighted in blue. Below the table is a large, empty rectangular area for editing. At the bottom of the window, there are two tabs: "Configuration" (which is active) and "XML editor".

1. column	2. column	3. column	4. column

Fig. 7-32: Template for a new list configuration

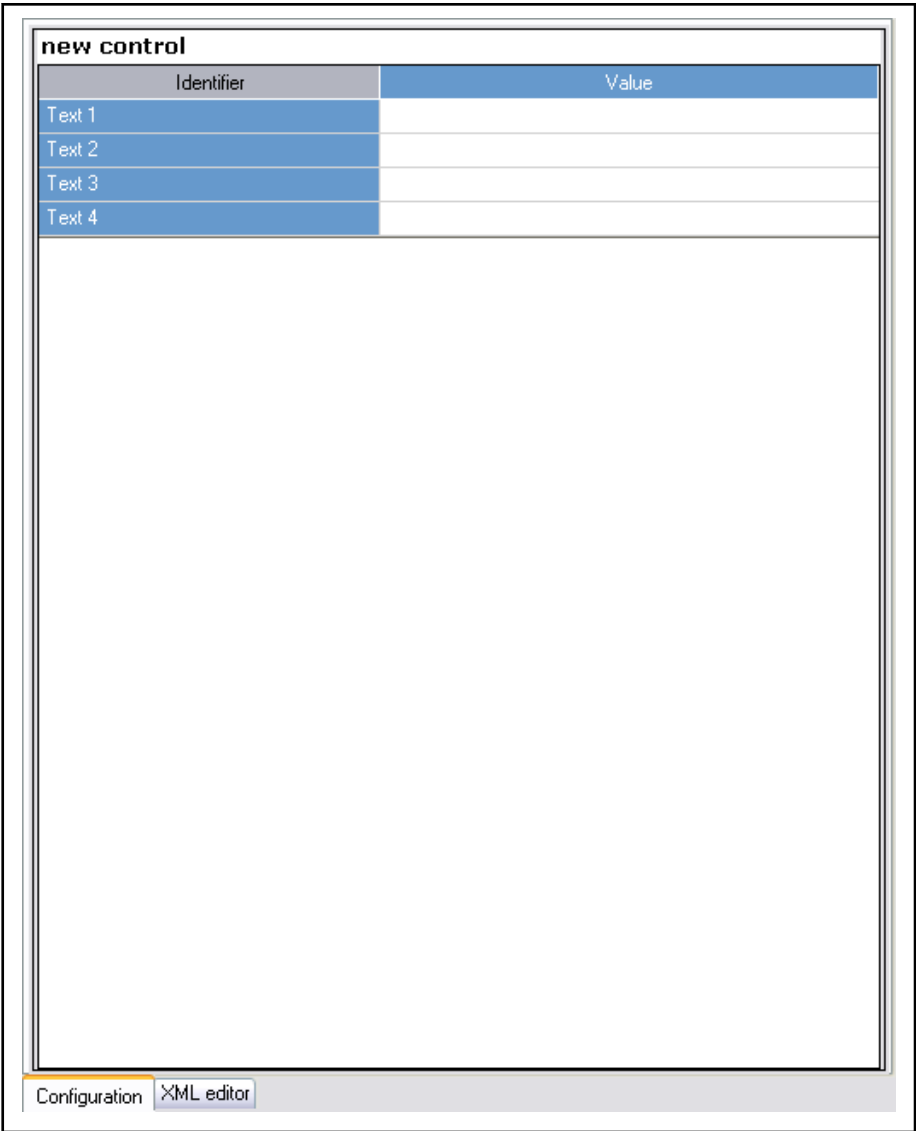


Fig. 7-33: Template for a new container configuration

Saving the changed configuration

If the configuration has been changed, the user will be asked whether the changes are to be saved or not when exiting the configuration dialog. In addition, see the current status any time by retrieving the function **Save** <Ctrl> +<S> in the pull-down menu.

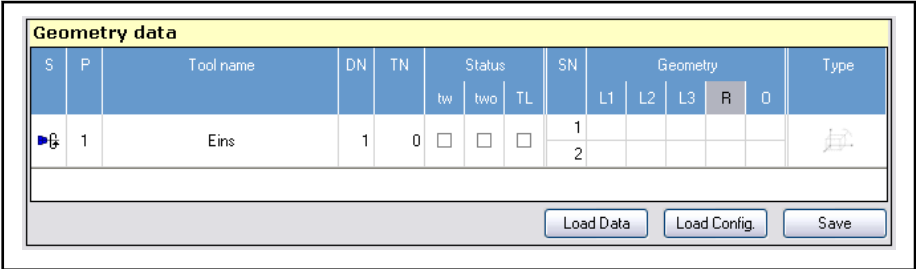


Fig. 7-34: ULC preview with function buttons

Creating the desired number of lines and columns

To obtain the desired number of lines and columns, the initial configuration must be modified. To do this, call the corresponding commands using the right mouse button and the pop-up menu. For example, to insert a column at

the front, use the right mouse button to click the first column and execute the command **Insert ► InsertColumn ► Before**.

The following commands exist:

- Insert
 - Insert column
 - Before
 - Behind
 - Insert subcolumn
 - Before
 - Behind
 - Insert row
 - Above
 - Below
- Delete
 - Column
 - Subcolumn
 - Row

In contrast to the command "Insert Column", the command "Insert Subcolumn" does not generate a complete column; instead, it generates a subcolumn that shares the topmost heading with the initial column.

Opening the cell editor

The cell editor is opened by double-clicking a cell or via the pop-up menu. When the editor is open, click another cell using the mouse and edit its data. The data of the cell that was edited previously is saved in the clipboard. The modified data block is added to the preview by pressing "Apply" or "OK".

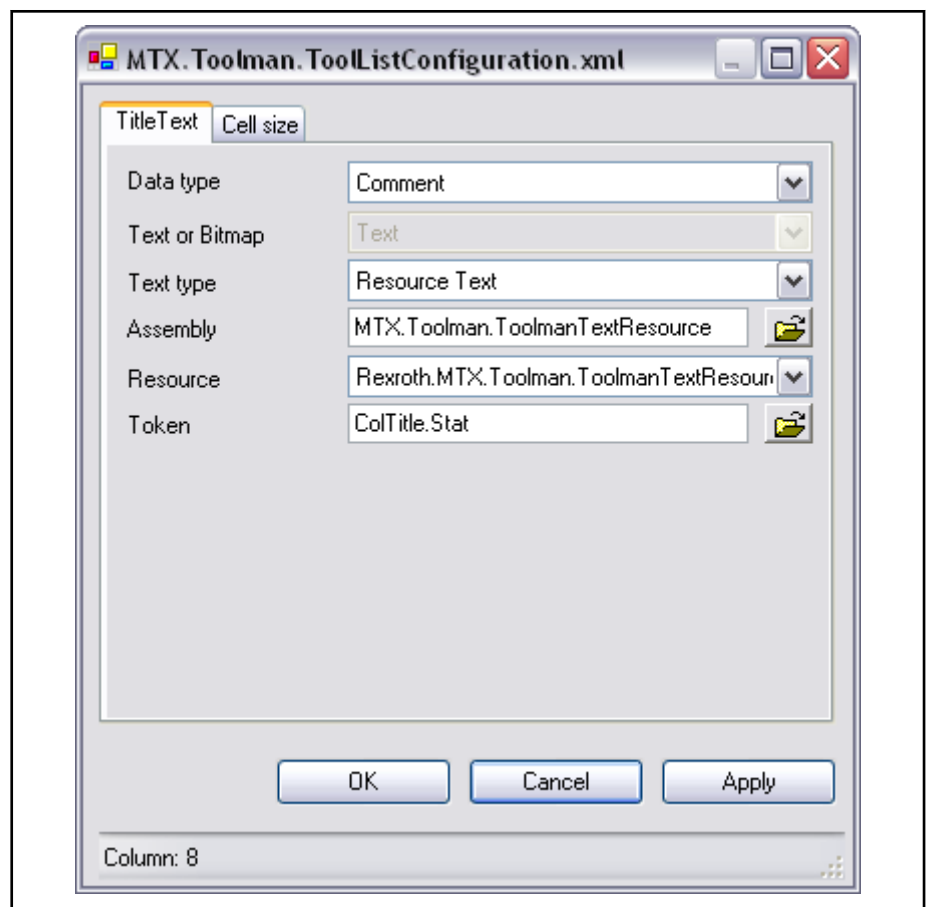


Fig. 7-35: Dialog: MTX.Toolman (editor for cells, columns, titles and global list properties)

Setting the width and height of rows and columns

The width and height can be set in two different ways. To select all settings using the mouse, enable the change mode using a right mouse-click in the popup menu **Edit column width / row height**. Now the width and height of every line and column can be modified using the mouse.

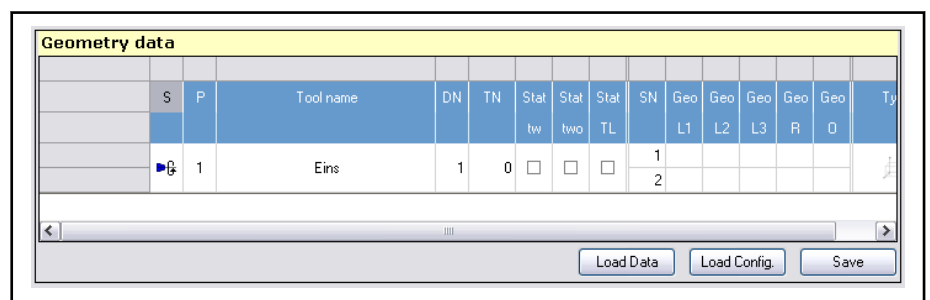


Fig. 7-36: The row height and column width can be modified directly using the mouse in this preview

The second method is using the editor. Open the editor as described above. Ensure that the cursor is not positioned over the title row, but that the row/column to be modified is selected. Open the second tab page "CellSize" and enter the desired values in the fields Height/Width.

Creating contents of individual cells

Logic of cell contents

In order to edit the content of individual cells, open the editor again, as described above. Ensure that only the cell whose data is to be modified is selected.

If you press "Apply", the modified data is applied to the preview, but are not permanently saved - instead, it is saved only in RAM and applied to the preview.

In general, a cell can have three states:

1. It can be empty, this means "data type = **empty cell**".
Nothing else can be set in this case.
2. The cell can contain a comment (**comment**).
3. The cell can reflect the value of a process variable (**process variable**).

Empty cell

The cell is empty; it does not display a comment or a process value. Nothing else can be set here.

Comment

The cell displays a fixed comment. This comment can consist of a bitmap or text. In turn, the text can be permanently entered or can be language-dependent. Language-dependent texts can originate from a resource file or can be selected from a CSV file. CSV files can be edited by the user in this dialog.

If the cell is defined as a comment cell, there are several input methods:

Text or bitmap

There are 3 methods here:

Text	Only text is displayed
Bitmap	Only a bitmap is displayed
Text and bitmap	Both text and a bitmap can be specified and selected. The corresponding style can be used to specify how the text and the bitmap are displayed, i.e. on top of or next to one another. Currently, the data of the style must still be modified directly in the configuration file using an XML editor

Text type

The following three selection possibilities exist:

User text	This reads out the CSV file mentioned above. If a user text is used, the elements "list name" and "token" need to be specified in addition
Text resource	The comment to be displayed is read from a resource file with text. File name and text name must be specified
Fixed comment	In this case, the text can be entered directly in field "comment"

List name

This field is available only if "User text" has been selected for the text type. Either a list can be selected or a new one can be generated with "New".



⇒ Although this field can be adjusted in every cell, it is global and thus always changes the content for every field!

Token (User Text)

A token can be entered directly. However, it is a good idea to select the token from the list. To do this, press the "Open" button. Now a box with the list of all texts that have already been defined opens. This list can be used to not only

select texts, but also to generate new ones. A description of this list can be found in "Edit Text List".

Assembly	The field "Assembly" is provided for the text type "Resource Text". The content of this field can be selected using a file dialog that opens when you press "Open" on to the right of the input field.
Resource	If you have selected an assembly, all the resources in this assembly are contained in the Resource combo box, where they can be selected.
Token (Resource)	If a resource that contains the texts has been selected, a token can be selected from the contained token/text pairs. Start the text list using the "Open" button. The description of this list can be found in "Edit Text List".
Comment	This field is available for the text type "Fixed comment". The entered text here, is entered later in the cell.
Assembly of image	If Bitmap was selected in field "Text or Bitmap", the fields for selecting a bitmap are accessible. As has already been the case for the text assembly, a file dialog is used in "Assembly of image" to select an assembly that contains a resource with images.
Image resource	If an assembly has been selected, this combo box contains the resources of this assembly so that you can select the resource.
Image name	If the selected resource contains images, they can now be selected using the text list. This text list contains the names of all the images. The description of this list can be found in "Edit Text List".

Process variable

The cell displays a process-dependent value. This value is addressed using an XPath expression. The value can be displayed as text and/or as a bitmap. It can be formatted as text and be edited in the cell. The following entries can be selected:

Process variable	The XPath for the process variables is entered here.
Text or bitmap	There are three possibilities:

Text	Only text is displayed
Bitmap	Only a bitmap is displayed
Text and bitmap	Both text and a bitmap can be specified and selected. The corresponding style can be used to specify how the text and the bitmap are displayed, i.e. on top of or next to one another. Currently, the data of the style must still be modified directly in the configuration file using an XML editor

Text format	Numerical	The text is displayed as a number that can be formatted
	Text	The text is displayed as string. Using this setting, the process value is displayed directly without any formatting

Numerical type	If the format "Numerical" has been selected for the text, the number can be specified in more detail now. The "numerical type" is used to define how the number is to be displayed:

Integer	The number is displayed as integer
Float	A digit with number of decimals is displayed
Bit	Checkboxes are displayed
Use format string	The representation of the number is specified in more detail in the field "format string"

Control commissioning

Editing mask	The field "EditMask" is available for the numerical types Integer, Float and "Use format string". This field can be used to specify which form the user's entry must satisfy. Fixed characters can be assigned to the Edit field of the cell. The syntax is described in chapter "Syntax of editing mask" on page 198 .	
Digits	Digits can only be set if "Numerical type" = "Integer". The number entered here indicates the minimum number of digits used to display the number. Missing digits are supplemented by leading zeros.	
ProcVar Type	<i>The following types can be selected:</i> • System.Double • System.Single • System.Decimal • System.Integer	
Decimal places	The minimum number of positions after the decimal point is specified here. Missing digits are filled by zeros. This field exists only if "Numerical type" = "Float".	
Bit no. (0..32)	This field is visible only if "Numerical type" = "Bit". Here, the bit is defined by an integer whose value you want to display as a checkbox.	
Format string	This field is available only if "Numerical type" = "Used format string". The entry in this field determines how the number is displayed in the cell. The syntax is as follows:	
	C or c	Display as a localized currency
	E or e	Display as an exponential number
	F or f	Specified number of positions after the decimal point
	N or n	Like F, but with separator symbols for thousands
	P or p	Number is multiplied by 100 and displayed as a percentage
	0	Placeholder for numeral. If no numeral is located at this position, one is inserted
	#	Placeholder for numeral. A digit is displayed at this place. If there is no digit, it is not filled with a 0
	.	The character "." indicates the position of the decimal separator symbol
Editable	There are 4 possible values for Editable (EditStatus):	
	0 Yes	Cell can be edited
	1 No	Cell cannot be edited
	2 Call-back of the application	The call-back mechanism is used to query the calling application whether the currently selected cell can be edited
	3 Depends on process variables	Important for conditional editability
Edit type	There are 3 options for the edit type (EditTypeSelection):	

Text editor	An input field is available in the edit mode
Enum combo box	If a cell is focused, it becomes a combo box whose entries must be available in the schema for the data element to be defined
User combo box	When the cell is focused, it becomes a combo box. The entries in the combo box list are queried by the embedded application using call-back

Copying cell definitions

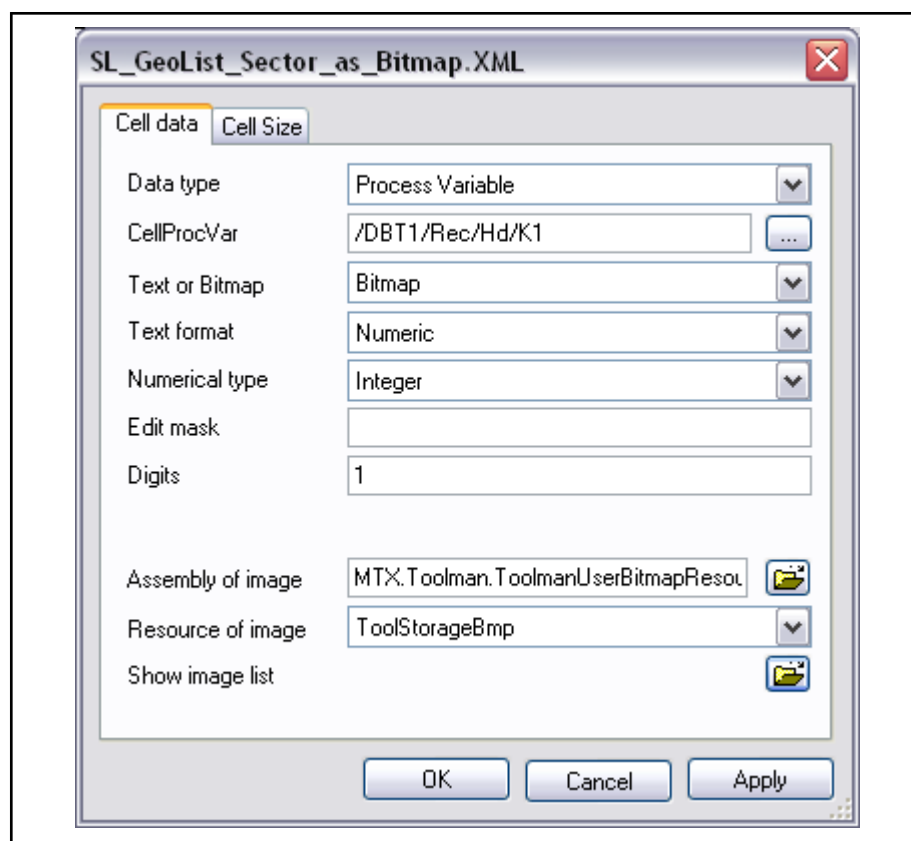
Complete cell definitions can be transferred from a selected cell to another cell by means of the function **"Copy cell data"** and/or **"Paste cell data"**.

Merging cells

Vertical cell rows can be merged by highlighting them and summarizing them in one cell system using the function **Merge Cells** to be called via the pop-up menu. Use the **Split cells** function to split a highlighted cell link.

Creating process-dependent bitmap selection

Screens can also be shown subject to process variables. To this effect, a list with process values and the corresponding screen is being defined. In the configurator, there is an own configuration dialog for the definition of the bitmap list. When selecting the display mode "Bitmap", select the dll and the resource to choose screens from first. Afterwards, define a "key - value - pair" using the dialog below. In this case, the value corresponds to the value of the variable which is linked to this cell. The value is the name of the screen from the resource. You can also indicate whether the value has to match the table value exactly or whether it has to be lower or greater.



Assembly of the bitmaps Dialogue for selection of the DLL with the screens
Image resource Combo box with all resources, the dll
Show screen list Opens the dialog with the list of the values and the screens

Fig. 7-37: Configuration dialog for creating the bitmap list



List

The value pairs of process value and screen name
Furthermore the screen itself is shown

Selection of radio button

Here it is determined how the process value has to be compared with the value from the list. The images are only displayed in the image above if the process value is equal to one of the values specified.

Add

The list will be extended by one entry at the end of the list

Delete

The selected entry is deleted from the list

Image

Another dialog opens in order to select a screen from the list of all screens contained in the resource

Ok

List is applied

Cancellation

Settings are discarded

Fig. 7-38:

Dialog: List of keys/value pairs



Fig. 7-39: List of the screens in the selected resource

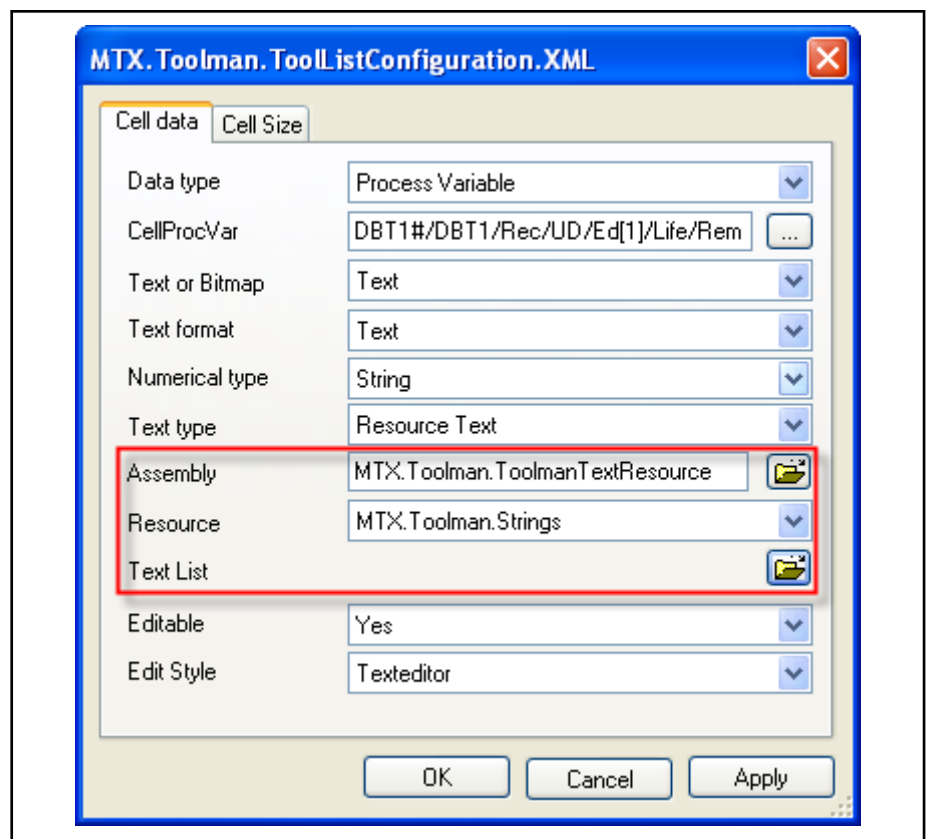
In order to select a screen, place the cursor on the desired screen and click on "OK".

Creating process-dependent text selection

Texts can also be shown depending on the process variables. Therefore, a process value list and the respective text is defined. In the configurator, there is an own configuration dialog for the text list definition. Select the text format "Text" (instead of "Numerical") to enter a list with value pairs. First, it is to be decided from where the texts are taken. There are three options (as for all text definitions): 1. Use text directly, 2. Use text of a dll (resource text) or 3. Use text from user text file.

If the resource or the user text list should be selected, the "Text List" button opens the dialog for the list creation.

Afterwards, define a "key - value - pair" using the dialog below. In this case, the value corresponds to the value of the variable which is linked to this cell. The value is the text name from the resource, the user text list or the text directly. You can also indicate whether the value has to match the table value exactly or whether it has to be lower or greater.



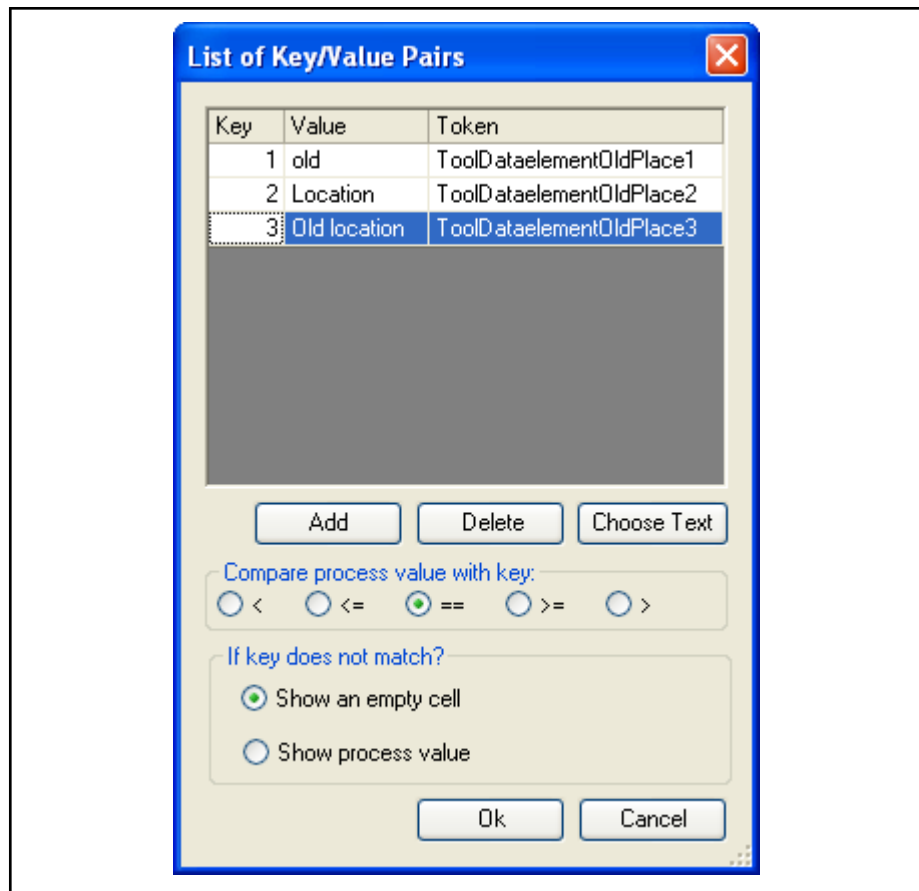
Assembly Dialog to select the DLL with the texts

Resource Combo box with all resources, the dll

Show text list Opens the dialog with the list of values and screens

Fig. 7-40: Configuration dialog to create the text list

The dialog to create a text list looks as follows:

**Add**

Add new line

Delete

Delete line

Select text

Select text from resource or from user text

Upper row of the button

Here it is determined how the process value has to be compared with the value from the list. The images are only displayed in the image above if the process value is equal to one of the values specified.

Button "If key does not fit?"

If the process value does not correspond to any of these values, either an empty line or the process value itself can be displayed.

Fig. 7-41: Dialog: List of keys/value pairs (text list creation)

Editing text list

Text lists are used to select texts from a list and to generate new ones. The texts that are selected can originate from a resource file and can be either texts or images; they can also come from a user text list. The list can be supplemented only if the texts come from a user text list.

Selecting text

Double-click to select a text.

Adding text

To add a new text, position the cursor over the last cell and enter a token in the first column and a text in the second column.

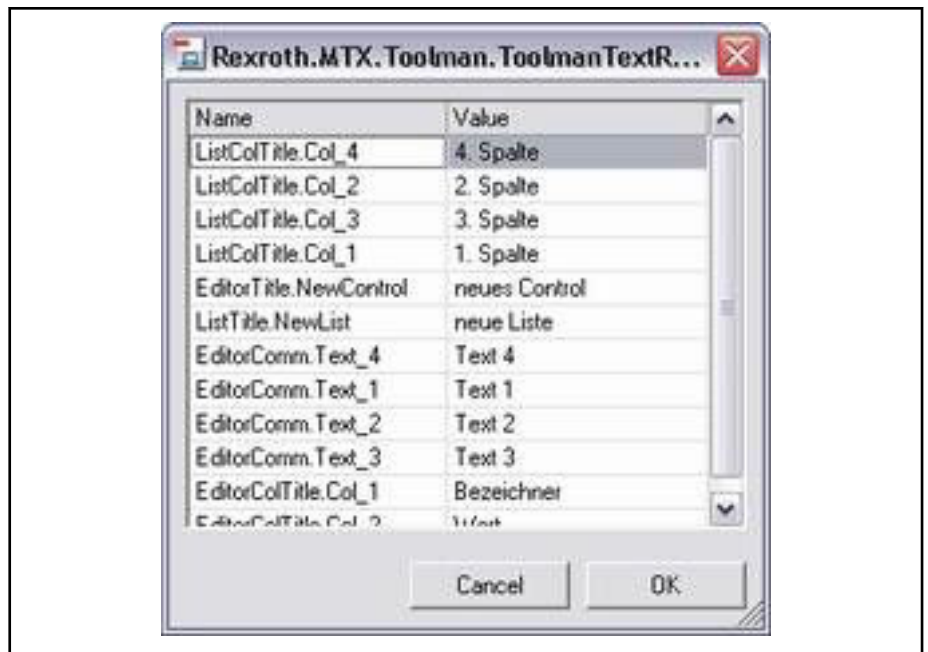


Fig. 7-42: Text list for selecting texts

Handling instruction: Define a new user text

The following handling instruction describes the procedure for defining a new user text in several languages.

Creating a user text in several languages

When creating new texts, the project language should always be the master language.

1. Create a new text (see [chapter "Editing text list" on page 196](#))
2. **File ► Save all**

A new text token is created.

If the set project language was not the master language, the text token for the master language is generated automatically with the prefix "####". Otherwise, the tokens for additional installed project languages remain empty; they can be generated externally using the functions **Project ► Language ► Export translation file...** followed by **Import translation file...**

3. In order to generate a variant of the new text in another language within IW Engineering, the project language must be switched to the desired language.
4. The text for the new text token is displayed in the master language in the text list. The text can now be modified.
5. **File ► Save all**

The new text is now available in two languages.

Screen		Documentation
Documentation:	MTX Functional Description	

Syntax of editing mask

The editing mask has to consist of the following characters:

1) Wildcards

0	Number
9	Number or space
#	Number or character
L	Letter
?	Letter or space
A	Letter or number
a	Letter, number or space
&	Any character

2) Localized characters

.	Localized decimal point
,	Localized separating symbol for thousands
:	Localized separating symbol for time
/	Localized separating symbol for date

3) Command characters

\	The next character will be interpreted as literal (directly as a character and not in its meaning)
>	Change letter to capital letter
<	Change letter to lowercase letter

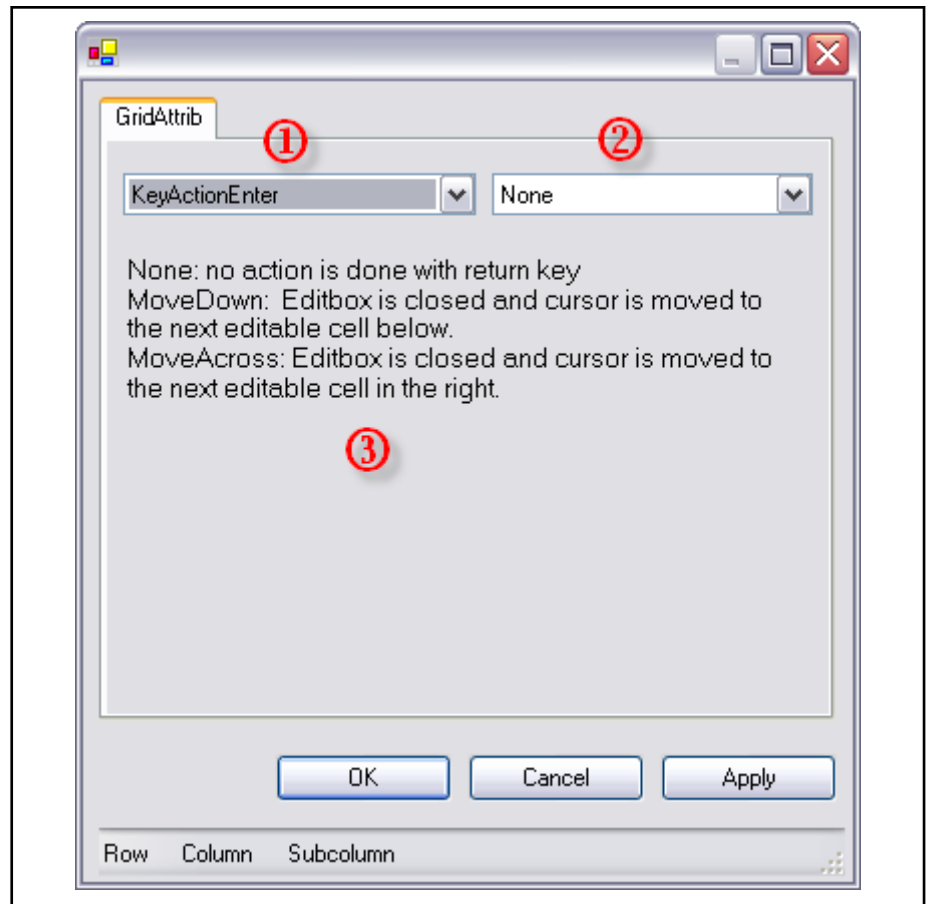
4) Placeholder definition

;	The next character is used as placeholder (default: underscore)

Example: EditMask = 000.00

Sets a mask that forces a three-digit number with two positions after the decimal point.

Setting general table properties



- 1 Process parameter
- 2 Value
- 3 Comment explaining the setting options

Fig. 7-43: Setting dialog for table attributes

General definitions (GridAttributes):

Element	Description
AllowFocusOnNonEditableCells	True: Cursor jumps to all cells False: Cursor skips all non-editable cells
AllowFreezingWithMouse	Shifting the frozen area: 0 = none 1 = Columns 2 = Rows 3 = Both
AllowMerging	True: Allow merging of cells False: Do not allow merging of cells

Control commissioning

Element	Description
AllowResizing	0 = none 1 = Columns 2 = Rows 3 = Both 4 = RowUniform 5 = BothUniform
AutoResize	Column width is set automatically when data exist.
AutoSearch	0 = none 1 = FromTop 2 = FromCursor
AllowSorting	0: Sorting not allowed 1: Single column sorting 2: Multiple column sorting
BorderStyle	1 = None 2 = FixedSingle 3 = Fixed3D 4 = Light3D
CheckRowVisibilityAtBeginning	No user setting required or reasonable
CursorKeyCanCloseEditMode	Cursor movements beyond the edge of the edit window terminates the edit mode
DrawTextFlexgridOrg	True: FlexGrid indication mode for texts and bitmaps False: ULC indication mode for texts and bitmaps
ExtendLastCol	True: The last column fills the entire area
FilenameOfDefaultValues	Name of the default setting file
GridHighlight	Defines when selected cells are highlighted: 0: Never 1: always 2: With focus
KeyActionEnter	Cursor movements after <ENTER>: 0 = None 1 = Down 2 = To the right
KeyActionTab	Cursor movement after <TAB>: 0 = None 1 = Down 2 = To the right

Element	Description
ListBackColor	Background color of the list (visible only if the indicated list area is smaller than the control)
PageDownTrack	0: Data of the next page are not updated while the user presses <PgDn> 1: Data of the next page are updated while the user presses <PgDn>
ScrollTrack	0: The list is not updated while the user moves the scroll bar 1: The list is updated while the user moves the scroll bar
ShowDebugMessages	Trace message output in the debug window
ShowTraceMessages	Display of error messages
SortEmptyRowsToEnd	1: Empty entries are allocated at the end of the list; irrespective of the sorting. Default value: 0
Stripline	Dividing line between fixed and scrollable area
SuppressComma	If UseNumberDecimalSeparator = False: True: Comma is replaced by point
TabCanCloseEditMode	TAB terminates edit mode
TestDOMToConfig	No user setting necessary
TestSOMToConfig	No user setting necessary
UseFixedNumberOfSubRows	1: The first sublist is checked and then used to generate the number of all SubRows. Default value: 0
UseNumberDecimalSeparator	True: The separator defined in the current country setting is used False: Point is used generally

Tab. 7-6: GritAttributes

XML file editor

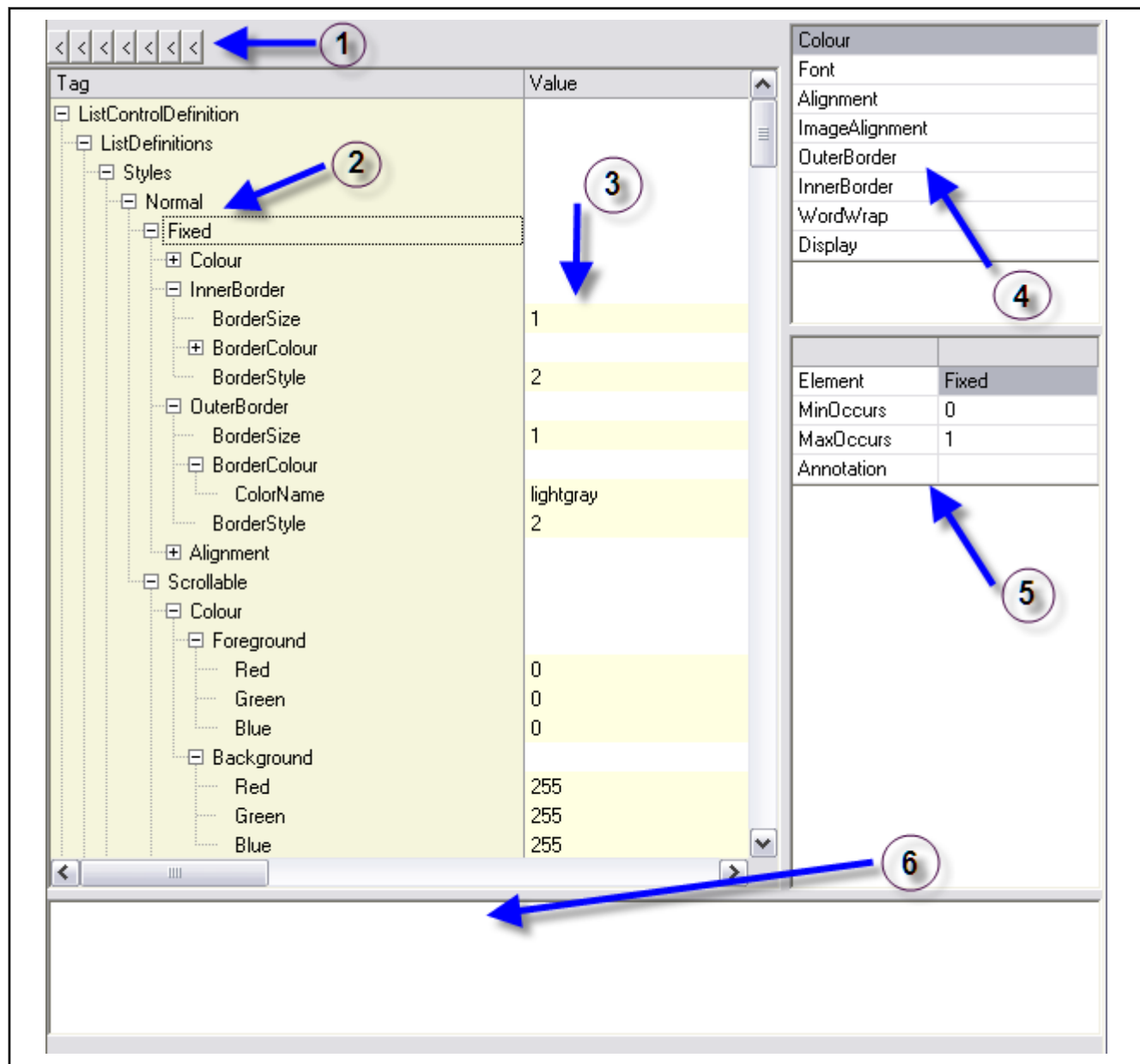
General information

The XML file editor can be accessed via the tab page "XML Editor" of the ULC configurator and can serve as a supplementary view of the tool list and tool editor control configurator.

The XML file editor is especially provided for schema-supported editing. The present version is only opened to configure tool lists and tool editors.

Control commissioning

The advantage of schema support is that nodes defined in the schema can be selected from a list; they do not have to be entered separately. In addition, values that are entered are checked whether they are appropriate for the schema.



- 1 Button for expanding/collapsing nodes
- 2 Node tree
- 3 Value list
- 4 List of possible children of the selected node
- 5 Attributes and comments for the selected node
- 6 Output window

Fig. 7-44: The tool list XML editor

The editor provides the following possibilities:

Pasting known nodes

When an element in the list of possible node children (4) is double-clicked, this element is pasted as a child in the selected node.

Editing a value

1. In the list of values (3), select a cell and double-click or press <Return> to switch to the edit mode.
2. Enter the value and confirm with <Return>.

If the selected node is defined in the schema, the entry is monitored. Only valid values may be entered.

Searching for strings

Press <Ctrl>+<F> or click with the right mouse button on "Find" to activate the search window, with which you can search for strings in the document.

Copying nodes

Select a node and, with the right mouse button, click "Copy node". Now the node, with all the subnodes, is copied as a partial XML tree to the clipboard. From there, it can also be used differently. For example, copy the node into a text editor, modify it there, copy it back to the clipboard and paste it back into the document.

Pasting nodes

If a node was copied to the clipboard previously, it can be pasted anywhere. This procedure is not monitored by the schema! Select the node which is supposed to become a "parent" and, with the right mouse button, open the pop-up menu pressing **Paste Node**.

Moving nodes up

Select the node, press the right mouse button and select "Move node up" in the pop-up menu.

Moving nodes down

Select the node, press the right mouse button and select "Move node down" in the pop-up menu.

Deleting nodes

Select the node and use the <Del> key or right-click and select the "Delete node" in the pop-up menu.

Pasting nodes with freely definable names

Select the node, right-click and select "Add new node" in the pop-up menu. This pastes a node whose name is freely definable. This name does not have to be defined in the schema and can lead to errors during validation.

Pasting a text node

A value can be assigned to nodes that do not have subnodes as children themselves (end nodes). This occurs using a text node. Select the node, right click and select "Add new text node" in the pop-up menu. The text node that was just created can now be filled with any text.

Copying XPath to clipboard

Select the node, press the right mouse key and select "Copy XPath to clipboard" in the pop-up menu. The path from the basic node to the selected object can now be found in the clipboard.

Expanding and collapsing individual node levels

For every node level, a button used to open or close the corresponding node level is created when the XML document is loaded. If a subordinate level is expanded, all superordinate nodes are also expanded.

7.10.3 General DB configuration

Number of the DBT used

General information

All general settings for the database visualizations (e.g. tool management) are selected via the "Properties" dialog which can for instance be called via **DBTx Screens (tool management) ► Properties....**

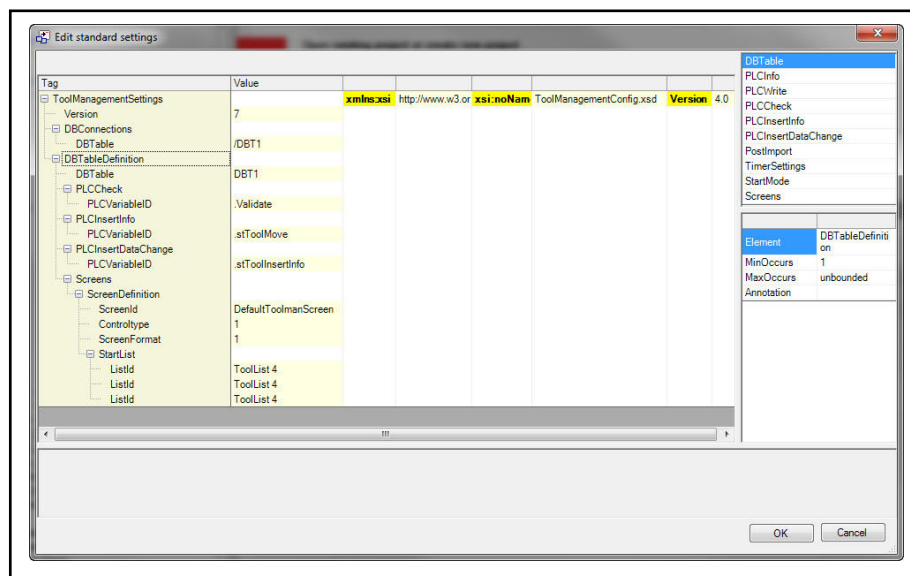


Fig. 7-45: Tool management

Activating second database table

Up to two database tables can be used.

If the second database table should be used, respective settings have to be made in the file "ToolManagementConfig.xml".

First, activate the communication in the second database table:

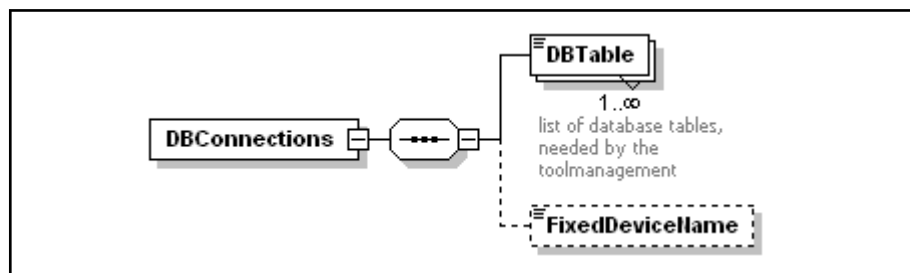
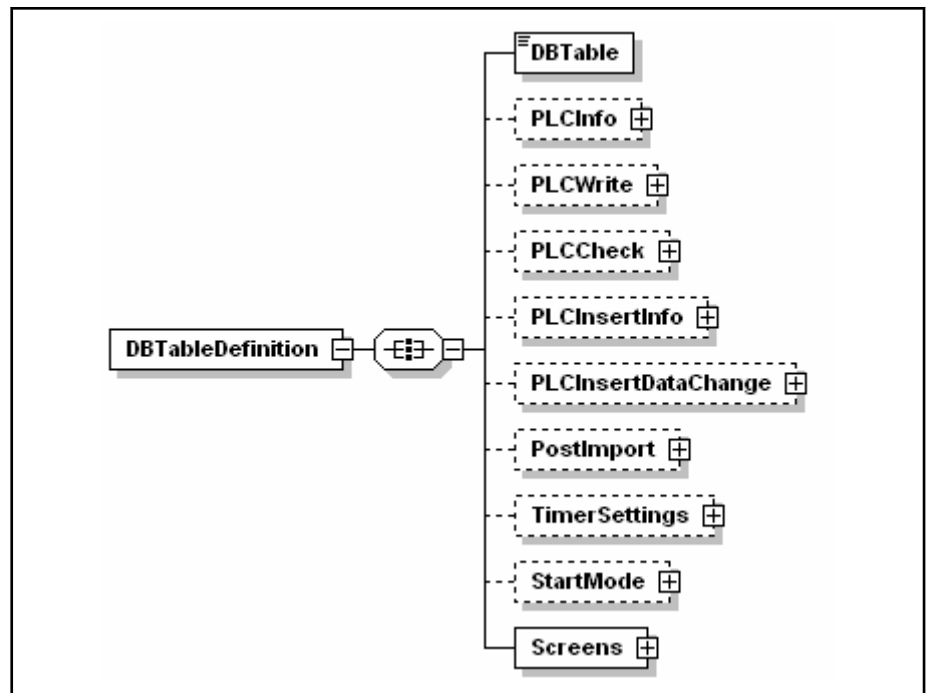


Fig. 7-46: Activating second database table

If these entries are missing under **DBConnections**, it is only communicated with the database table 1.

Furthermore, if applicable, all functional basic settings should be made for DBT2 as made for DBT1.



DBTable	ID entry
PLCInfo	Definition of the PLC variable name for the "Info" function
PLCWrite	Definition of the PLC variable name for the "Write" function
PLCCheck	Definition of the PLC variable name for the "Validate" function
PLCInsertInfo	Definition of the PLC variable name for the InsertInfo function
Post import	Definition of the "Postimport" function
TimerSettings	Low-pass filter configuration
StartMode	Setting the edit mode used for starting. "true" or "false" are used for the configuration. Note the spelling.
Screens	Contains screen definitions for DBT

Fig. 7-47: Structure diagram 1: ToolManagementConfig.xml

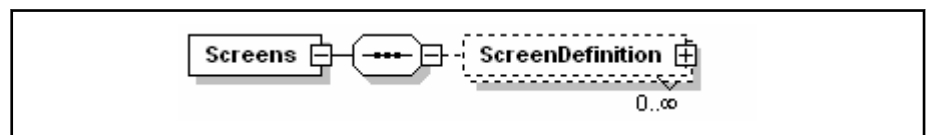
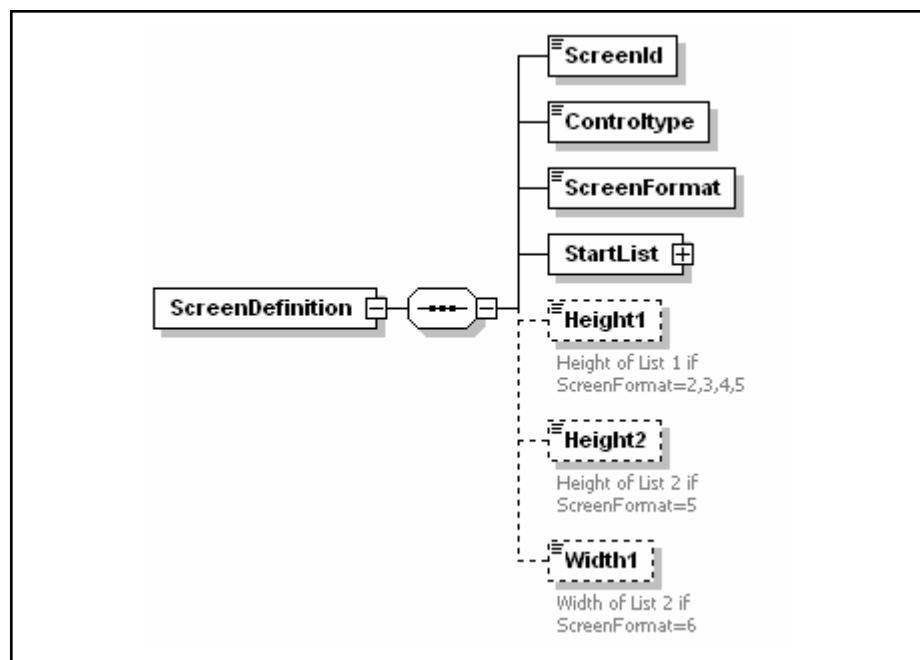


Fig. 7-48: Block diagram 2 ToolManagementConfig.xml



- ScreenID** is automatically specified
- Controltype** is automatically specified (see [chapter "General information" on page 207](#))
- ScreenFormat** is automatically specified (see [chapter "General information" on page 207](#))
- StartList** defines the starting list of the screen
- Height1** Height of list 1 (relevant for manual size setting of the lists for ScreenFormat 2,3,4,5)
- Height2** Height of list 2 (relevant for manual size setting of the lists for ScreenFormat 5)
- Width1** Width of list 1 (relevant for manual size setting of the lists for ScreenFormat 6)

Fig. 7-49: Block diagram 3 ToolManagementConfig.xml

Creating a DBT2 screen in IndraWorks Engineering

To visualize the DBT2 data in the IWO, it is possible to define an own visualization screen. Furthermore, it is still possible to call and visualize DBT2 lists within the operating area tool management.

Handling instruction: Configuration to use the second database table

The following handling instruction describes the process to be followed when further database tables are to be used.

Activate communication with the second database table

1. Open the "Properties" dialog of the database applications via **DBT1 Screens (tool management) ► Properties....**
2. Add the entry for the second database table under the node "**<DBConnections>**" (refer to the following Fig.).

Program:

```

<DBConnections>
  <DBTable>/DBT1</DBTable>
  <DBTable>/DBT2</DBTable>
</DBConnections>
  
```


Basic setting definitions of the management for the second database table

1. Open the "Properties" dialog of the database applications via **DBT1 Screens (tool management) ► Properties....**
2. Add another node for the DBT2 in the same way as for partial tree "**<DBTableDefinition>**" for the DBT1 and enter all necessary entries for the second database table under this node (see example in the following figure).

Program:

```
<DBTableDefinition>
  <DBTable>DBT2</DBTable>
  <PostImport>
    <Action>
      <ProcVariableID>Hd/BQ2</ProcVariableID>
      <Operation>AND</Operation>
      <Value>FFFFFF7F</Value>
    </Action>
  </PostImport>
  <PLCCheck>
    <PLCVariableID>.Validate</PLCVariableID>
  </PLCCheck>
  ...
</DBTableDefinition>
```

Activation of DBT2 in the project and setting up the user interface configuration

1. Close the project to apply the changes.
2. Open the project again in IndraWorks Engineering.

The database table 2 in the project tree does now have an individual node.

Under this node all interface settings for the DBT2 can be made.

	Documentation	
Documentation:	MTX Functional Description	Apply new data structure

Configuring several DBT visualizations

General information

There is the possibility of visualizing the database tables in one or in several applications. This applies both for database table 1 which is exclusively intended for the management of tool data and for database table 2 which also allows for the management of other user data.

The configuration of the database is the same for all database tables (refer to [chapter 7.10.4 "Configuring the database" on page 213](#)).

The visualization of the data of one database table requires at least one visualization screen per DBT. For the DBT1 there is always a "Default ToolmanScreen". Further screens can be defined for a DB table via the menu function **New Screen....** Depending on the node under which the new screen is created, the DBT allocation of the latter takes place.

In the following dialog all further configuration steps for the screen definition are made:

- Specifying the application type
- Specifying the list number and the display format
- Specifying the start lists

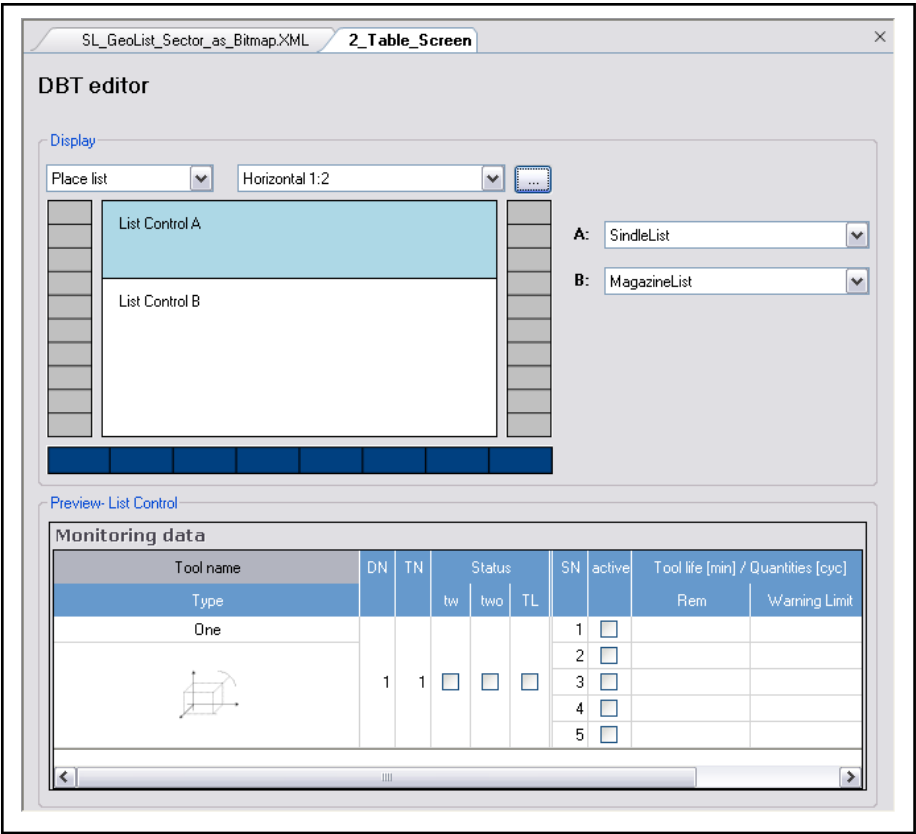
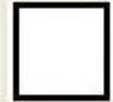
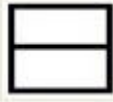
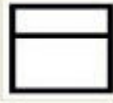
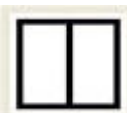


Fig. 7-50: Screen configuration

Display format	Number of lists	Properties	ScreenFormat
Default	1		1
Horizontal 1:1	2		2
Horizontal 1:2	2		3
Horizontal 1:3	2		4

Display format	Number of lists	Properties	ScreenFormat
Horizontal 1:1:1	3		5
Vertical 1:1	2		6

Tab. 7-7: Display format

If the database tables are used in different operating areas - which in turn requires the definition of different applications or screens - these applications can have the following properties.

Applications type	Use case	Properties
As place-related data block management Control type = 1	Tool management	• SKQ as a criterion for assigned place • derived from this, the F-key and menu functions are placed actively or inactive • Evaluation of the PD attribute in the schema
	Workpiece management	
As universal data block management Control type = 2	any data container (reasonable only for DBT2)	• All data elements are equal without downstream display and function logic. • The attributes "UA", "SE" and "PD" in the schema do not have any function. • The data element BQ3 for the type-related display control is free and can be made further use of as bit array. • As opposed to the system data, the DBT2 has the effect of a data container in sectors with predefined memory and access functions typically for a database.

Tab. 7-8: Application types for database tables

There is still the possibility of visualizing both database tables via different lists in one screen.

Handling instruction: Configuration of several different applications

The following handling instruction describes the process to be followed when visualizations of database tables in different operating areas are to take place.

Creating a new DBT Screen

1. Open IndraWorks Engineering
2. Select the node **"Screens"** in the project tree under the HMI device.
3. Select **New screen...** in the menu.
4. Define the screen name in the following dialogue.

A new node will appear in the tree with the name of the new screen.

Setting the Properties of the new DBT Screen

1. Select the node of the new screen in the tree and press <ENTER> or double-click on this node.

The screen editor for the new picture is opened.

2. In the selection box "Screen type" the type **"MTX ToolList"** is selected.
3. In the selection box "Table" the application type is selected (refer to [tab. 7-8 "Application types for database tables" on page 209](#)).

Subject to the selection, the following is displayed in the field "Table":

Selection	Entry
DBT1 place list	DBT1 ;1
DBT2 place list	DBT2 ;1
DBT1 general List	DBT2 ;2

4. In the selection box "Operating area" the application is allocated to an operating area.



F- or M-key configurations should not be made, since these take place when defining the list.

Call Definition via F-key or M-key

1. Selection of the corresponding F- or M-keypad of the operating area which the application was assigned to and which contains the key provided for the call of the application.
2. Opening the selected panel.
F- or M-key configurator is opened.
3. Selection of the corresponding key and definition of the key labeling.
4. In the selection box "Function" **"Screen change"** is selected.
The selection box "Screen name" becomes active.
5. The name of the newly created DBT application is selected in the selection box "Screen name".
6. Close editor and save change.

	Documentation	
Documentation:	MTX Functional Description	Apply new data structure

Configuring multiple-table screens

Creating a multiple-table screen

None

1. Create a new tool screen or DBT screen via <New Screen>.

Depending on the node under which the new screen is created, the DBT assignment of the latter takes place.

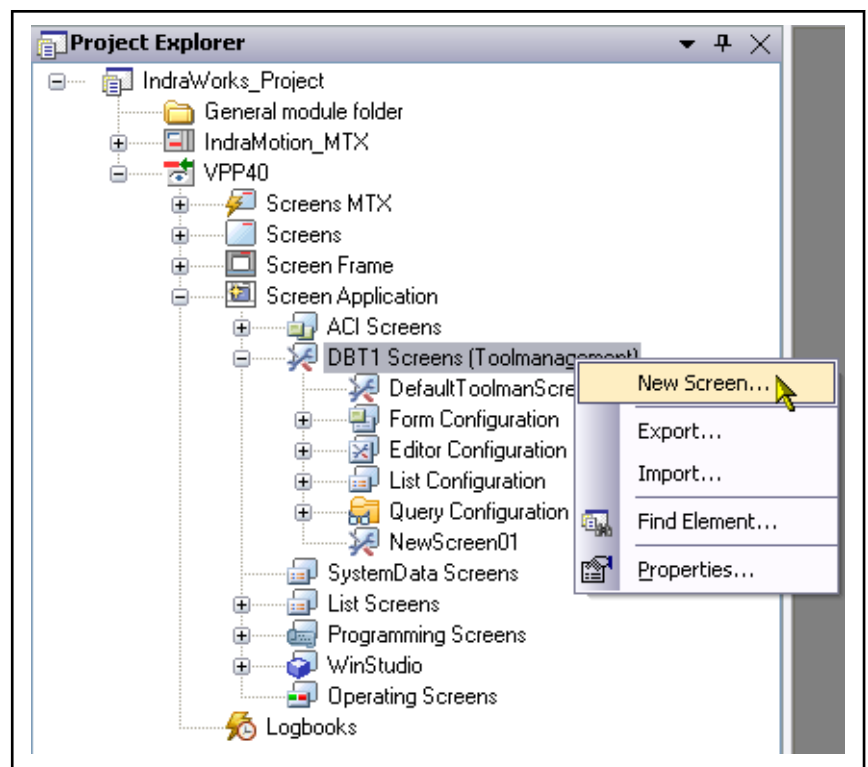
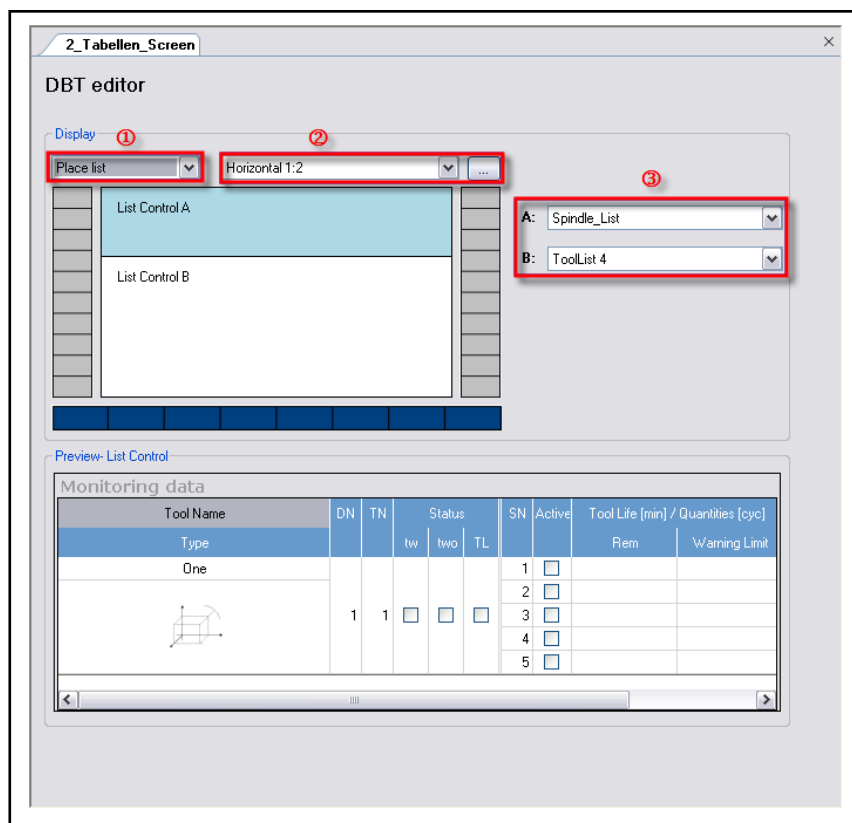


Fig. 7-51:

New entry under "DBT1 Screens (Tool Management)" or "DBT2 Screens"

2. Configure new screen

Control commissioning

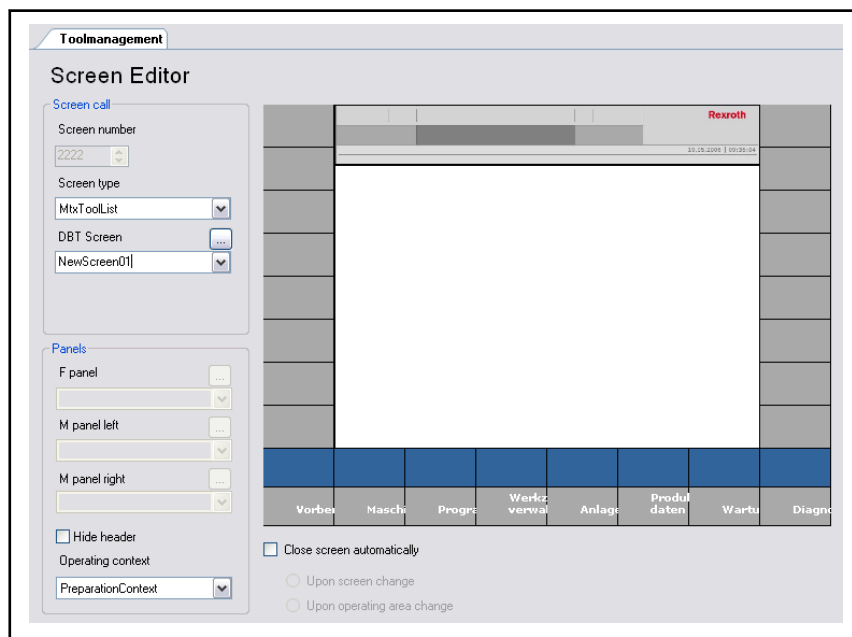


- 1: Determine list type
- 2: Determine display format
- 3: Specify starting list(s)

Fig. 7-52:

Configure new screen

3. Identify new screen for HMI device and assign to operating area.



- 1: Specify image type
- 2: Panel definition performed in list definition
- 3: Assign operating area

Fig. 7-53:

Use DBT1 screen in screen editor

4. Opening a new screen via an F-key in the selected operating area.

7.10.4 Configuring the database

General information

Settings for a control reset in case of changes made to the database configuration:

Changes in:	Necessary action:
dbtxprms.dat	Control reset with restart mode 6
Changing the element structure in the tool data schema	Control reset with restart mode 6
Changing attributes/restrictions in the tool data schema	Control reset with restart mode 0

Tab. 7-9: Reset settings



A control reset with restart mode 6 and a control reset with restart mode 0 and the parameter 2 = 6 in the file "dbtxprms.dat" always causes the deletion of all data blocks of the database table.

Defining sector and place distribution of database table

General information

The sector-place structure of the database table is specified as follows in the configuration files. For each database table, a specific file is available: for DBT1, it is the file "dbt1prms.dat", and for DBT2, it is the file "dbt2prms.dat".

Program:

```

; -----
;           Configuration file for the DB table DBT1
; -----
; general design:
; P0 P1 P2 P3 P4 P5 P6
; |  |  |  |  |  |  |
; |  |  |  |  |  |  +--- string: may also contain " " and \t
; |  |  |  |  |  |  End identification is "line end"
; +-----+-----+-----+-----+-----+-----+ 6 int values
; P0 describes the object type and may not be modified
; Empty string marked with "."
; ";" marks comment lines At the end of a data line
; No comment must be added
;
; *****
;           Configuration of the DB table
; *****
; Control parameter
; P0: 0
; P1: Controls the creation of the data records in the database
;       0 Standard operation
;       Data records are only created when the tool database
;       is empty
;       6 All existing data records are deleted first and;
;       then created again according to the
;       sector configuration
0 0
; Sector configuration
; P0: 1-99 (corresponds to sectors 1 - 99)
; P1: Number of places in the sector
1 5
2 5

```

Changing the sector and place distribution is applied after a control reset with mode 6.



The existing content of the database is deleted and cannot be re-stored! If existing tool data is required, export the data before making any change to the database. If the number of data blocks is smaller than the number of data blocks included in the export file, this file has to be adjusted using a suitable XML editor.

Handling instruction: Defining the sector and place distribution of a database table

This handling instruction describes how to modify the sector and place distribution for a database table.

IW Operation/Program: Edit the file "dbt?prms.dat"

1. Copy the file "dbt?prms.dat" (?:= 1 [DBT1] or 2 [DBT2]) to the mount directory (\mnt) in the control directory "\usrfep" or "\feprom".
2. Edit the file via the editor, enter the desired values for the number of sectors and places (see the program in [chapter "General information" on page 213](#)).
3. Save file.
4. Copy the file to the "\usrfep" control directory.

	Flowchart	Documentation
Flowchart		Flowchart fig. 1-7
Documentation:	MTX Functional Description	Define data blocks

NC: Data transfer

1. Close IW Operation.
2. Start the control with "Startup Mode 6". Use the **MTX System Status** function.



⇒ It is absolutely necessary to execute an NC data backup and to import the data.

	Flowchart	Documentation
Flowchart		Flowchart fig. 1-7
Documentation:	MTX Functional Description	Define data blocks

Configuring data blocks

General information

Since the current version does not yet have a Settings dialog for configuring the user interface, the corresponding scheme or XML document files must be adjusted by means of a suitable editor (for example, see [chapter "Schema editor" on page 179](#)). This documentation proceeds on the assumption that the universally available Notepad editor is used.

To describe the tool data block schema, the following XSD schema files are installed by default in the FEPROM/scheme range of the MTX. To adjust them, copy these schema files to the Userfep/Schema directory. If there is no such directory, create it.

The files can only be edited in the mount or another Windows directory. Subsequently, copy the changed schema files to the Userfep/Schema directory. The changes in the control kernel are applied after control restart.

The tool data structure is described by a hierarchically structured system of XSD files:

dbt1sd.xsd	Tool data system structure
dbt1ud.xsd	Tool user data structure
basic_tool_ty.xsd	Tool data type collection, part 1
tool_ty.xsd	Tool data type collection, part 2
basic_ty.xsd	Basic data type collection

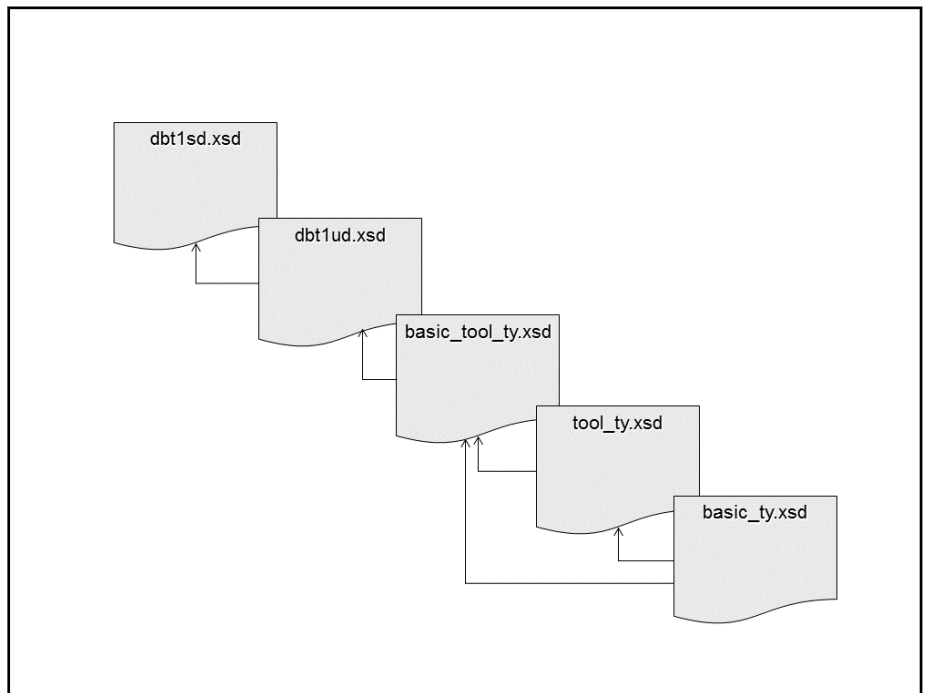
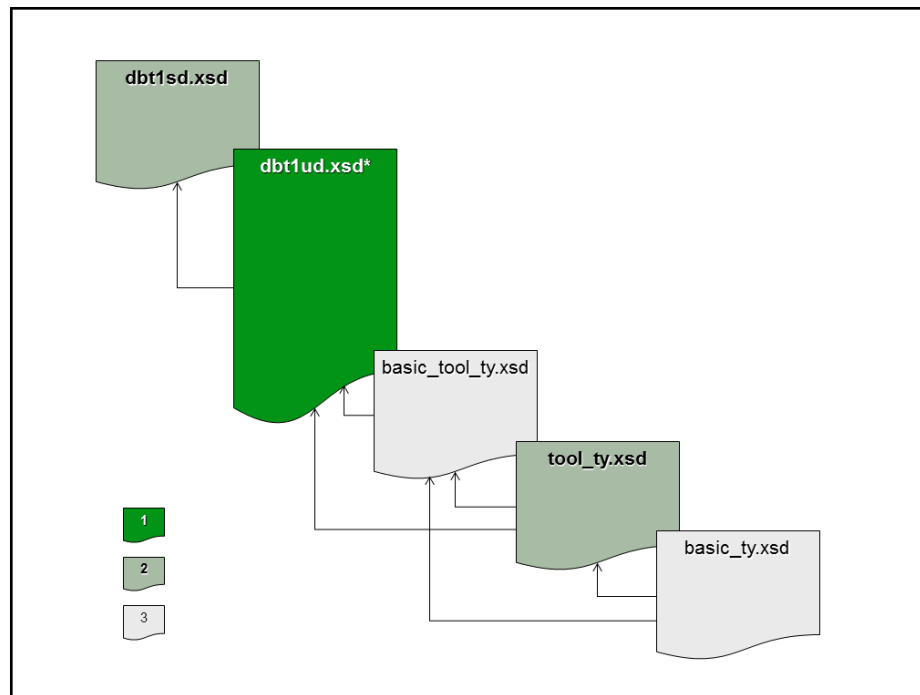
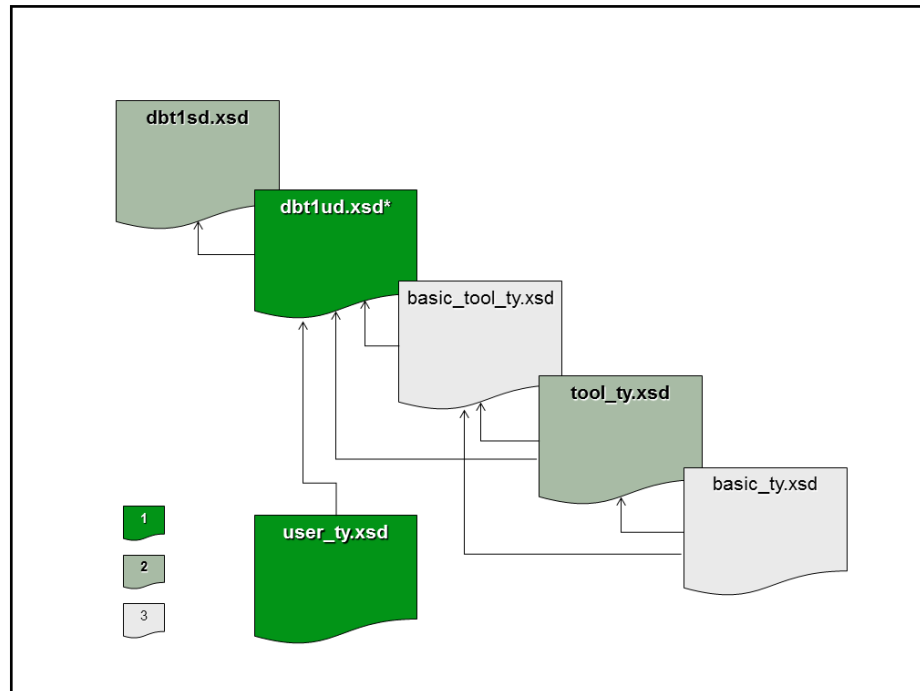


Fig. 7-54: Schema configuration (delivery state)



- 1 Changed scheme file
- 2 Modifiable scheme file
- 3 Scheme file that **cannot** be modified

Fig. 7-55: Modified schema configuration (recommended variant)



- 1 Changed scheme file
- 2 Modifiable scheme file
- 3 scheme file that **cannot** be modified

Fig. 7-56: Modified schema configuration

Ideally, the user only has to extend the user data structure in the "dbt1ud.xsd" file.

Tool system data structure (dbt1sd.xsd)

This file describes the system structure permanently preset in the MTX for all tool management systems implemented with database table 1. The "DBT1Hd_t" data type in this file is a standard data type preset by the system. It contains a series of basic tool data which are used by the system functions, e.g. the tool catalog, and standard functions such as place and tool search functions. Thus, the user must not change the data structure of this schema.

include	loc:dbt1du.xsd
element	DBT1
complexType	DBT1Rec_t
complexType	DBT1Hd_t

Tab. 7-10: Content of the system structure

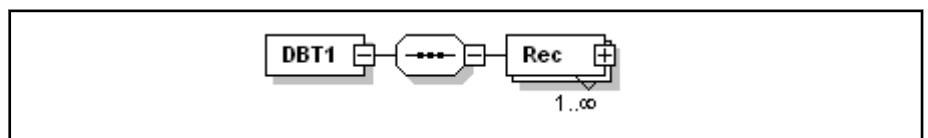


Fig. 7-57: Data block schema (1)

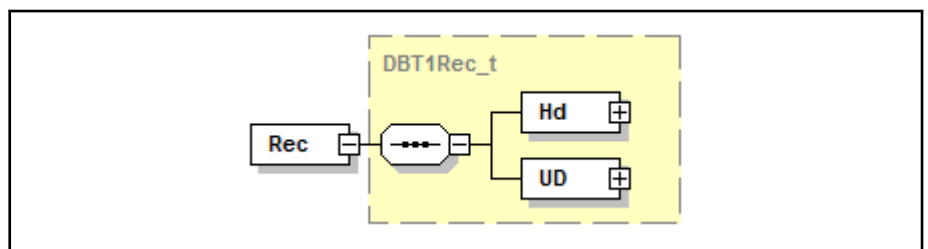


Fig. 7-58: Data block schema (2)

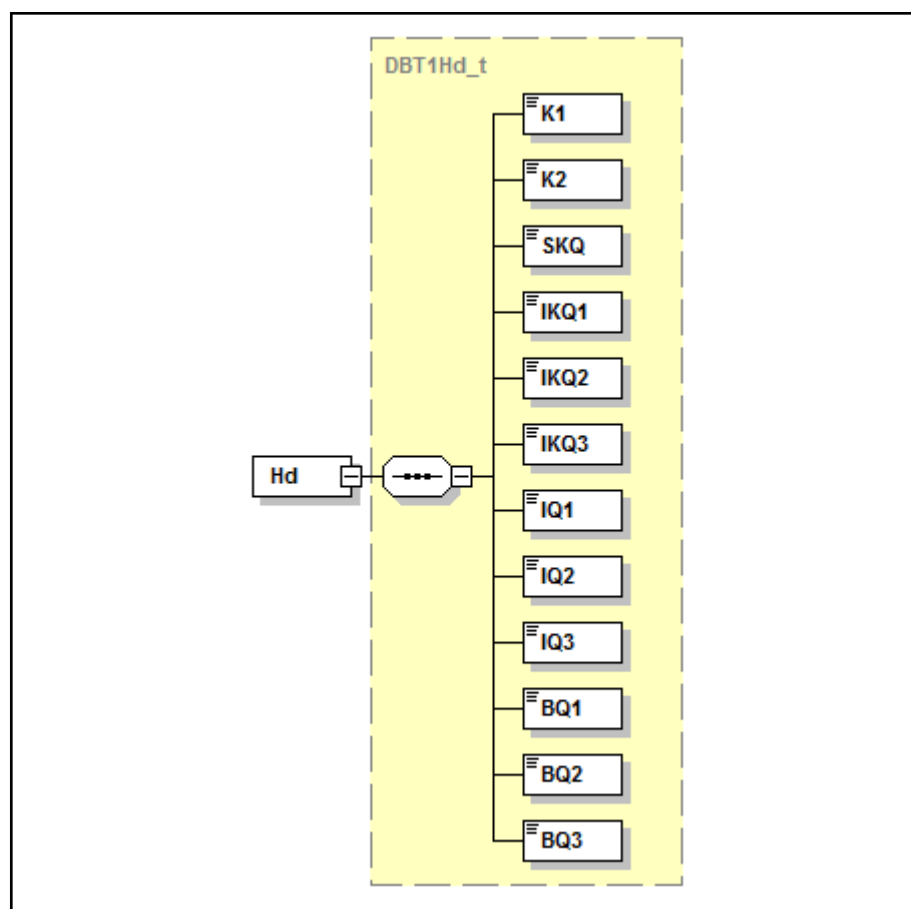


Fig. 7-59: Data block schema (3) (header schema)

The significance of the following header data elements is permanently defined for the tool data block:

K1:	Sector (memory)	Integer
K2:	Place	Integer
SKQ:	Tool ID	String
IKQ1:	Duplo number	Integer
IKQ2:	Tool type	Integer
IKQ3:	Tool number	Integer
IQ1:	Free (reserved for multi-table system)	Integer
IQ2:	Free (reserved for multi-table system)	Integer
IQ3:	Free (reserved for multi-table system)	Integer
BQ1:	Place status	Bit array
BQ2:	Tool status	Bit array
BQ3:	Tool type description	Bit array

Tool user data structure (dbt1ud.xsd)

Using the substructures predefined in the tool data type collection, this file describes the parts of the tool data structure to be modified. In the current case, the user structure consists of:

		Type from V14:	Type > V09:	Type > V04:	Type in V02:
Tool basic data structure	TI	TI_V14_t	TI_V09_t	TI_V04_t	TI_t
Tool edge data structure	Ed		Ed_V09_t	Ed_V04_t	Ed_t

Tab. 7-11: Tool user data structure

These types are defined in type collection "tool_ty.xsd" and "basic_tool_ty.xsd".

The maximum number of tool edge data structures and, thus the maximum tool edge number is defined in the data block via the entry `<xs:element name="Ed" maxOccurs="2">`.

include	loc:tool_ty.xsd
include	loc:basic_tool_ty.xsd
complexType	DBT1UD_t

Tab. 7-12: Content of the user data structure

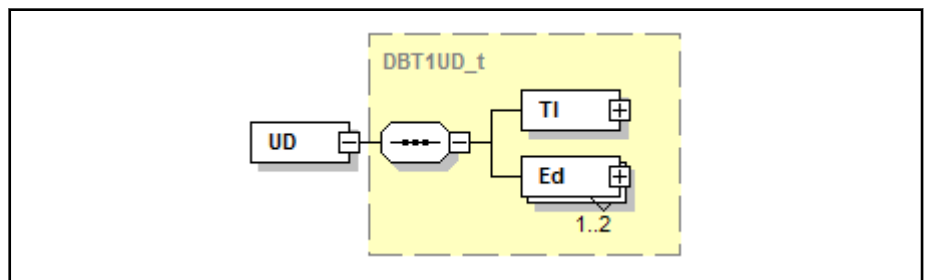


Fig. 7-60: Data block schema (4) (user data structure upon delivery)

Change of maximum number of tool edges

The number of tool edges can be changed using the following entry (maxOccurs) in this schema file:

Example:

Maximum number of tool edges = 2:

Program:

```
<xs:complexType name="DBT1UD_t">
  <xs:sequence>
    <xs:element name="Tl" type="Tl_V14_t"/>
    <xs:element name="Ed" type="Ed_V09_t" maxOccurs="2"/>
  </xs:sequence>
</xs:complexType>
```

The default tool data block was defined for up to two cutting edges. If this number should be changed, the following system data setting must be adjusted in the **SDDat.xml** file to ensure a correct display of the active tool correction in the operating area "Machine".

Program:

```
...
<SysMaxEd>
  <MaxEd>2</MaxEd>
```

```
</SysMaxEd>
...
```

Tool data type collection (tool_ty.xsd)

This schema file includes a collection of partial tool data structures predefined in the MTX as a standard and intended as components to define the tool user data structure. Note that all data elements in the MTX not contained in the header has to be considered as user data. This type collection was further developed for the MTX 04VRS and contains redundant data types due to compatibility reasons. These data types are labeled in the following table with "Only for V02 standard projects".



If legacy projects are used without modifications, the following modification must be made in file "dbt1ud.xsd":

The following lines

...

```
<xs:element name="TI" type="TI_V14t"/>
<xs:element name="Ed" type="Ed_V09_t" maxOccurs="2"/>
```

...

are replaced by

...

```
<xs:element name="TI" type="TI_V09t"/>
<xs:element name="Ed" type="Ed_V09_t" maxOccurs="2"/>
```

...

or

...

```
<xs:element name="TI" type="TI_V04_t"/>
<xs:element name="Ed" type="Ed_V04_t" maxOccurs="2"/>
```

...

or

...

```
<xs:element name="TI" type="TI_t"/>
<xs:element name="Ed" type="Ed_t" maxOccurs="2"/>
```

...

The modified file has then to be located in directory usrfepl\schemas.

To use new functions, the schemas have to be extended according to the standard of version 04.

If the new structure is used, existing PLC and CPL programs from version 02VRS have to be adapted.

		Remarks
include	loc:basic_ty.xsd	
complexType	TI_t	For V02 default projects only
simpleType	LifeUnit_t	
complexType	TI_V04_t	New default type from V04

		Remarks
include	loc:basic_ty.xsd	
complexType	TI_V09_t	New default type from V09
complexType	Ed_t	For V02 default projects only
complexType	Ed_V04_t	New default type from V04
complexType	Ed_V09_t	New default type from V09
complexType	Geo_t	For V02 default projects only
complexType	Geo_V04_t	New default type from V04
complexType	Wear_t	
complexType	EdLife_t	For V02 default projects only
complexType	EdLife_V04_t	New default type from V04
complexType	Limits_V04_t	New default type from V04

Tab. 7-13: Content of the tool data type collection (tool_ty.xsd)

		Remarks
include	loc:basic_ty.xsd	
include	loc:tool_ty.xsd	
complexType	TI_V14_t	New default type from V14
simpleType	ToolOri_t	Tool orientation

Tab. 7-14: Content of the tool data type collection (basic_tool_ty.xsd)

Only the data structure of the new default projects for V04 and higher is described below.

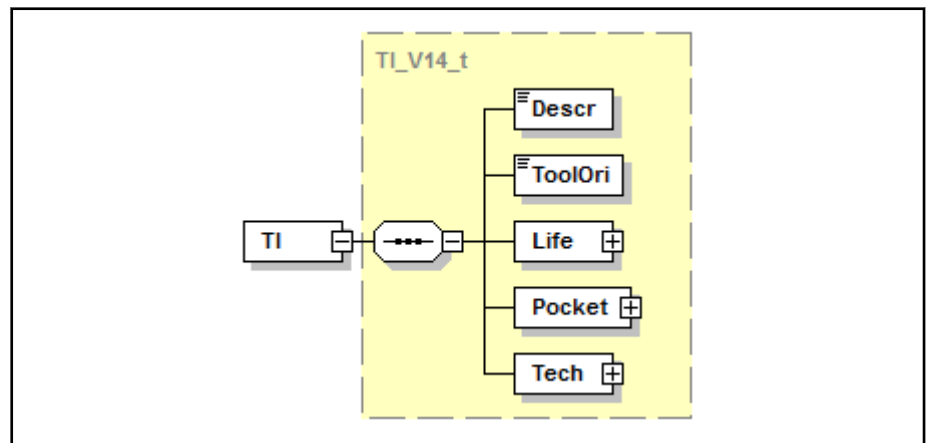


Fig. 7-61: Data block schema (5) (tool data upon delivery)

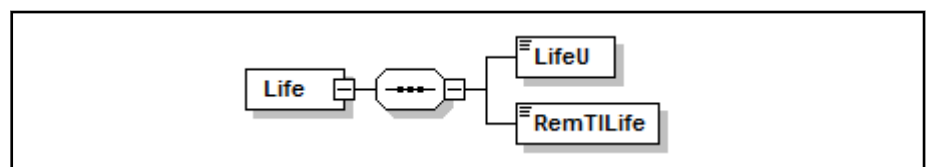


Fig. 7-62: Data block schema (6) (tool life data)

Control commissioning

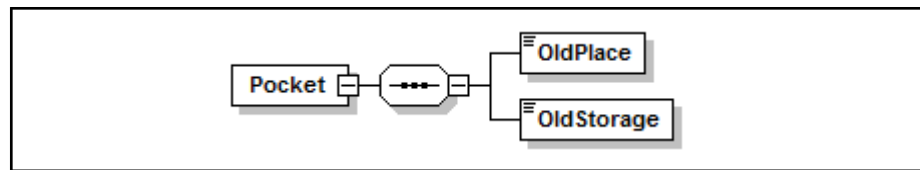


Fig. 7-63: Data block schema (7) (tool data)

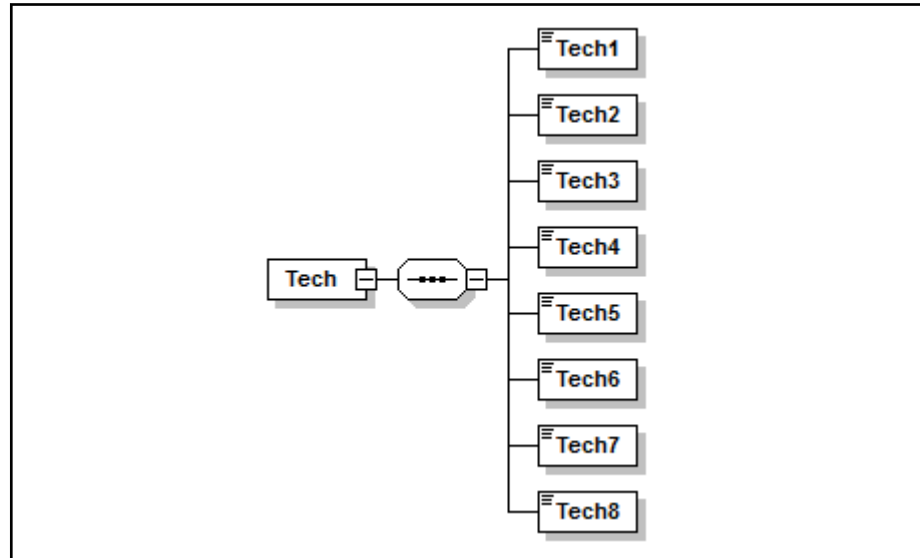


Fig. 7-64: Data block schema (8) (tool technology data)

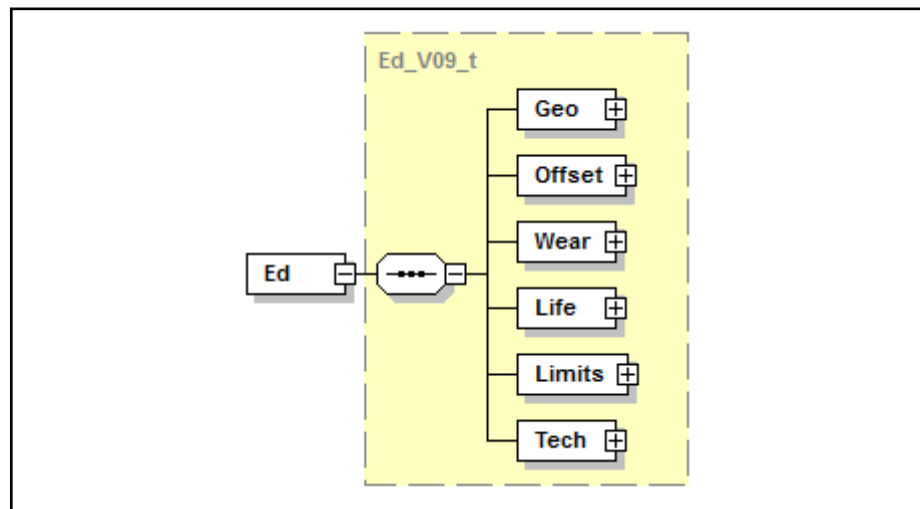


Fig. 7-65: Data block schema (9) (tool edge data upon delivery)

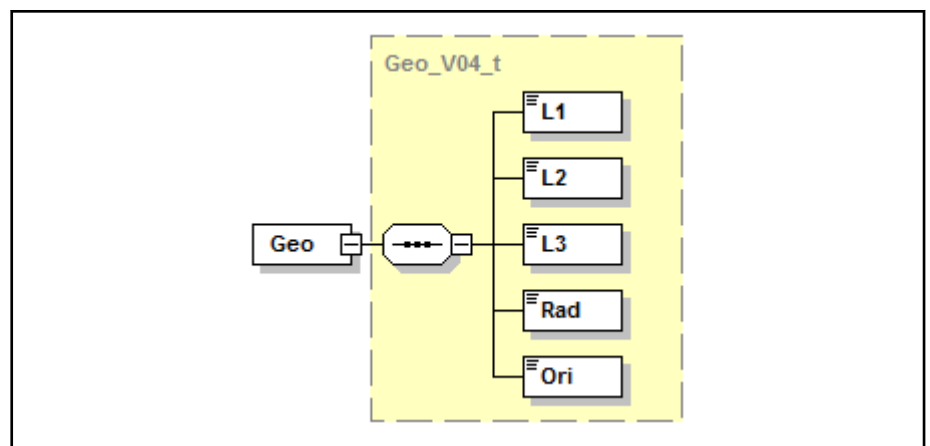


Fig. 7-66: Data block schema (10) (geometry data upon delivery)

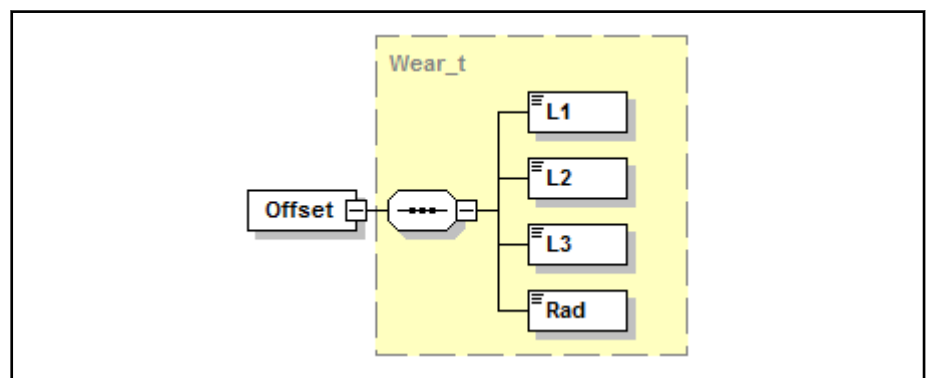


Fig. 7-67: Data block schema (11) (offset data upon delivery)

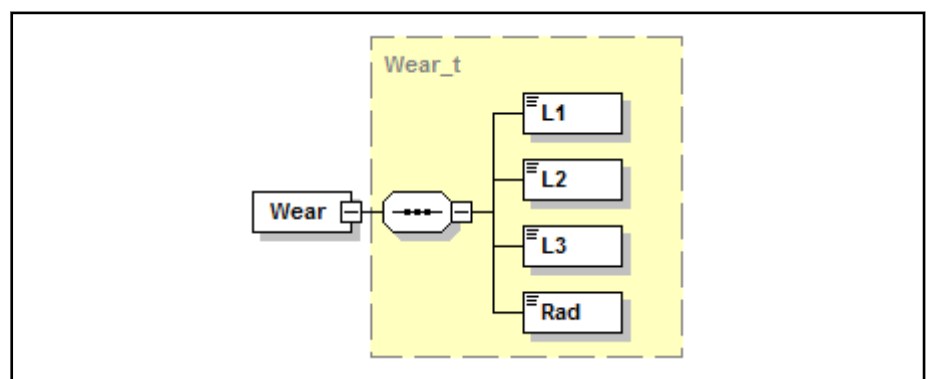


Fig. 7-68: Data block schema (12) (wear data upon delivery)

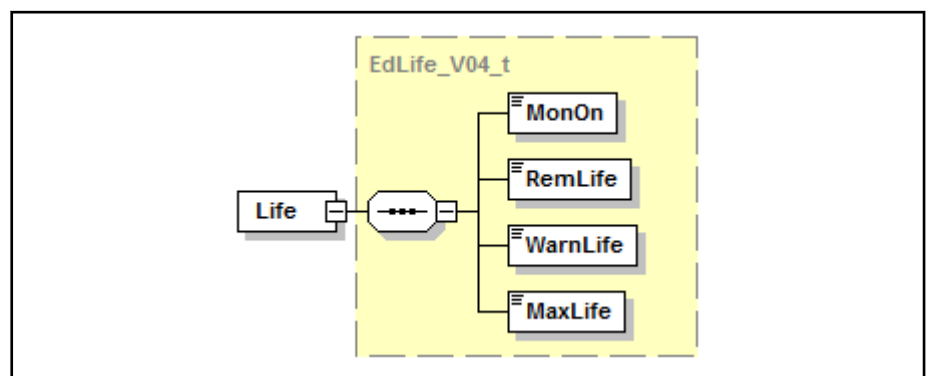


Fig. 7-69: Data block schema (13) (tool life data upon delivery)

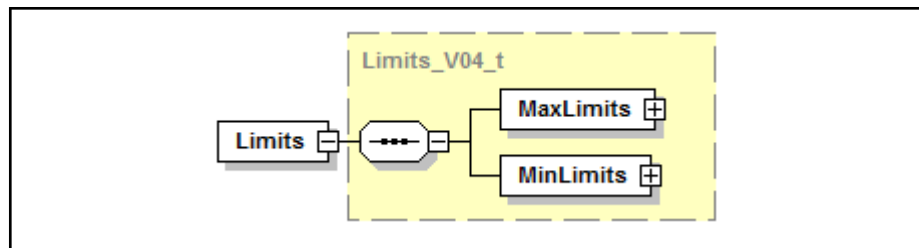


Fig. 7-70: Data block schema (13) (limit value data)

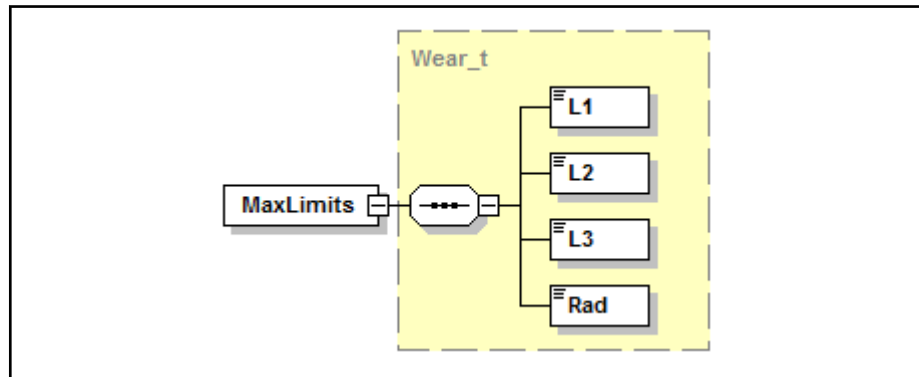


Fig. 7-71: Data block schema (14) (max. limit value data)

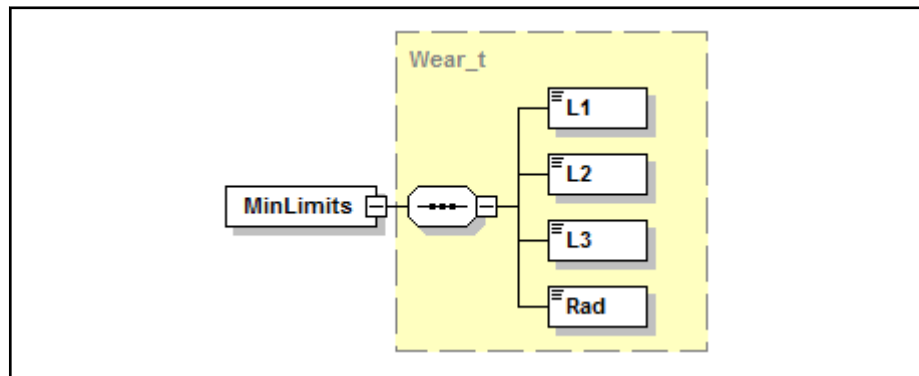


Fig. 7-72: Data block schema (15) (min. limit value data)

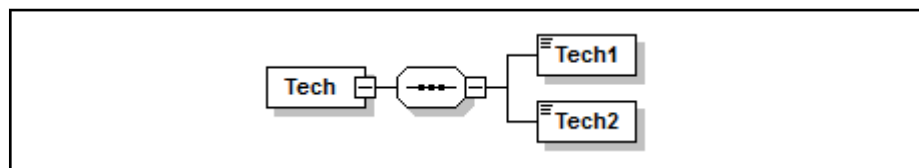


Fig. 7-73: Data block schema (16) (cutting technology data)

Basic data type collection (basic_ty.xsd)

The schema file is available for all data schema definitions used by the MTX. It cannot be changed by the user. However, the user can use these basic types in the new data structures generated in the schema file **dbt1ud.xsd**.

Defining basic settings of the tool management

The following settings are made using the attribute definitions in the schema files:

- Definition of user rights
- Additive value input for certain data elements yes/no
- Limits for value increase with additive value input
- Data element type definition (place or tool data element)

- Activation of tool-specific limit value check of input values in the PLC
- Activation of the message to the PLC on a value change in the data block
- Activation of a value change in the data block by the PLC
- Activation of the message to the PLC on inserting, deleting or moving a data block using the interface

Definition					Note			Use	
				Example					
Name	Type	Use	Default	Fixed	Description	Do-main	Description	Node	Element
L1	xs:string	Option-al		R	User rights for L1 users	R	Read-only	x	x
						RW	Read and write		
L2	xs:string	Option-al		R	User rights for L2 users	R	Read-only	x	x
						RW	Read and write		
L3	xs:string	Option-al		R	User rights for L3 users	R	Read-only	x	x
						RW	Read and write		
L4	xs:string	Option-al		RW	User rights for L4 users	R	Read-only	x	x
						RW	Read and write		
L5	xs:string	Option-al		RW	User rights for L5 users	R	Read-only	x	x
						RW	Read and write		
ETA	xs:string	Option-al		IA	Valid editing type	With-out	Only absolute without diameter		x
						IA	Incremental and absolute without diameter		
						DI	Incremental with diameter and absolute without diameter		
MaxIncIn	xs:dou-ble	Option-al	0.5		Maximum growth ≤ Limit value				x
MaxIncEx	xs:dou-ble	Option-al	0.5		Maximum growth < Limit value				x
DataType	xs:string	Option-al		PD	Data type attribute	PD	Place date		x
						- or TD	Tool / workpiece date		
PLCCheck	xs:string	Option-al		Lim	Type of communication with the PLC during editing	Lim	Limit value check by PLC		x
						Write	Checking and writing by PLC		
						Info	Information that writing occurred to PLC		

Tab. 7-15: User-definable data element attributes for database tables

Control commissioning

For completeness, the following two tables describe additional attribute definitions. However, these cannot be changed by the user during the configuration of the data block.

Definition					Note		
			Example				
Name	Type	Use	Default	Fixed	Description	Domain	Description
UA	xs:string	Optional		ID	Use attribute: - Application should be unique for the entire scheme, i.e. each value may be used only once in the scheme - The attribute is required as information source for system routines	TSt	Sector
						TPI	Place
						TID	ID element
						TT	Type element
						TC	Type code el.
						TS	Tool/part status
						PS	Place status
						TN	Number element
						DN	Duplo number element
						Rec	Record node
						Hd	Header node
						Ud	User data node
						Ed	Tool edge data node
SE	xs:Boolean	Optional		1	Significant data element - useful only for list display - If used together with UA=ID in the tool list, the data element in the list is protected	1	Char. DS valid/place occupied
U	xs:string	Optional	mm		Unit attribute	mm	
						Inch	
						Nm	
						%	
Class	xs:string	Optional		DBTAB	Assignment attribute		
V	xs:string	required		00T00	Version attribute		

Tab. 7-16: System-specific data element attributes for database tables

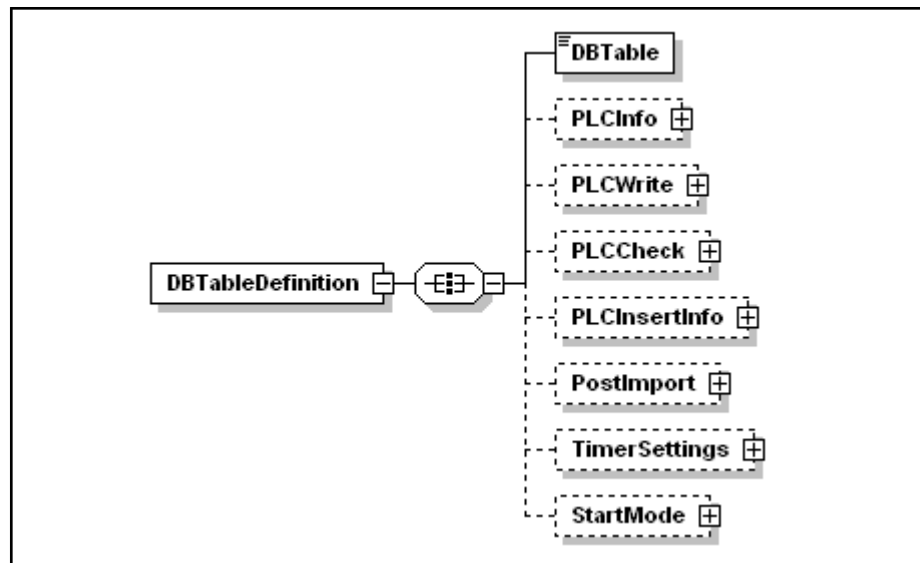
Increase definition

The maximum valid value increase for additive value input for the corresponding data element is specified by the data attribute **MaxIncIn** or **MaxIncEx**.

Defining the input type

The following selection is available via the **ETA** attribute when defining the input type:

	Without Only absolute value input possible IA Additive and absolute value input DI additive diameter input (absolute value input without diameter), i.e. input value / 2 is added
Defining data-related user rights	Use the data attributes L1 to L5 to assign data element-relevant read and write rights to each user level. These attributes can be defined for nodes and elements. The attribute definitions can structurally overlap. The definition closest to the element in the tree is valid.
Defining limit and enumeration values	Data element-relevant limit value and enumeration definitions are implemented by defining a basic type with corresponding restriction. Example: Limit value definition: $1 \leftarrow \text{double value} \leftarrow 10,000,000$ <i>Program:</i> <pre><xs:simpleType name="Double1_10M_t"> <xs:annotation> <xs:documentation>Type Double 1 - 10 Mio</xs:documentation> </xs:annotation> <xs:restriction base="xs:double"> <xs:minInclusive value="1"/> <xs:maxInclusive value="10000000"/> </xs:restriction> </xs:simpleType></pre> Example: Enumeration definition: $0 \leftarrow \text{byte value} \leftarrow 3$ with range of values: 1;2;3 <i>Program:</i> <pre><xs:simpleType name="Byte0_3_t"> <xs:annotation> <xs:documentation>Byte 0 - 3</xs:documentation> </xs:annotation> <xs:restriction base="xs:byte"> <xs:minInclusive value="0"/> <xs:maxInclusive value="3"/> <xs:enumeration value="0"/> <xs:enumeration value="1"/> <xs:enumeration value="2"/> <xs:enumeration value="3"/> </xs:restriction> </xs:simpleType></pre>
Checking input value via PLC	The following settings can be made in the file "ToolManagementConfig.xml" in the project directory. This file is structured as follows:



PLCCheck	Definition of the PLC variable name for the "Validate" function
PLCInfo	Definition of the PLC variable name for the "Info" function
PLCWrite	Definition of the PLC variable name for the "Write" function
Post import	Definition of the "Postimport" function
DBConnections	Activating multiple DB tables
TimerSettings	Low-pass filter configuration
StartMode	Setting the edit mode used for starting. "True" or "false" are used for the configuration. Note the spelling.

Fig. 7-74: Structure diagram 1: ToolManagementConfig.xml

The data attribute **PLCCheck** of the data schema can be used to activate the input value check in terms of tool-specific limit value exceedance in the PLC for every data element using the entry "Lim".

Example: Activation of limit value monitoring in the PLC in the data structure "Geo_V04_t" of the schema definition file "tool_ty.xsd":

Program:

```
<xs:element name="Rad">
  <xs:complexType>
    <xs:simpleContent>
      <xs:extension base="Double-10M_10M_t">
        <xs:attribute name="ETA" type="xs:string"
          use="optional" fixed="IA"/>
        <xs:attribute name="PLCCheck" type="xs:string"
          use="optional" fixed="Lim"/>
      </xs:extension>
    </xs:simpleContent>
  </xs:complexType>
</xs:element>
```



This function cannot be used for the bit arrays (BQ1-3) and the string elements (SKQ).

Sequence:

1. Value input in the editor or in the list.
2. After completing the input, by e.g. pressing <ENTER>, the interface (Act = TRUE) activates the PLC via the structure variable and transfers the entered value.
3. PLC checks the value to see if it exceeds the defined limit value.
4. PLC sends an acknowledgement to the interface (Act = FALSE).

The interface expects the acknowledgement by the PLC within 500 ms (5 times the 100 ms pattern). If this acknowledgement is missing, the value is not transferred to the database. An error message is then output in the status line.

Editing can only be closed when entering either a correct value or by pressing <ESC>.

5. In the case of positive feedback (result = 0) from the PLC, the interface writes the value change to the database. Otherwise, the interface outputs an error message in the status line and the value is not written to the database.

The PLC expects the following return values via the variable "RESULT" (applies to all following PLC functions):

Return Value	Description	Comment
<0	Reserved	Special error case
0	Value is OK	
1	Value was rejected by the PLC	Normal error without description of the cause
2	The selected place does not contain any tool	Special error case
3	The tool database is locked	Special error case
4-10	Reserved	
>10	The value was rejected by the PLC with number (e.g. tool edge number; correction value number)	Normal error case with description of the cause The error number is output to describe the cause



For all return values unequal to 0, the entered value is NOT applied. Negative return codes less than -1 may not be returned by the PLC, since these are used internally by the interface.

Structure of PLC variable:

TYPE stValidate:

STRUCT

Place:	INT;	(*Place*)
Storage:	INT;	(*Sector*)
Act:	BOOL;	(*Activation*)
XPath:	STRING(79);	(*Data element*)
Value:	LREAL;	(*Input value*)
EditType:	INT;	(*Additive / absolute - not relevant!*)
Result:	INT;	(*Error return*)

END_STRUCT

END_TYPE

The name of the PLC variable can be specified in the "ToolManagementConfig.xml" file under **PLCCheck/PLCVariableID** in the project directory.

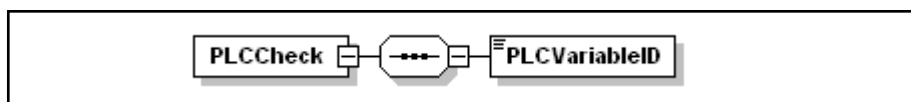


Fig. 7-75: Block diagram 2 ToolManagementConfig.xml

Program:

```

...
<PLCCheck>
  <PLCVariableID>.Validate</PLCVariableID>
</PLCCheck>
<PLCInfo>
  <PLCVariableID>.TLInfo</PLCVariableID>
</PLCInfo>
<PLCWrite>
  <PLCVariableID>.TLDataWrite</PLCVariableID>
</PLCWrite>
...

```

Message to PLC on value change in data block

The data attribute **PLCCheck** of the data schema can be used to activate the message of changes and the input value check for every data element to the PLC using the entry "Info".

Example: activation of PLC info function in the data structure "Geo_V04_t" of the schema definition file "tool_ty.xsd":

Program:

```

<xs:element name="Rad">
  <xs:complexType>
    <xs:simpleContent>
      <xs:extension base="Double-10M_10M_t">
        <xs:attribute name="ETA" type="xs:string"
          use="optional" fixed="IA"/>
        <xs:attribute name="PLCCheck" type="xs:string"
          use="optional" fixed="Info"/>
      </xs:extension>
    </xs:simpleContent>
  </xs:complexType>
</xs:element>

```



This function can also be applied for string elements (SKQ). However, no value is assigned to the variable "Value".

Sequence:

1. Value input in the editor or in the list.
2. Interface writes the value change to the database.
3. Interface activates the PLC via a structure variable and transfers the input value.
4. PLC can react accordingly to the messages from the user interface.

Structure of PLC variable:

TYPE stChangelInfo:

STRUCT

Place:	INT;	(*Place*)
Storage:	INT;	(*Sector*)
Act:	BOOL;	(*Activation*)
XPath:	STRING(79);	(*Data element*)
Value:	LREAL;	(*Input value*)
EditType:	INT;	(*Additive / absolute - not relevant!*)

Result: INT; (*Error return - not relevant!*)

END_STRUCT

END_TYPE

The name of the PLC variable can be specified in the "ToolManagementConfig.xml" file under **PLCInfo/PLCVariableID** in the project directory.

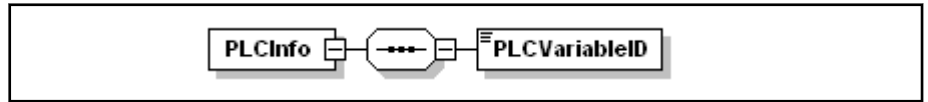


Fig. 7-76: Block diagram 3 ToolManagementConfig.xml

Program:

```
...
    <PLCCheck>
    <PLCVariableID>.Validate</PLCVariableID>
</PLCCheck>
<PLCInfo>
    <PLCVariableID>.TLInfo</PLCVariableID>
</PLCInfo>
<PLCWrite>
    <PLCVariableID>.TLDataWrite</PLCVariableID>
</PLCWrite>
...
```

Activating value change in data block by PLC

The data attribute **PLCCheck** of the data schema can be used to activate the change message and the input value check for every data element to the PLC via the entry "Write".

Example:

Activation of limit value monitoring in the PLC in the data structure "Geo_V04_t" of the schema definition file "tool_ty.xsd":

Program:

```
<xs:element name="Rad">
  <xs:complexType>
    <xs:simpleContent>
      <xs:extension base="Double-10M_10M t">
        <xs:attribute name="ETA" type="xs:string"
          use="optional" fixed="IA"/>
        <xs:attribute name="PLCCheck" type="xs:string"
          use="optional" fixed="Write"/>
      </xs:extension>
    </xs:simpleContent>
  </xs:complexType>
</xs:element>
```

Sequence:

1. Value input in the editor or in the list.
2. After completing the input, by e.g. pressing <ENTER>, the interface (Act = TRUE) activates the PLC via the structure variable and transfers the entered value.
3. PLC checks the value to see if it exceeds the defined limit value.
4. If the test is successful, the PLC writes the value change to the database.
5. PLC sends an acknowledgement to the interface.

PLC sends an acknowledgement to the interface (Act = FALSE).

The interface expects the acknowledgement by the PLC within 500 ms (5 times the 100 ms pattern). If this acknowledgement does not occur, the value is written from the interface to the database. An error message is then output in the status line.

6. The interface issues a message on the success or failure of the writing procedure in the status line with regard to the value of the variable "RESULT".

Structure of PLC variable:

TYPE stTLDataWrite:

STRUCT

Place:	INT;	(*Place*)
Storage:	INT;	(*Sector*)
Act:	BOOL;	(*Activation*)
XPath:	STRING(79);	(*Data element*)
Value:	LREAL;	(*Input value*)
EditType:	INT;	(*Additive / absolute*)
Result:	INT;	(*Error return*)

END_STRUCT

END_TYPE

The name of the PLC variable can be specified in the "ToolManagementConfig.xml" file under **PLCWrite/PLCVariableID**.

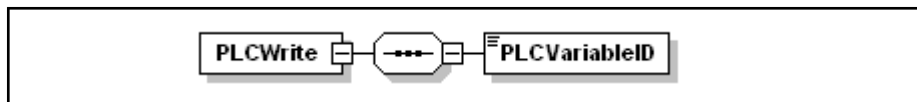


Fig. 7-77: Block diagram 4 ToolManagementConfig.xml

Program:

```
...
<PLCCheck>
  <PLCVariableID>.Validate</PLCVariableID>
</PLCCheck>
<PLCInfo>
  <PLCVariableID>.TLInfo</PLCVariableID>
</PLCInfo>
<PLCWrite>
  <PLCVariableID>.TLDataWrite</PLCVariableID>
</PLCWrite>
...
```



This function cannot be used for string elements (SKQ).

Message to the PLC on inserting, deleting or moving a data block

The function can be activated and the name of the PLC variable can be specified in the "ToolManagementConfig.xml" file under **PLCInsertInfo/PLCVariableID**.

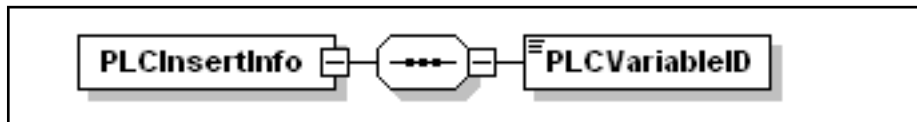


Fig. 7-78: Block diagram 5 ToolManagementConfig.xml

Program:

```
...
<PLCCheck>
  <PLCVariableID>.Validate</PLCVariableID>
</PLCCheck>
<PLCInfo>
  <PLCVariableID>.TLInfo</PLCVariableID>
</PLCInfo>
```

```

    <PLCInsertInfo>
    <PLCVariableID>.TLDeleteMoveInsertInfo</PLCVariableID>
  </PLCInsertInfo>
...

```

Structure of PLC variables

TYPE stInsertTool:

STRUCT

SrcPlace :	INT;	(*Place - source*)
SrcStorage :	INT;	(*Sector - source*)
DstPlace :	INT;	(*Place - destination*)
DstStorage :	INT;	(*Sector - destination*)
Act:	BOOL;	(*Activation*)
Value:	INT;	(*Functional ID*)

END_STRUCT

END_TYPE

Function ID:

Value	Function
1	Tool was inserted
2	Tool was deleted
3	Tool data blocks were imported
4	Tool was displaced

Sequence for tool list import (inserting/overwriting several data blocks - no single tool import):

1. Activating and executing tool list import
2. Interface activates PLC via structure variable:
 - SrcPlace = 0 (not relevant)
 - SrcStorage = 0 (not relevant)
 - DstPlace = 0 (not relevant)
 - DstStorage = 0 (not relevant)
 - Act = 1
 - Value = 3
3. PLC sets Act = 0
4. PLC **can** react accordingly to the message from the user interface.

Sequence when inserting or copying a tool or importing a single tool:

1. Activating and executing tool import or the copy or insert function
2. Interface activates PLC via structure variable:
 - SrcPlace = n (n = number of the place on which the insertion took place)
 - SrcStorage = m (n = number of the sector into which the insertion took place)
 - DstPlace = 0 (not relevant)
 - DstStorage = 0 (not relevant)

- Act = 1
 - Value = 1
3. PLC sets Act = 0
 4. PLC **can** react accordingly to the message from the user interface.

Sequence during tool deletion:

1. Activating and executing tool deletion function
2. Interface activates PLC via structure variable:
 - SrcPlace = n (n = number of the place on which the deletion took place)
 - SrcStorage = m (n = number of the sector in which the deletion took place)
 - DstPlace = 0 (not relevant)
 - DstStorage = 0 (not relevant)
 - Act = 1
 - Value = 2
3. PLC sets Act = 0
4. PLC **can** react accordingly to the message from the user interface.

Sequence while moving tool:

1. Activating and executing tool displacement function
2. Interface activates PLC for the first time via structure variable:
 - SrcPlace = k (k = number of the place on which the deletion took place)
 - SrcStorage = l (l = number of the sector in which the deletion took place)
 - DstPlace = n (n = number of the place to which insertion occurred)
 - DstStorage = m (n = number of the sector into which the insertion took place)
 - Act = 1
 - ID = 4
3. PLC sets Act = 0
4. PLC **can** react accordingly to the message from the user interface.

Communication between user interface and PLC during insertion

The PLC can be informed about the change of data elements while inserting a tool data block via the entry **PLCInsertDataChange/PLCVariableID** in the file "ToolManagementConfig.xml".

During this process, PLC and interface correspond next to the PLC variable structure to exchange the handshake signals via the system data structure "**SD.SysToolInsert**". This system variable structure is of type "DBT1Rec_t" and thus corresponds to the a complete tool data block.

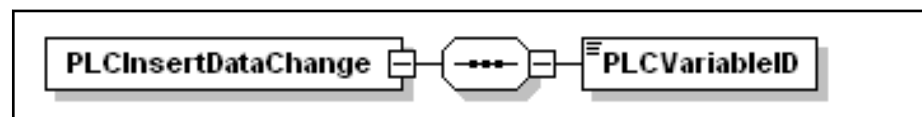


Fig. 7-79: Block diagram 6 ToolManagementConfig.xml

Program:

```
... <PLCCheck>
```

```

        <PLCVariableID>.Validate</PLCVariableID>
    </PLCCheck>
    <PLCInfo>
        <PLCVariableID>.TLInfo</PLCVariableID>
    </PLCInfo>
    <PLCInsertDataChange>
        <PLCVariableID>.TLInsertDataSet</PLCVariableID>
    </PLCInsertDataChange>
...

```

Structure of PLC variables

TYPE stlInsertDataChange:

STRUCT

Place:	INT;	(*Place*)
Storage:	INT;	(*Sector*)
Act:	BOOL;	(*Activation*)
XPath:	STRING(79);	(*Data element*)
Value:	LREAL;	(*Input value*)
EditType:	INT;	(*not relevant*)
Result:	INT;	(*Error return*)

END_STRUCT

END_TYPE

Procedure in case of offline editing of a tool data block to be inserted

1. Applying the data block of the insert position from the database to SD.SysToolInsert.
2. Interface activates PLC via structure variable:
 - Place = n (n = number of the place on which the insertion took place)
 - Storage = m (n = number of the sector into which the insertion takes place)
 - Act = 1
 - XPath = " (not relevant)
3. PLC sets Act = 0
4. Change in the editor
5. Applying change in SD.SysToolInsert
6. Message to PLC
 - Place = n (n = number of the place on which the insertion took place)
 - Storage = m (n = number of the sector into which the insertion takes place)
 - Act = 1
 - XPath = Modified data element
7. PLC reads SD.SysToolInsert
8. PLC modifies SD.SysToolInsert
9. PLC sets Act = 0
10. Interface reads modified variable SD.SysToolInsert
11. Applying data block to the database as completion of inserting process

Post-Import Function

In addition to the variable name definition described above, post-importing is configured using the file "ToolManagementConfig.xml".

If an action entry is defined in this file, a post-import function automatically follows a list or data block import.

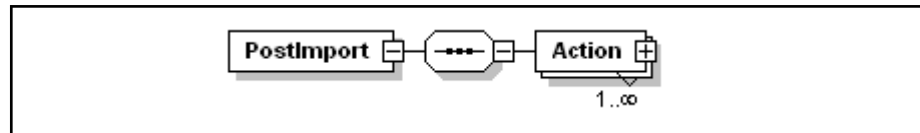


Fig. 7-80: Structure diagram 7: ToolManagementConfig.xml

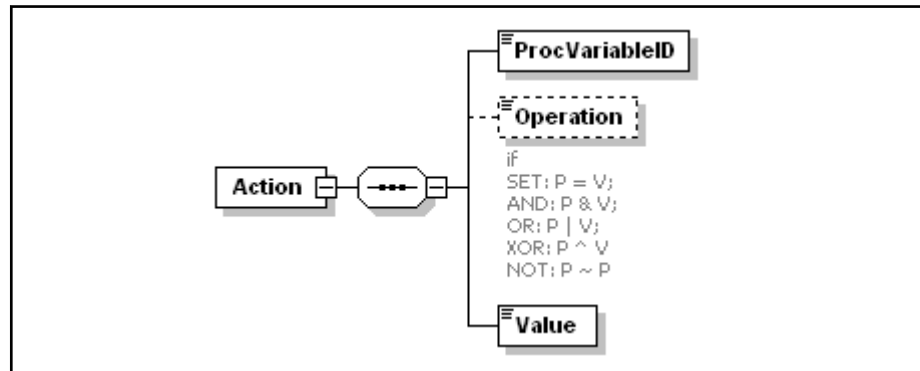


Fig. 7-81: Structure diagram 8: ToolManagementConfig.xml

Explanation of abbreviations:

P:	Process variable	(ProcVariableID)
V:	Value	(Value)

Permitted operations:

SET:	Process variable receives value from "Value"
AND:	The process variable becomes log. AND-gated with value of "Value"
OR:	The process variable becomes log. AND-gated with value of "Value"
XOR:	The process variable becomes log. AND-gated with value of "Value"
NOT:	The value of the process variable is negated

Tab. 7-17: Abbreviations and permitted operations

Data import procedure with active post-import function:

1. Import of tool list or tool data block
2. Apply the post-import function on all imported data blocks

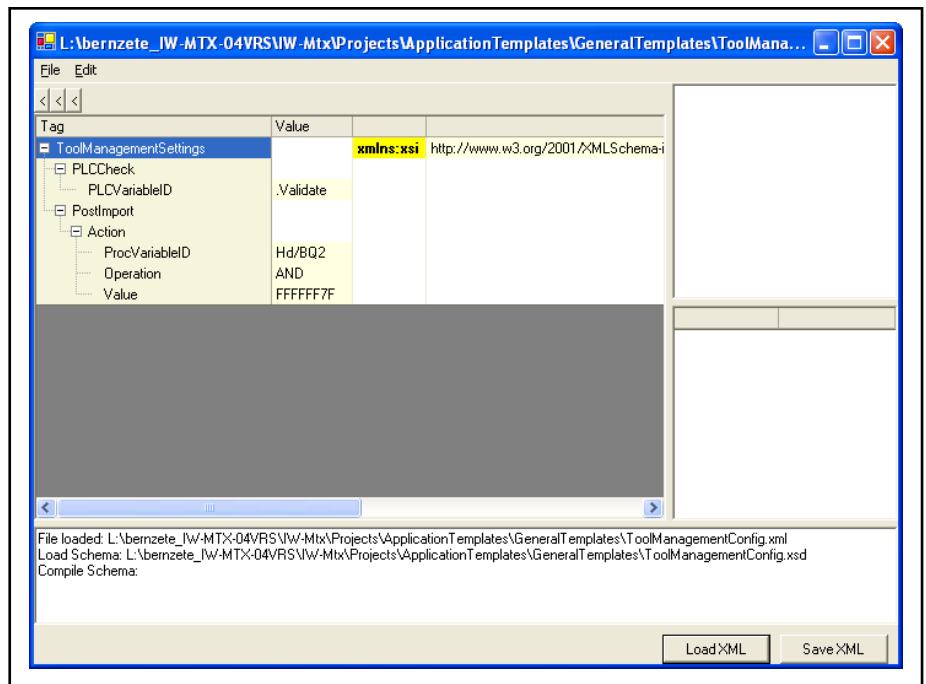


Fig. 7-82: Configuring basic tool management settings

Example:

ProcVariableID	Hd/BQ2
Operation	AND
Value	FFFFFF7F

In the example, the "Tool broken" flag (TD) is reset in all imported data blocks after the tool list import.

Activating second database table

File "ToolManagementConfig.xml" is also used to activate the communication with the second database table.

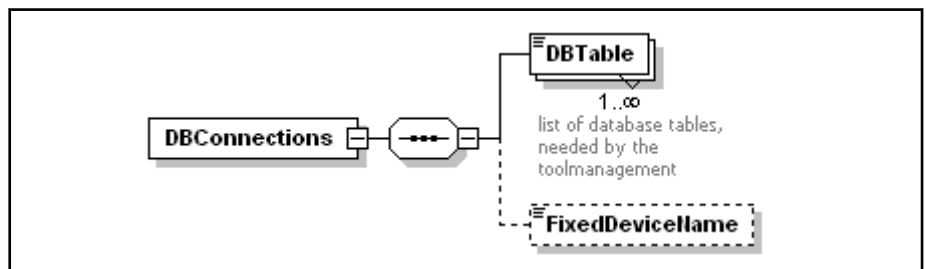


Fig. 7-83: Activating second database table

Program:

```
<DBConnections>
  <DBTable>/DBT1</DBTable>
  <DBTable>/DBT2</DBTable>
</DBConnections>
```

If these entries are missing, it is only communicated with database table 1.

If the database table is also to be displayed in IW Engineering as separate configuration node, this has to be configured by adding the following entry before creating the project in the file **MTX.MnoApplicationConfig.xml** in the runtime directory **\Library\Devices\Visualization\ApplicationTemplates\Visualization** device (e.g. BT40).

Program:

```

...
<HmiMnO ID="MTX DB SCREENS" Text="Pallet Screens" Category="MTXPALLETScreens"
Assembly="MTX.Toolman.MnOPackage"
Factory="Rexroth.MTX.Toolman.MnOPackage.ToolmanMnOHandlerFactory"/>
...

```

Low-pass filter for change events

To prevent blocking the communication by, for example, the PLC if the database is modified several times in sequence very quickly, a low-pass filter for change events can be activated and dimensioned. This is also performed using the file "ToolManagementConfig.xml".

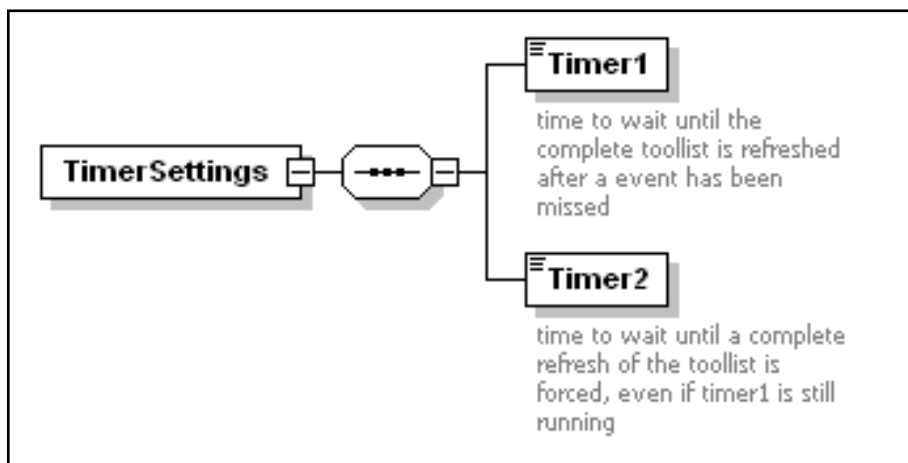


Fig. 7-84: Low-pass filter for change events

Timer 1:

This timer is started if it is determined that a change event was lost. If another event is lost during this time, the timer is restarted and tool data is not read. Otherwise, all data in the database table is read after the timer elapses.

Timer 2:

If timer 1 never counts down completely to subsequently read all data, a forced reading of all data takes place after timer 2 elapses. Therefore, timer 2 is activated when timer 1 is started for the first time and stopped each time after all data is read in the database.

Timer 1:	(small timer)	e.g.: 500 ms
Timer 2:	(large timer)	e.g.: 20 s

Setting change events to be buffered in the control

The tool list display is informed via changes in the database tables to visualize the current state. Due to performance reasons, every single change is identified by the user interface.

In case of a fast sequence of writing operations by the PLC or CPL, the user interface can only be informed completely if a sufficient number of change events is buffered in the MTX.

The number of change events to be buffered can be defined via the configuration parameter "/NCO/DBTables/DBEventBuffSize".

Default values for data blocks

When creating data blocks, deleting their contents and displacing them, all data elements are initialized by default with 0 or an empty string. If other default values are desired, create one of the database-specific files "dbt1dat.xml" or "dbt2dat.xml". This file has to contain a data block with the

default values of the respective database table and has to be located in the "\usrfep".

Handling instruction: Modifying data block schema

The following handling instruction describes the process to be followed when the data block of a database table is to be modified.

W-Operation/program: Edit the Schema File "dbt?ud.xsd"

1. Copy the file "dbt?ud.xsd" (?:= 1 [DBT1] or 2 [DBT2]) to the control directory "\usrfep\schemas" or "\feprom\schemas" or "\root\schemas" to the mount directory (\mnt).
2. Edit file with schema editor.
3. Save file.
4. Copy the file back to the control directory "\usrfep" or "\root".

Screen		Documentation
Screen		Screen: Data block schema
Documentation	MTX Functional Description	Edit schema file

NC: Apply new data structure

1. Complete IW Operation
2. Reset control

		Documentation
Documentation	MTX Functional Description	Apply new data structure

Handling instruction: Modifying table-specific data element-relevant limit values

The following handling instruction describes the process to be followed when the limit values for the individual data elements of a tool data block (DBT1, DBT2) are to be modified.

IW-Operation/program: Edit the Schema File "dbt?ud.xsd"

1. Copy the file "dbt?ud.xsd", "dbt?sd.xsd" (?:= 1 [DBT1] or 2 [DBT2]) or "tool_ty.xsd" to the control directory "\usrfep\schemas" or "\feprom\schemas" or "\root\schemas" to the mount directory (\mnt).
2. Value range of the corresponding elements via
`<xs:minInclusive value = ".."/>`
`<xs:maxInclusive value = ".."/>`
3. Save file.
4. Copy the file to the control directory "\usrfep\schemas" or "\root\schemas".

Screen		Documentation
Screen		Screen: Data block schema
Documentation	MTX Functional Description	Edit schema file

NC: Apply new data structure

1. Complete IW Operation
2. Reset control

		Documentation
Documentation	MTX Functional Description	Apply new data structure

Handling instruction: Activating tool-specific limit value monitoring in user interface

The following handling instruction describes the procedure for activating tool-specific limit value monitoring for the individual data elements in the editors of the user interface.

IW-Operation/program: Edit the Schema File "dbt?ud.xsd"

For this function, the limit value data elements have to be contained in the tool data block.

1. Call the tool list configuration in the current IndraWorks Engineering project
2. "Show XML"
3. In the XML editor, search for the corresponding CellDef entry for the data element to be edited using limit value monitoring.
4. Search/insert: "RepresentationDefinitions"
5. Search/insert: "Dependency_for_Representation"
6. Search/insert: "Validation"
7. Search/insert: "DepProcessVariableID"
8. In the "Value" column, define the process variable containing the limit value.
9. Search/insert "Operation"
10. In the "Value" column, define the test condition (for the help, see the "Properties" window)
11. If further test conditions are to be defined: Search/insert "FollowingDep-Condition" value = 1
12. Search/insert "Condition"
13. Continue with step 7 if the condition under step 11 is fulfilled.
14. Save file.

Screen		Documentation
Screen		Screen: Data block schema (3) (header schema)
Documentation	MTX Functional Description	Tool-specific limit values

User interface: Call the modified list display again

Screen		Documentation
Documentation	MTX Functional Description	Tool-specific limit values

Handling instruction: Creating a file with default values for database table

This handling instruction can be used to create a new file with default values for a database table.

IW Operation / tool management: Create "dbt?dat.xml" file

1. Enter the desired default values into any data block of the respective database table

- Export this data block into the file **"dbt?dat.xml"** (?:= 1 [DBT1] or 2 [DBT2]) into the mount directory (\mnt).

		Documentation
Documentation	MTX Functional Description	Create dbt?dat.xml file

IW-Operation/program: Adapt "dbt?dat.xml" file

- Open file in editor.
- Replace content of data element SKQ with an empty string.
- Save file.
- Copy the file to the **"usrfep"** control directory.

		Documentation
Documentation	MTX Functional Description	Adapt dbt?dat.xml file

IW-Operation/program: Modify the file "dbt?prms.dat"

- Open file in editor.
- Set P1 = 6
- Save file.
- Copy the file to the **"usrfep"** control directory.

		Documentation
Documentation	MTX Functional Description	Modify the file "dbt?prms.dat"

Program:

```

; -----
;           Configuration file for  DB table DBT1
; -----
; General structure:
; P0 P1 P2 P3 P4 P5 P6
; |  |  |  |  |  |  |
; |  |  |  |  |  |  +--- string: may also contain " " and \t
; |  |  |  |  |  |  End ID is "end of line"
; +---+---+---+---+---+ 6 int values
; P0 describes the object type and may not be modified
; Empty string is marked by "."
; ";" marked comment lines. At the end of the data line
; no comment may be added
;
; *****
;           Configuration of DB table
; *****
; Control parameter
; P0: 0
; P1: Controls data record creation in the database
;     0 Normal mode
;     Only if tool database is empty, data records are
;     created
;     6 First, all data records are deleted
;     and subsequently, created again according to the sector
;     configuration
0 6
; Sector configuration
; P0: 1-99 (corr. to sector 1 - 99)
; P1: Number of places in the sector
1 5
2 5

```

NC: Apply default values

- Complete IW Operation
- Reset control

		Documentation
Documentation	MTX Functional Description	Apply default values

IW-Operation/program: Undo the modification of the file "dbt?prms.dat"

1. Open file in editor.
2. Set P1 = 0
3. Save file.
4. Copy the file to the "**usrfep**" control directory.

		Documentation
Documentation	MTX Functional Description	Modify the file dbt?prms.dat

7.10.5 Tool catalog

Definition of terms and general explanations

When working with the MTX tool management, the tool catalog has the following tasks:

- Tool type management
- Basic data management

It follows that two types of data management can be differentiated:

Tool type management: ToolCatalog.xml

Basic data management: ToolBDSCatalog.xml

The following image explains the terms for the following variants:

- Tool technology
- Tool type
- Tool class
- Alternate tool
- Master data block
- Basic data block

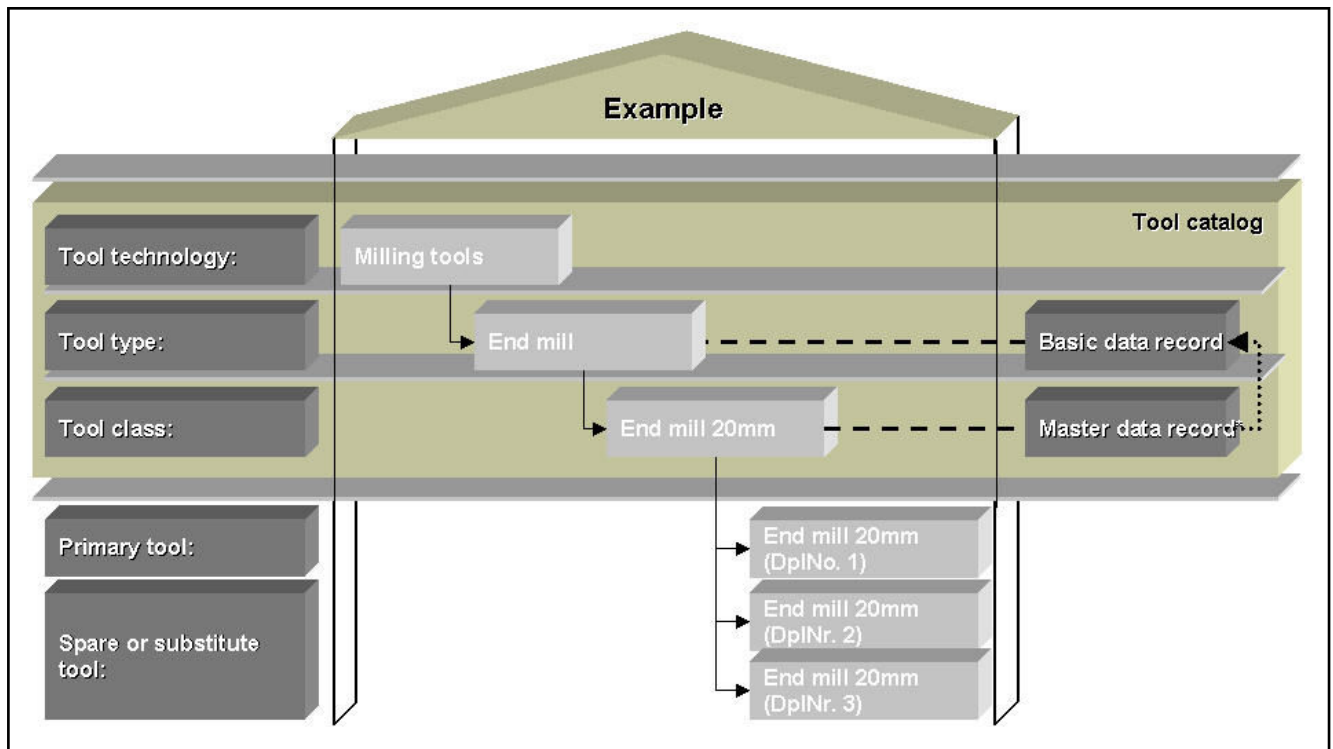


Fig. 7-85: Explanation of tool catalog terms

Tool Technology Comprises all tool types belonging to a machining technology.
The tool catalog includes the following machining technologies:

- General machining
- Drilling
- Milling
- Rotate
- Measurement
- Special processing
- Punching

Tool Type Describes the total of all tool classes with identical type-defining properties. The individual tool classes of a tool type differ by the tool ID. Together with the tool technologies, the tool types form the basis of the tool catalog and, thus, the basis of master tool data management. Furthermore, bitmaps are provided to go with the tool types in the tool catalog.

Tool Class The amount of all tools with the same ID is called tool class. The individual tools of a class differ by their Duplo number.

Replacement or Alternate Tool A replacement tool belongs to a tool class and can replace the respective primary tool if necessary. It is, for example, not blocked or worn.

Primary Tool A primary tool is the tool of a tool class which will be used for machining at the next call-up via the T word. All other tools of the tool class are called secondary tools.

Basic Data Block The basic data block exists exactly once for each tool type when a tool has already been defined for this type and when a basic data block has been explicitly created in this context. If an additional tool (tool class) of this type is created, the basic data block is used to initialize the data block.

Master Data Block (in Preparation) The master data block exists exactly once for each tool class when a tool has already been defined for this class. If another tool (alternate tool) is created, the master data block is used for data block initialization. If a tool is created for a class for the first time, the data block of this tool is turned into the master data block for the tool class.



Master data management is still in preparation.

Predefined tool types (Standard Types)

Currently, the user can access a collection of pre-defined tool types when creating the tool data for machining technologies:

- Drilling
- Milling
- Rotate
- Punching
- Measurement
- Special processing

These tool types are stored in the tool catalog and are characterized by the following tool properties:

Designation of the element in the tool catalog		Description	Element in the DS header
TypeNo		Type number	IKQ2
TypePic		Name of the bitmap file	-
CoTe (bits 1 - 4)	EdNo	Edge number	BQ3
Relevance of the correction values (correction type):			BQ3
CoTe (bits 6 -15)	CVR	Length reference of ball center	
	CH	Angle head tool	
	L1	1st length correction value	
	L2	2nd length correction value	
	L3	3rd length correction value	
	R1	big radius	
	R2	small radius	
	O	Edge position	
	DIA	Tool with diameter information	

Designation of the element in the tool catalog		Description	Element in the DS header
Relevance of the technology data:			BQ3
CoTe (bits 15 - 24)	T1	d or tw	
	T2	ds	
	T3	lfc or ll or tw or tc	
	T4	tl	
	T5	lu	
	T6	tp or ta or tal	
	T7	tac	
	T8	nt	
	ET1	de	
	ET2	lfe	

Tab. 7-18: Type-defining tool data

The technology data bits T1 to T8 refer to the relevance of the tool technology data and ET1 and ET2 of the edge technology data in the standard tool data block.

As can be seen from the table, the technology data in the standard tools can have a different meaning subject to the tool type:

Technology data in standard tools

Data element	Abbreviation	Description
T1	d	Diameter
		Large diameter
		Milling diameter
		Outer diameter
		Outer limitation of the cutting diameter area
		Edge radius
		Nominal diameter of thread
		Diameter dimension 1
	tw	Width of trapezoid thread
T2	ds	Small diameter
		Shaft diameter
		Drilling diameter
		Smallest diameter that can be machined
		Inner limitation of the cutting diameter area
		Countersinking diameter
		Diameter dimension 2

Control commissioning

Data element	Abbreviation	Description
T3	lfc	Chamfer length
		Length of tool tip
	ll	Loss length
		Spigot length
	tw	Cutting edge width
	tc	Corner radius
		Tool tip size
		Sphere radius
		Width dimension
T4	tl	Max. cutting height
		Max. thread height
		Depth gauge
T5	lu	Usable length
		Length dimension
T6	tp	(Thread) lead
	ta	Setting angle
	tal	Orientation angle
		Angle (edge angle)
		Tool setting angle
		Angle (kappa)
		(General) Angle
		Opening angle
		Angular dimension 1
T7	tac	Apical angle
		Countersink angle
		Corner angle
		Edge angle
		Angular dimension 2
T8	nt	Number of teeth
ET1	de	Edge radius
		Small diameter
		Cut length
		Edge dimension 1
ET2	lfe	(General) Length
		Edge dimension 2

Tab. 7-19: Significance of the technology data for standard tool types

Significance of the technology data for user-defined tool types is limited and specified as follows:

Technology data for user tools

Data element	Abbreviation	Description
T1	d	Diameter dimension 1 (large)
T2	ds	Diameter dimension 2 (small)
T3	tc	Width dimension
T4	tl	Depth gauge
T5	lu	Usable length
T6	tal	Angular dimension 1
T7	tac	Angular dimension 2
T8	nt	Number of teeth
ET1	de	Edge dimension 1
ET2	lfe	Edge dimension 2

Tab. 7-20: Significance of the technology data for user-defined tool types

For users who wish to use only a part or none of the tool catalog functions, the tool type "general tool" with the machining technology "general Machining" as well as a general drilling, milling or turning tool with a variable tool edge number for the individual technologies has been introduced. Furthermore, the two "UsedFlag" data elements can be used to deselect both individual types and technologies in the tool catalog. Then, these are not provided for selection anymore in the tool definition.



If a tool data block is created using the PLC or CPL with reference to the tool catalog, data elements IKQ2 and BQ3 must be written in accordance with the catalog definition. The "SD.SysToolType" system data provides the relevant values.

Create Data Block via PLC

If the PLC is to create tool data blocks which are compatible with the default tool catalog - e.g. by means of a Balluff BIS identification system - and which are to be entered into the database, it is recommended to determine the corresponding value pairs IKQ2 - BQ3 via a prepared system data structure "SD.SysToolType" and the corresponding function block and to enter them in the tool data block.

This system data is structured as follows:

SD.SysToolType structure

TypeNo	IKQ2 value
SysToolType[n]. Edge_variable	Signifies if the number of edges is fixed or can be edited
BQ3_Basic	BQ3 value

n: Number of tools in the tool catalog

Tab. 7-21: SD.SysToolType

This system data contains the value pairs for IKQ2 and BQ3 for all tools in the tool catalog.

If the user creates user-defined tools in the tool catalog, the tools also have to be entered in the system data. See **Saving system data** in [chapter "Functions of the tool type editor" on page 248](#).

Visualizing and configuring the tool catalog

General information

In MTX tool management, the user is provided with a default tool catalog which the user can extend by user-defined tool types. It is also possible to deactivate default tool types or tool technologies which are not required.

To configure the tool catalog, a tool type editor is available in IndraWorks Engineering.

Calling the tool type editor

The tool type editor is called under the node **DBT1 Screens (Tool management) ► Tool Catalog ► Open** in IndraWorks Engineering.

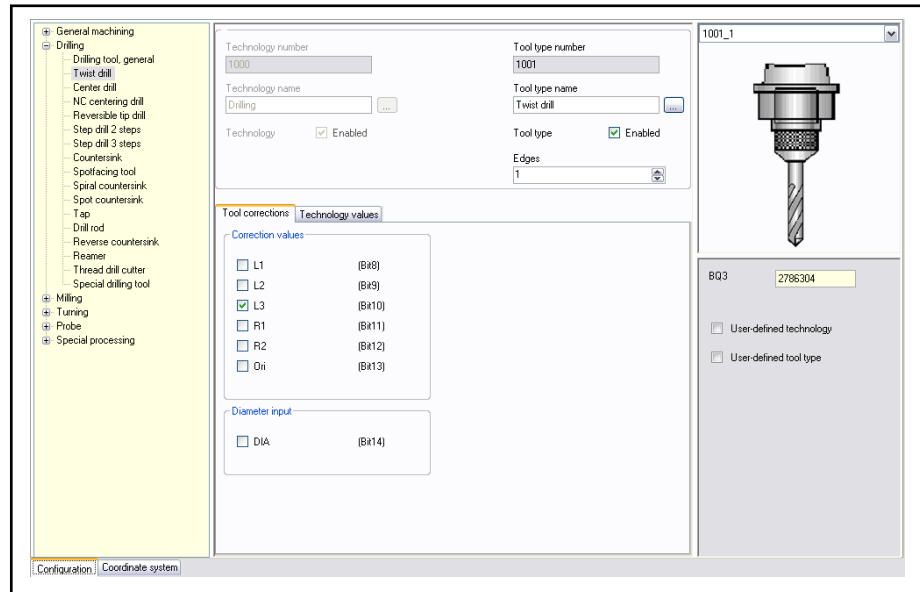


Fig. 7-86: Tool type editor

Text definitions in the tool type editor

Texts can be defined as follows in all configuration dialogs of the tool management:

- Resource text
- User text
- Fixed text

Functions of the tool type editor

Defining tool technologies

The dialog to define a tool technology is displayed via **New ► Technology** in the application window. The first available technology number is used automatically. In the second step, the technology name has to be defined.



Fig. 7-87: Dialog window to define a tool technology

Creating a new tool type

A technology for the selection in the tool editor can be enabled or disabled via the "Enabled" drop-down menu.

The dialog to define a tool type is displayed via **New ► Tool type** in the application window. The first available tool type number is used automatically in the technology group. In the second step, the tool identifier has to be defined.

Fig. 7-88: Dialog window to define a tool type

Defining/displaying tool corrections

A tool for the selection in the tool editor can be enabled or disabled via the "Enabled" drop-down menu.

Furthermore, the number of edges can be selected in a user-defined tool.

The relevance of the different correction values can be defined in the dialog window with several tabs:

Fig. 7-89: Displaying/defining the used correction data

The relevant correction data is displayed for all predefined standard tool types. However, they cannot be changed.

Defining/displaying technology values

The second page of the setup dialog is used to specify the relevance of the predefined technology data:

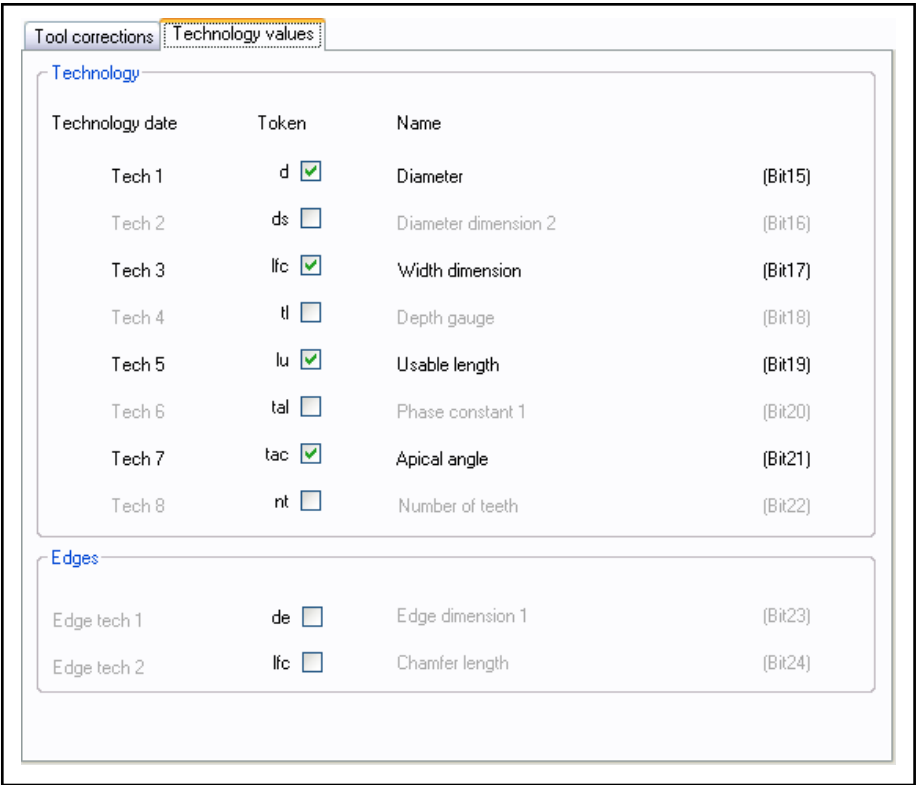


Fig. 7-90: *Displaying/defining the used technology data*

The relevant technology data and its description is displayed for all predefined standard tool types. However, they cannot be changed.

Editing tool images

New tool images are selected and defined via the menu function **Image ▶ New ▶ 1 / 2 / 3**, depending on the use in the tool editor.

Tool image types

Image type	Display in tool editor during active window:
1	"Basic tool data" or "tool data" or "tool status"
2	"Edge data" or "Edge data - Geometry limit values"
3	"Technology data"

Tab. 7-22: *Display of image types in the tool editor*

The following dialog is displayed to define an image file

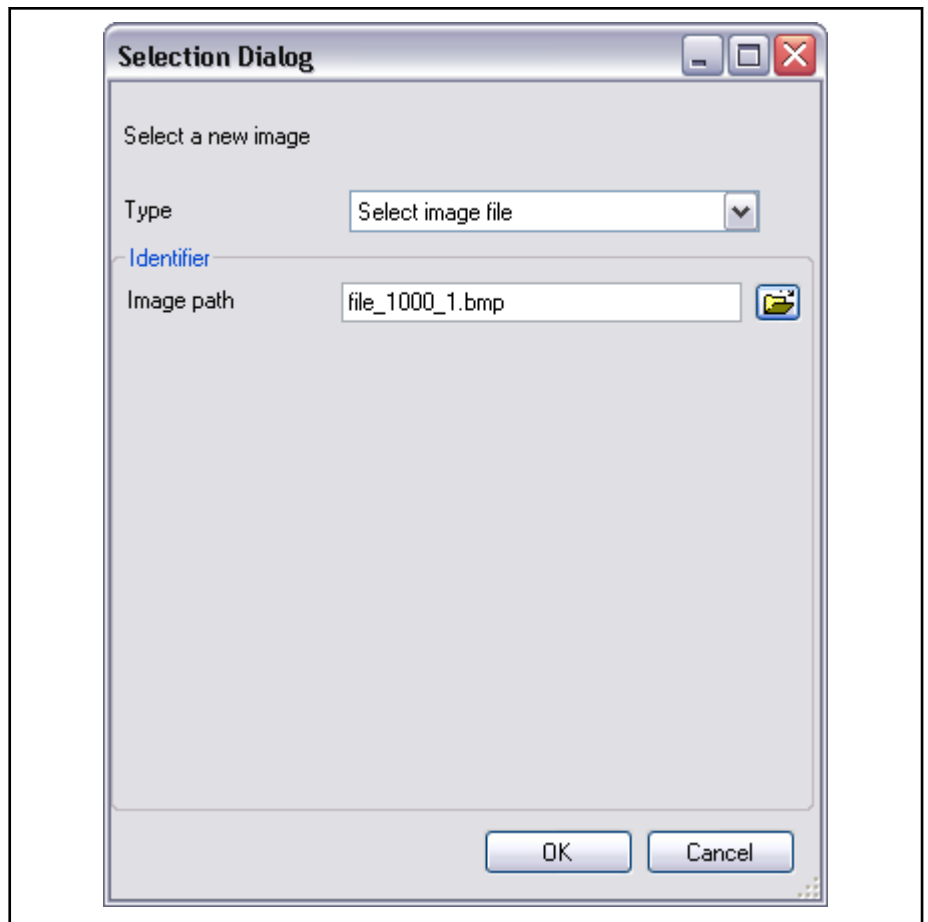


Fig. 7-91: Drop-down menu to define a tool image

Graphic files of type "*.bmp" or "*.gif" as well as image resource files can be used as source for user tool images.



Images are generated again and saved in the visualization project, complying with the specified naming convention.

Save system data

Refresh the system data structure **"SD.SysToolType"** after the user tool types have been defined.

This system data contains the corresponding value pairs for IKQ2 and BQ3 for all tools in the tool catalog and is the requirement to generate tool data blocks by the PLC, compatible with the tool catalog, e.g. by means of a Baluff BIS identification system.

This is done via the menu function **Save System Data** on the "Tool Catalog" node



Prerequisite for this function: the control device was switched online in IndraWorks Engineering.

Note that it has to be changed from offline status to online status after opening the tool type editor.

This system data contains the value pairs for IKQ2 and BQ3 for all tools in the tool catalog as well as information on whether the number of edges is variable as per the definition.

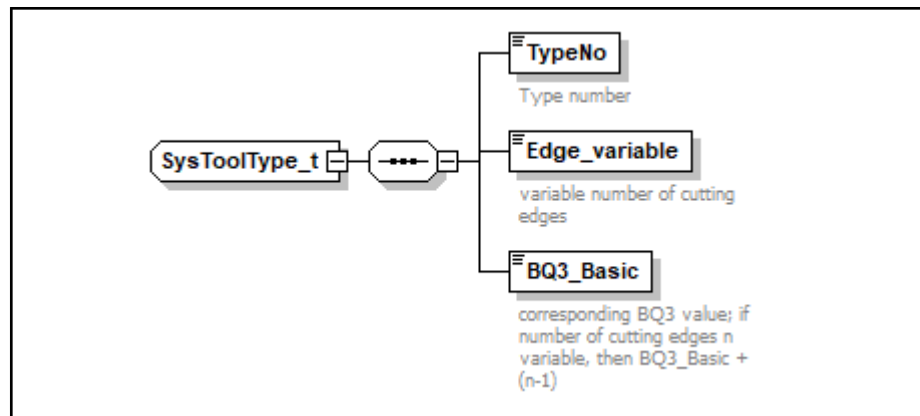


Fig. 7-92: Data scheme of the system variables $SD.SysToolType[n]$ [$n = 1 \dots$ Number of tool types]

Adjust BQ3 value

After the tool type definitions have been changed, old, exported data blocks are sporadically not compatible anymore. In these cases, the error message "Incorrect type definition" is output in the tool editor.

This function facilitates the adjustment of the BQ3 value of exported tool data blocks (.tld files) to the changed tool catalog.

A tool data export file of type "*.tld" can be selected via the "Search File" dialog in the file system of the control. Subsequently, the BQ3 values are automatically refreshed in the data blocks.



Prerequisite for this function: the control device was switched on-line in IndraWorks Engineering.

Note that it has to be changed from offline status to online status after opening the tool type editor.

Initializing the basic data block

In case of initial use of a tool type, a basic tool data block is generated for this type by the tool editor. A basic tool data block is used to initialize the user data of the tool data block upon creation of a tool of the same type. See **Basic data management** in [chapter 7.10.5 "Tool catalog" on page 242](#).

This function is used to generate basic data block structures for all tool types in the tool catalog. Subsequently, the basic data blocks can be prepared for the user by editing the "ToolBDSCatalog.xml" file in the visualization project using an XML file editor.

The following prompt illustrates the function.

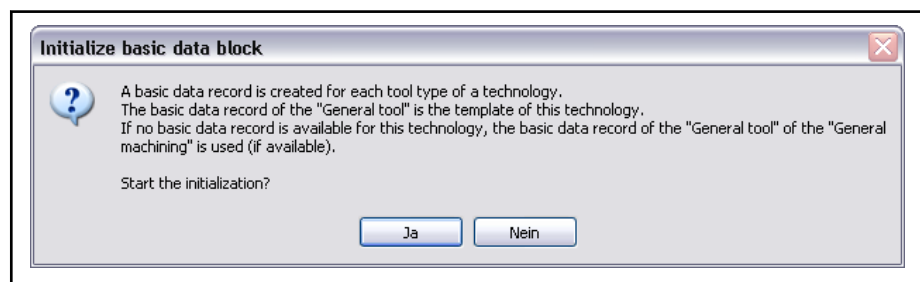


Fig. 7-93: Initializing the prompt: Basic data block

Defining the coordinate system display of the tool editor

It is possible to store the tool screen for a better illustration in a coordinate system.

On the "Coordinate system" tab, the configuration of the bit map display in the tool editor is supported.



Prerequisite for this function: the control device was switched on-line in IndraWorks Engineering.

Note that it has to be changed from offline status to online status after opening the tool type editor.

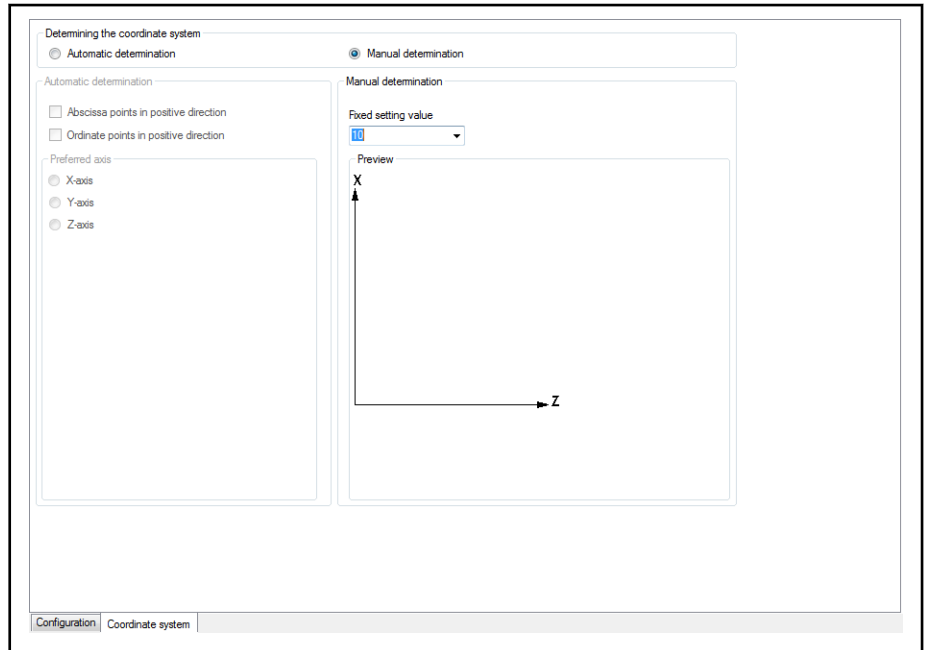


Fig. 7-94: Definition of the displayed coordinate system in the tool editor

This display is controlled via the system variable **SD.SysCoordSystem**:

SD.SysCoordSystem.	Value	Description
Value	See table	
Calc	1	Automatic determination of the coordinate system
	0	No automatic determination of the coordinate system
Horizontal	1	Abcissa in positive direction
	0	Abcissa in negative direction
Vertical	1	Ordinate in positive direction
	0	Ordinate in negative direction
X_axis_pref	1	The x-axis is the axis preferred to display the coordinate systems for milling tools
Y_axis_pref	1	The y-axis is the axis preferred to display the coordinate systems for milling tools
Z_axis_pref	1	The z-axis is the axis preferred to display the coordinate systems for milling tools

Tab. 7-23: Meaning of SD.SysCoordSystem

If no automatic determination of the coordinate system is set (SD.SysCoordSystem.Calc = 0), the stored coordinate system can be set as follows via the variable **SD.SysCoordSystem.Value**:

Control commissioning

SD.SysCoordSystem	Abscissa	Ordinate
	Direction	Direction
0	-	-
1	-	X
		Positive
2	-	X
		Negative
3	-	Y
		Positive
4	-	Y
		Negative
5	-	Z
		Positive
6	-	Z
		Negative
10	Z	X
	Positive	Positive
11	Z	X
	Negative	Positive
12	Z	X
	Negative	Negative
13	Z	X
	Positive	Negative
20	Y	X
	Positive	Positive
21	Y	X
	Negative	Positive
22	Y	X
	Negative	Negative
23	Y	X
	Positive	Negative
30	X	Y
	Positive	Positive
31	X	Y
	Negative	Positive
32	X	Y
	Negative	Negative
33	X	Y
	Positive	Negative

SD.SysCoordSystem	Abscissa	Ordinate
	Direction	Direction
40	Z	Y
	Positive	Positive
41	Z	Y
	Negative	Positive
42	Z	Y
	Negative	Negative
43	Z	Y
	Positive	Negative
50	X	Z
	Positive	Positive
51	X	Z
	Negative	Positive
52	X	Z
	Negative	Negative
53	X	Z
	Positive	Negative
60	Y	Z
	Positive	Positive
61	Y	Z
	Negative	Positive
62	Y	Z
	Negative	Negative
63	Y	Z
	Positive	Negative

Tab. 7-24: Control of the stored coordinate system via SD.SysCoordSystem.Value



After changing the setting, back up the control data, as the change does not affect the initialization data in the SDDat.xml file.

Basic data management

General information

In the insertion editor, the "Store basic tool data" <F5> function can be used any time to create a basic data block for the currently selected tool type, or to overwrite an existing basic data block. If another tool of this type has been created later, the basic data block is used for data block initialization. All basic data blocks are stored in the "ToolBDSCatalog.xml" file.

The basic data block only comprises the user data of the tool data block.

Initializing the tool data block when creating a tool

Array/element	Identifier	Description	Source of initialization
1	K1	Storage	No initialization
2	K2	Place	
3	SKQ	ID	
4	IKQ1	Duplo no.	
5	IKQ2	Type	ToolCatalog.xml: TypeNo
6	IKQ3	T. No.	No initialization
7	IQ1	Reserve	ToolBDSCatalog.xml: BaseDS
8	IQ2	Reserve	
9	IQ3	Reserve	
10	BQ1	P-status	No initialization
11	BQ2	T-status	
12	BQ3	Technology	ToolCatalog.xml: CoTe
13	aaa	1. freely configurable data element	ToolBDSCatalog.xml: BaseDS
:	:	:	
n	zzz	nth freely configurable data element	

Tab. 7-25: Tool data block initialization using the basic data block

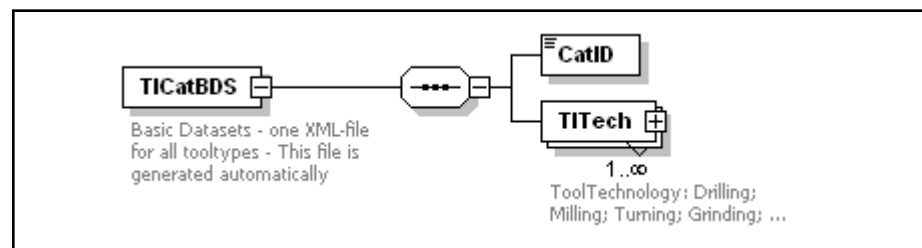


Fig. 7-95: Data scheme of tool basic data catalog (1)

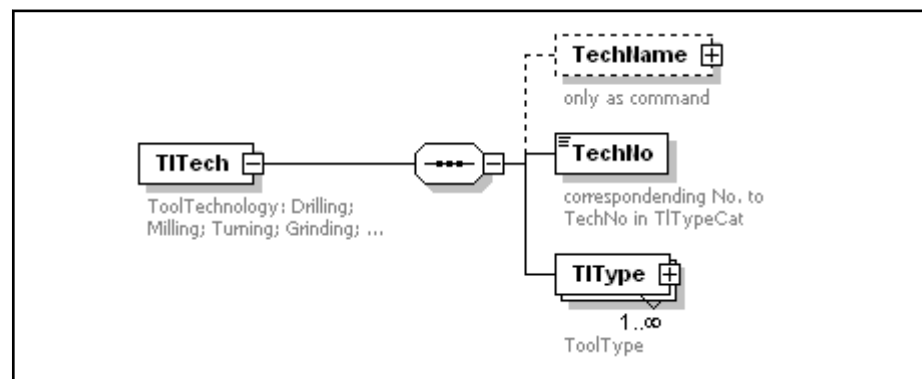


Fig. 7-96: Data scheme of tool basic data catalog (2)

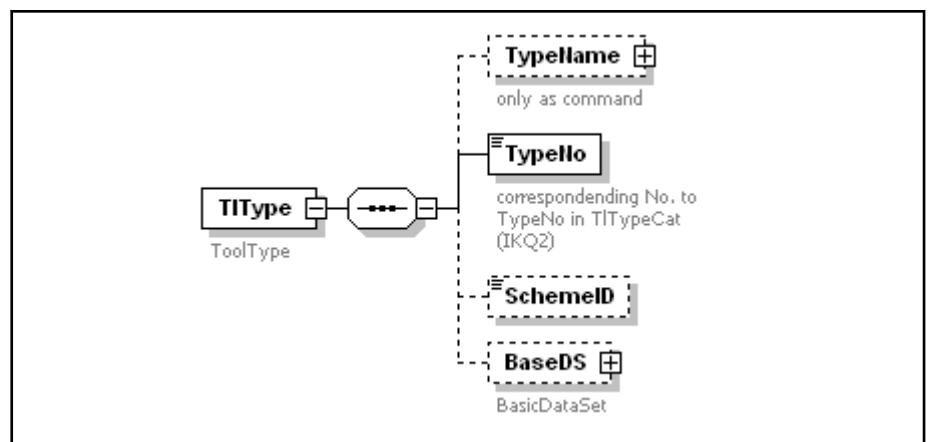


Fig. 7-97: Data scheme of tool basic data catalog (3)

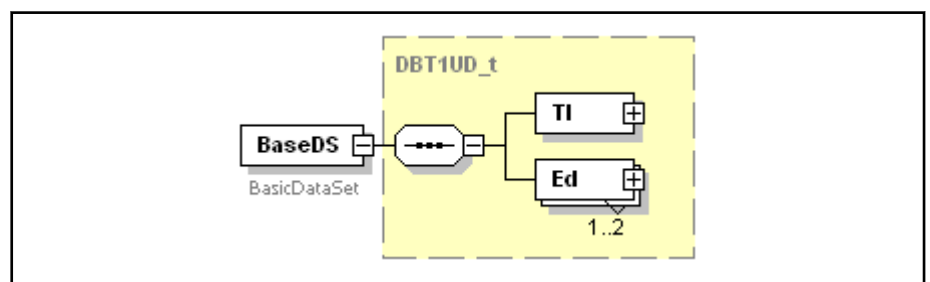


Fig. 7-98: Data scheme of tool basic data catalog (4)

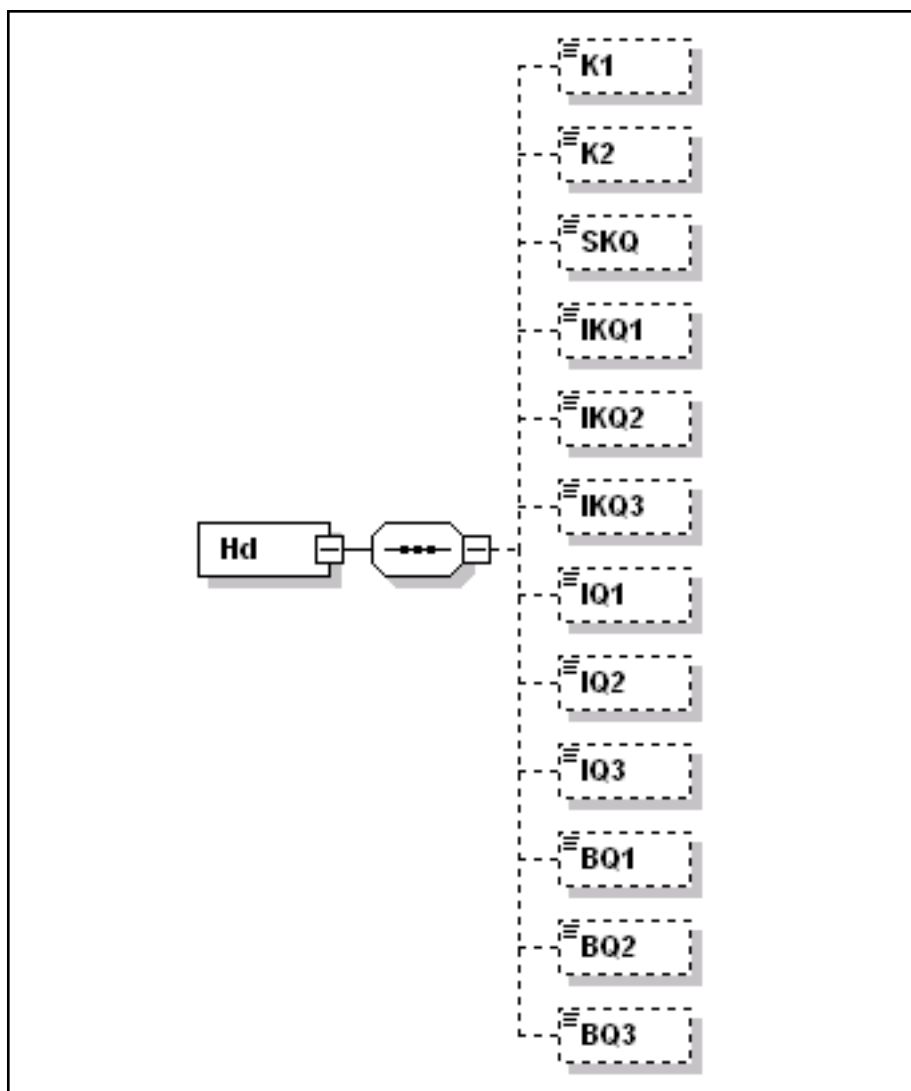


Fig. 7-99: Data scheme of tool basic data catalog (5)

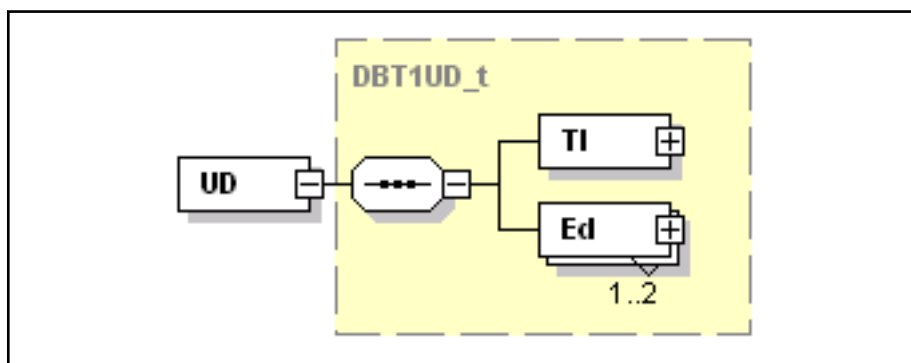


Fig. 7-100: Data scheme of tool basic data catalog (6)



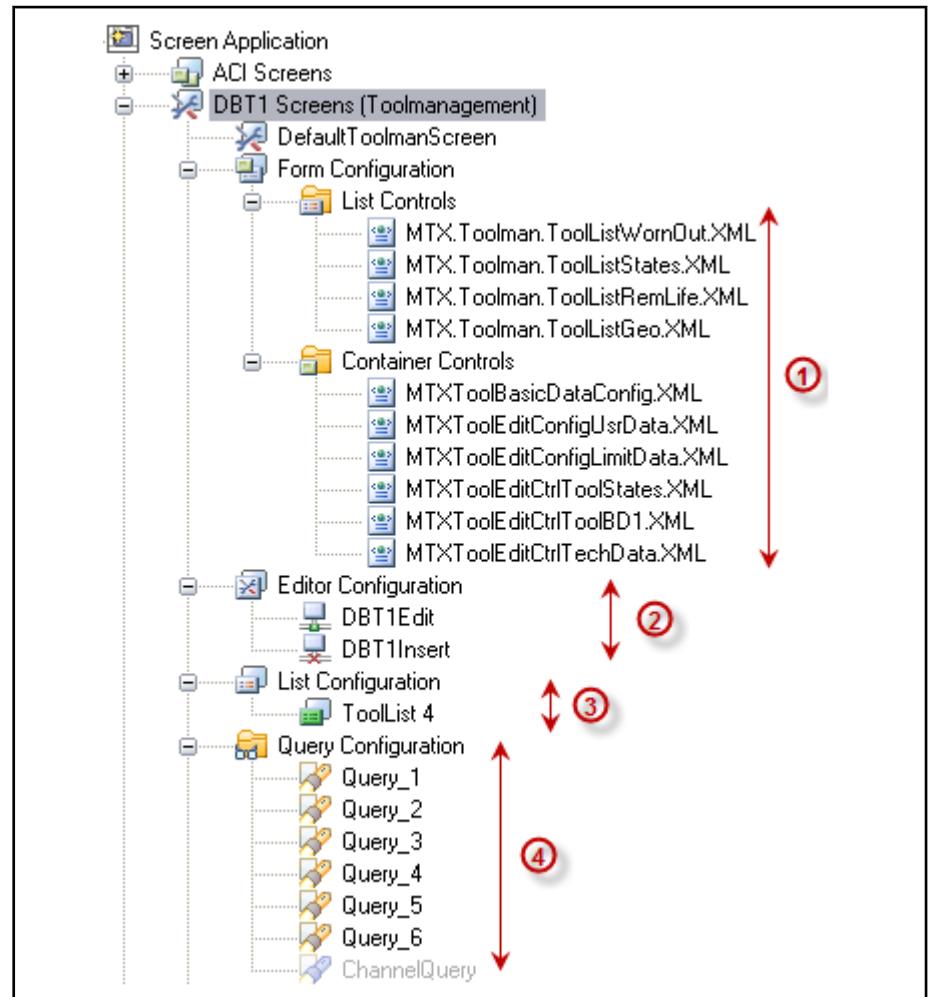
The data type "DBT1UD_t" used in the catalog scheme is the same data type which is used in the tool data scheme. This is ensured using the include instruction `<xs:include schemaLocation="dbt1sd.xsd"/>`.

7.10.6 User interface

General information

The configuration of the tool management user interface is carried out in IndraWorks Engineering.

After an HMI device has been created, a **"Tool Management Screens"** project node with the following structure is displayed in the project tree:



- 1 Predefined ULC configurations
- 2 Tool editors
- 3 Tool lists
- 4 Query definitions

Fig. 7-101: Project Explorer (tool management)

Called editors

1. Forms

- List controls
XML file editor
(refer to [chapter "XML file editor" on page 201](#))
or
ULC configurator
(refer to [chapter "ULC configurator" on page 183](#))
- Container Controls

XML file editor

(refer to [chapter "XML file editor" on page 201](#))

or

ULC configurator

(refer to [chapter "ULC configurator" on page 183](#))

2. Editor configuration

Editor configurator

(refer to [chapter "Configuration of tool editors" on page 282](#))

3. List configuration

List configurator

(refer to [chapter 7.10.6 "User interface" on page 259](#))

4. Query configuration

Query configurator

(refer to [chapter "Definition of list content" on page 261](#))

Configuring the tool lists

General information

In the "Tool management" operating range, various tool lists can optionally be defined with different contents and forms of representation.

Double-click on a list configuration to call the following setting dialog or use the pop-up menu function **Open**.

ToolList 4

Tool lists configuration file: MTX.Toolman.ToolListItemLife.XML

FKey bar: ToolmanList3

ID of database request: Query_6

MKey bar on the left side: MtxToolList_L

MKey bar on the right side: MtxToolList_R

K1<99

S	P	Tool name	DN	TN	Status	SN	activ	Tool
					tw	two	TL	
1	1	Eins	1	1	☐	☐	☐	1 ☐
								2 ☐

Fig. 7-102: List configurator

The lower part of the list configurator shows a preview of the list. In addition, the relevant configuration dialogs, which are described in detail in the follow-

ing chapters, can be called via the configuration keys .

The **"ToolManagementApp.xml"** contains all the required list information.

The query comments are language dependent and can - as it is the case for all other language-dependent texts - be defined as follows:

- A resource text to be defined in a resource DLL or as
- user text in the user text file (MLS_StringTables.csv)

Information about the list call (list identification)

Various list views can also be called via F-keys or M-keys in the MTX. For this purpose, there are 16 defined list call functions for the list ID "ToolList 1 ... 16" (without specifying the list name). Further lists can be viewed by stating the list names in the parameter field above the function "ToolList".

Definition of list content

Here, 'list content' is understood to be the selection of the indicated tool data blocks.

The query definitions in the file specify which data blocks are to be shown in a tool list.

The key "Configuration" can be used to directly call the query configurator for modifying the selected query definition.

Fig. 7-103: Query configurator

The query editor allows, in addition to the actual query definition, a query comment in the "Comment" dialog. This comment is displayed next to the list title in the headline row. Both elements are linked via a query ID in the list configuration.

Furthermore, the query configurator allows to define the database query in a user-friendly manner in the "Definition of the database query" area of the dialog.

Here, the user can decide in the first selection box whether he wants to use predefined queries or to create a freely defined query. A freely defined query is created on the screen using data element "Tool status (BQ2)".

The following default predefined queries have been supplied:

Description	Query text
All worn tools	BQ2=0x10:0x10
All sectors	K1<99
Sector 1	K1=1
Sector 2	K1=2
Sector 3	K1=3
All tools	SKQ<>"

Channel-dependent database query

The tool list can be visualized with regard to the channel.

Therefore, the default query with the ID "ChannelQuery" exists in a newly created project.

This query definition is already provided with a predefined, but changeable title "**active channel**" that is displayed additionally in the list title as for each query comment.

As shown in the following figure, the query configurator for this query type has an additional input box to select the channel.

Database query ID: ChannelQuery

Comment

Text type:

Resource Text

Assembly:

MTX.Toolman.QueryConfigControl

Resource:

Rexroth.MTX.Toolman.Controls.QueryConfigControlTextResource

Token:

txtActiveChannel

Comment:

Active channel

Definition of database request

free definable requests 1 Channel number

Sector = 1

Preview of configured database query:

K1=1

Apply

Active database query:

Function and or Overwrite

K1=1

Edit

Delete

Fig. 7-104: Channel-dependent query definition

This channel-dependent query definition is saved in a channel-dependent system variable "**SD.SysChannelQuery**" which is analyzed in the tool list display at runtime.

Database queries

Using the <Edit button>, the experienced user can enter the query condition for defining the list contents with no restraints and without any configuration help.

The following compare operators are admissible. Meaning of the data elements:

Operator	Syntax	Note
Equals	<header component> = <value>	
Greater than	<header component> > <value>	Not for SKQ
Greater than or equal to	<header component> >= <value>	Not for SKQ
Less than	<header component> < <value>	Not for SKQ
Less than or equal to	<header component> <= <value>	Not for SKQ

Operator	Syntax	Note
Not equal to	<header component> <> <value>	
Bit mask	<Header component> = <Value> : <Mask>	Not for SKQ

Tab. 7-26: Syntax for query string

Example:

K1=1, IKQ2=1001, BQ2=0x10:0x10

In example 1, all worn twist drills are shown in sector 1.

Example:

SKQ <>

In example 2, all tools in the database are shown.

For interlinked query conditions, several QueryStr can be defined for a list.

F-key panel definition

F-keypad

The valid call panel for the tool list to be defined can be selected from the F-key configurations contained under the project node "F panels".

The key "Configuration" can be used to directly call the F-key configurator for modifying the desired F-keypad.

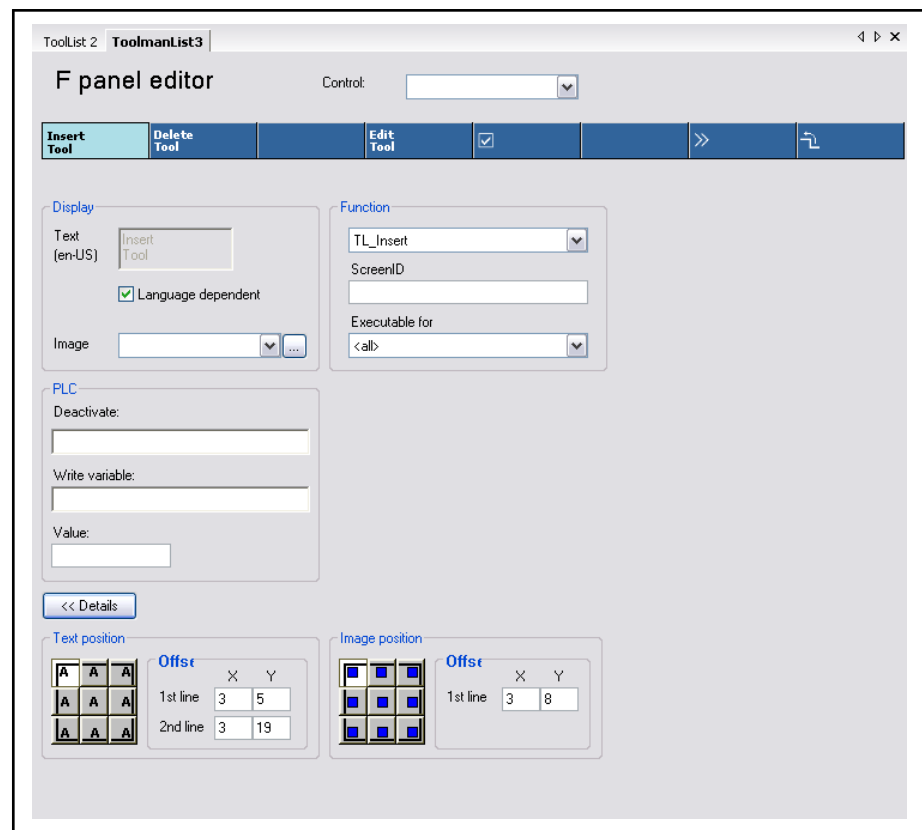


Fig. 7-105: F-key configurator

Tool management makes the following functions available for the available F-key/M-key configuration:

Name	Description	Parameter / notes
TL_Copy	Copy tool	-
TL_Delete	Delete tool	-
TL_Edit	Edit tool	-
TL_Edit_Additive	Switching additive ↔ absolute	-
TL_Export	Export list etc.	-
TL_Import	Import list	-
SingleTool_Import	Import single tool	-
TL_Insert	Insert tool	Screen ID possible
TL_Move	Move tool	Screen ID possible
TL_Editor_Command	Only applicable to M-key definitions in the tool editor	(refer to tab. 7-28 "M-key functions in the tool editor" on page 265)
TL_ChangeConfig	Modification of the list configuration	Name of the configuration file
TL_ChangeQuery	Modification of the list query	Query ID
TL_ActivateNextList	Switching to the next list if several lists are displayed	-
TL_SPS_Copy	Copy tool using PLC	PLC var
TL_SPS_Delete	Delete tool using PLC	PLC var
TL_SPS_Move	Move tool using PLC	PLC var
ToolCursor	Indicating the cursor position to the PLC	PLC var
ToolCursorEditor	Message of the tool to be edited in the editor to the PLC	PLC var
ToolList	Universal list call	ListID
ToolList 1	Call A_ToolList_1 (predef.)	-
:	:	:
ToolList 16	Call A_ToolList_16 (predef.)	-

Tab. 7-27: Function table for F-and M-key configuration

Command strings for the function TL_Editor_Command

Name	Description
Delete_BasicData_ToolType	Delete the basic data block
Save_BasicData_ToolType	Save the basic data block
Edit_Next_Tool	Selection of the next tool
Edit_Prev_Tool	Selection of the previous tool

Tab. 7-28: M-key functions in the tool editor

The following PLC data structures are required for the functions with a PLC connection:

TL_SPS_Copy Sequence:

1. Select the tool to be copied (cursor position)
2. Enter the target place using the selection dialog

3. The interface activates the PLC using the structure variable (Act = TRUE)
4. The PLC executes the copy function (writes to the database)
5. The PLC returns an acknowledgement to the interface; in a negative case, this results in an error being output in the status bar. The interface expects this answer from the PLC within 500 ms. If no answer arrives, an error message is also issued.

Structure of PLC variable:

 TYPE stCopy:

STRUCT

SrcPlace :	INT;	(*Place - source*)
SrcStorage :	INT;	(*Sector - source*)
DstPlace :	INT;	(*Place - destination*)
DstStorage :	INT;	(*Sector - destination*)
Act:	BOOL;	(*Activation*)
Value:	INT;	(*Error return*)

END_STRUCT

END_TYPE

TL_SPS_Delete**Sequence:**

1. Select the tool to be deleted (cursor position)
2. Safety prompt via the dialog "Delete: yes/no"
3. The interface activates the PLC using the structure variable (Key = TRUE)
4. The PLC executes the delete function (writes to the database)
5. The PLC returns an acknowledgement to the interface; in a negative case, this results in an error being output in the status bar. The interface expects this answer from the PLC within 500 ms. If no answer arrives, an error message is also issued.

Structure of PLC variable:

 TYPE stDelete:

STRUCT

K1:	INT;	(*Sector*)
K2:	INT;	(*Place*)
Key:	BOOL;	(*Activation*)
Value:	INT;	(*Error return*)

END_STRUCT

END_TYPE

TL_SPS_Move**Sequence:**

1. Select the tool to be moved (cursor position)
2. Enter the target place using the selection dialog
3. The interface activates the PLC using the structure variable (Act = TRUE)
4. The PLC executes the copy function (writes to the database)

5. The PLC returns an acknowledgement to the interface; in a negative case, this results in an error being output in the status bar. The interface expects this answer from the PLC within 500 ms. If no answer arrives, an error message is also issued.

Structure of PLC variable:

TYPE stMove:

STRUCT

SrcPlace :	INT;	(*Place - source*)
SrcStorage :	INT;	(*Sector - source*)
DstPlace :	INT;	(*Place - destination*)
DstStorage :	INT;	(*Sector - destination*)
Act:	BOOL;	(*Activation*)
Value:	INT;	(*Error return*)

END_STRUCT

END_TYPE

ToolCursor Sequence:

1. Transfer of the selected tool position (cursor position in the tool list) in the structure variables (K1 and K2).
2. The interface activates the PLC using the structure variable (Key = TRUE)
3. The PLC executes the specified function.

Structure of PLC variable:

TYPE stCurPos:

STRUCT

K1:	INT;	(*Sector*)
K2:	INT;	(*Place*)
Key:	BOOL;	(*Activation*)

END_STRUCT

END_TYPE

ToolCursorEditor Sequence:

1. Transfer of the tool position (of the tool to be edited) to the structure variables (K1 and K2).
2. The interface activates the PLC using the structure variable (Key = TRUE)
3. The PLC executes the specified function.

Structure of PLC variable:

TYPE stCurPos:

STRUCT

K1:	INT;	(*Sector*)
K2:	INT;	(*Place*)
Key:	BOOL;	(*Activation*)

END_STRUCT

END_TYPE

M-key panel definition

M-keypad left and right

The valid left and right M-key panels for the tool list to be defined can be selected from the M-key configurations contained under the project node "M-panels".

The key "Configuration" can be used to directly call the M-key configurator for modifying the desired M-keypad.

The screenshot shows the 'M panel editor' window for 'ToolList 2' and 'MtxToolList_R'. The device is set to 'IndraMotion MTX P60'. On the left, a sidebar contains buttons for 'Wear Data', 'Geometry Data', 'Status Data', 'Monitoring Data', and 'Tool Worn Out'. The main area is divided into several sections:

- Display:** Includes a 'Text (en-US)' field with 'Wear Data', a checked 'Language dependent' checkbox, and an 'Image' field with a dropdown and a browse button.
- Function:** Includes a 'Function' dropdown set to 'ToolList 1', an empty text field, and an 'Executable for' dropdown set to '<all>'. There is also a 'PLC flag' section with a 'Type' dropdown set to 'Free flags' and three input fields for 'Softkey', 'aktiv', and 'Disabled'.
- Text position:** Includes a 3x3 grid of buttons labeled 'A' and an 'Offset' table:

	X	Y
1st line	35	7
2nd line	35	21
- Image position:** Includes a 3x3 grid of buttons and an 'Offset' table:

	X	Y
1st line	11	10

A '<< Details' button is located at the bottom left of the main area.

Fig. 7-106: M-key configurator

For the functions, see [tab. 7-27 "Function table for F-and M-key configuration" on page 265](#).

M-keypads can be assigned to the tool lists using the following 2 methods:

Prio	Variant	Implementation
1	List-specific (local) M-keys	As described above, M-key-pads are assigned to the tool list in the list configurator.
2	Global M-keys	M-keypads are not assigned to either the list or the screen.

Tab. 7-29: Variants of M-key configuration

Definition of list display

Tool lists configurator

There are two ways of viewing the tool lists configurator. However, they use the same database (configuration file), i.e. changes made in the configurator have the same effect on the XML editor and vice versa, i.e. changes made in the XML editor have the same effect on the configurator.

1. Configurator for common configuration steps
2. XML editor for special settings which cannot be made by means of the configurator.

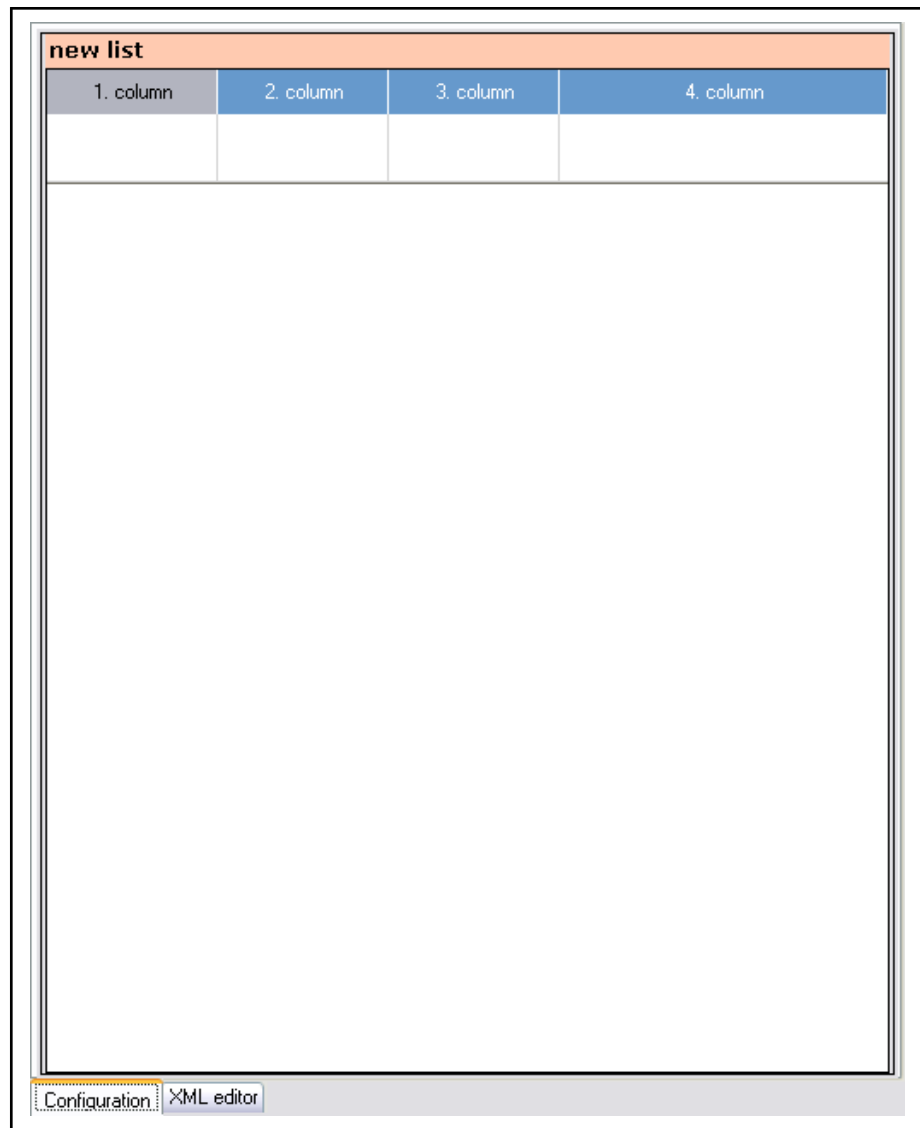


Fig. 7-107: Tool lists configurator

Most of the settings can be made using the ULC configurator (configuration) (see [chapter "ULC configurator" on page 183](#)). Only in exceptional cases do special changes need to be made directly in this configuration file using the XML editor (see [chapter "XML file editor" on page 201](#)). These cases are described in the next section.

Tool lists configuration file

The valid configuration file defining the list structure for the tool list to be defined can be selected from the controls list contained under (1).

The list structure is defined by the tool list configuration file assigned to the tool list.

All tool list configuration files are based on the schema file "**UniversalListcontrol.xsd**".

The following overview in the appendix [chapter 10.2 "Tool lists configuration file" on page 449](#) shows whether the list/editor control configuration file can be opened using the ULC configurator for all process parameters.

Description of application-specific display modifications outside of the ULC configurator

Setting the compressed display mode

If work is carried out with a list configuration that displays the data block in several partial lists (e.g. using **MTX.Toolman.ToolListConfiguration.xml**), the following handling instruction can be used to switch the display to a compressed version:

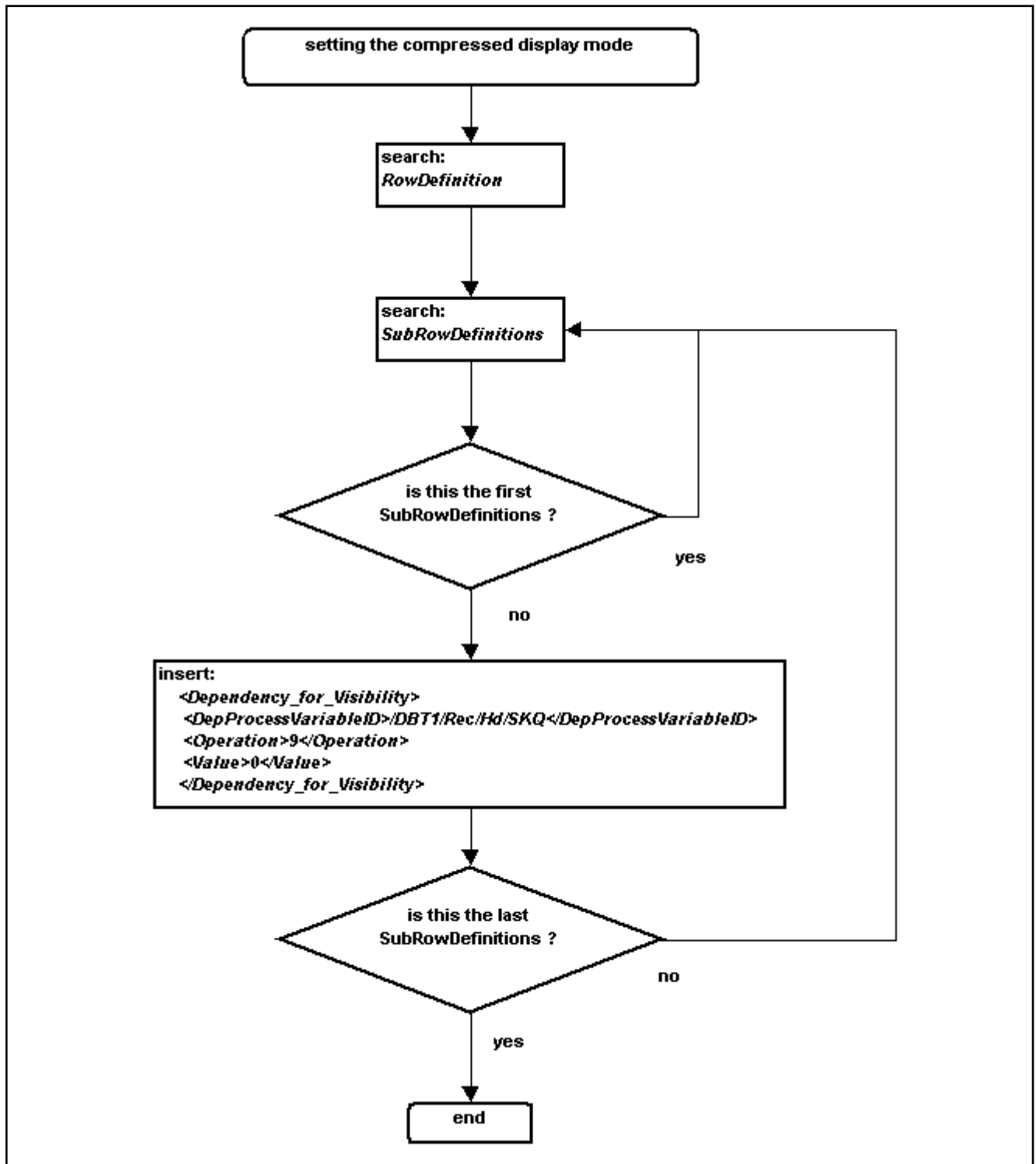


Fig. 7-108: Setting the compressed display mode

Conditional display of cell contents

During the editor/list configuration, it is possible to define the editability of a data element depending on an additional process variable.

The setting is made in the configuration file in configuration node **"Dependency_for_Visibility"** (see P1 - P5).



The prerequisites for this function is that the edit status = 3 for this cell.

Example:

The value of the tool edge-dependent data element "Geometry value L2" is displayed depending on data elements SKQ (tool name) and BQ3 (tool technology code) (determined by the CellDef node to which the "Dependency_for_Visibility" definition belongs).

Tag	Value
CellDef	
CellDef	
CellDef	
CellAddress	
ContentType	
EditType	
RepresentationDefinitions	
Dependency_for_Representation	
Dependency_for_Visibility	
DepProcessVariableID	/DBT1/Rec/Hd/SKQ
Operation	9
Value	0
FollowingDepCondition	1
Condition	
DepProcessVariableID	/DBT1/Rec/Hd/BQ3
Operation	7
Value	512
Condition	
DepProcessVariableID	/DBT1/Rec/Hd/BQ3
Operation	19
Value	15
Value2	3
RepresentationForm	1
TextFormat	false
NumFormat	
CellDef	
CellDef	

- 1
- 2
- 3

1. Display condition
2. Display condition
3. Display condition

Fig. 7-109:

Example: Conditional display of cell contents

1. Display condition: P.stringlength longer V

Tool name may not be an empty string, i.e. the place has to contain a tool

2. Display condition: P AND V != 0

Bit 10 in the technology code word has to be set, i.e. the L2 correction value has to be relevant for the type of the displayed tool data block (see tool catalog)

3. Display condition: (P AND V) higher or equal {MN}

The number of the subordinate column is less than or equal to the number of tool edges of the tool data block, i.e. the tool edge datum is relevant for the number of tool edges of the data block

Explanation for MN: MN = 0

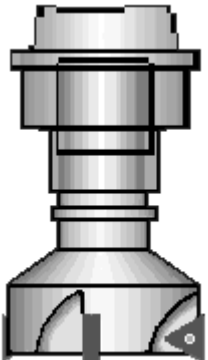
Partial list

MN = 1	Subordinate line
MN = 2	Column
MN = 3	Subordinate column

Setting process-dependent bitmap display

It is possible to display bitmaps from resource files depending on the value of a process variable in the tool list.

Since the bitmap files must be a component of a resource DLL, a default resource DLL (assembly file) is included; in turn, this contains the following bitmap resource files (.resx files):

AssemblyName	BitmapResourceName	Contents	Example (original size)
MTX.Toolman.ToolmanUserBitmapResource	ToolTypes_MediumSizeBmp.resx	Tool type	
	ToolTypes_SmallSizeBmp.resx	Tool type	
	ToolTypes_IconSizeBmp.resx	Tool type	
	ToolStorageBmp.resx	Storage type	
	ToolTecTypesBmp.resx	technology type	

Tab. 7-30: Provided bitmap libraries

Example:

Storage type bitmap

A bitmap is assigned to the value of K1 (sector number) using an assignment table.

Value of K1	Bitmap	Description
1		Spindle
2		Grippers
3		Magazine

Control commissioning

Value of K1	Bitmap	Description
4		Turret
99		Tool cabinet

Tab. 7-31: Example storage type bitmaps

Settings in the configuration file:

1. Setting: Display of the process variables as a bitmap.

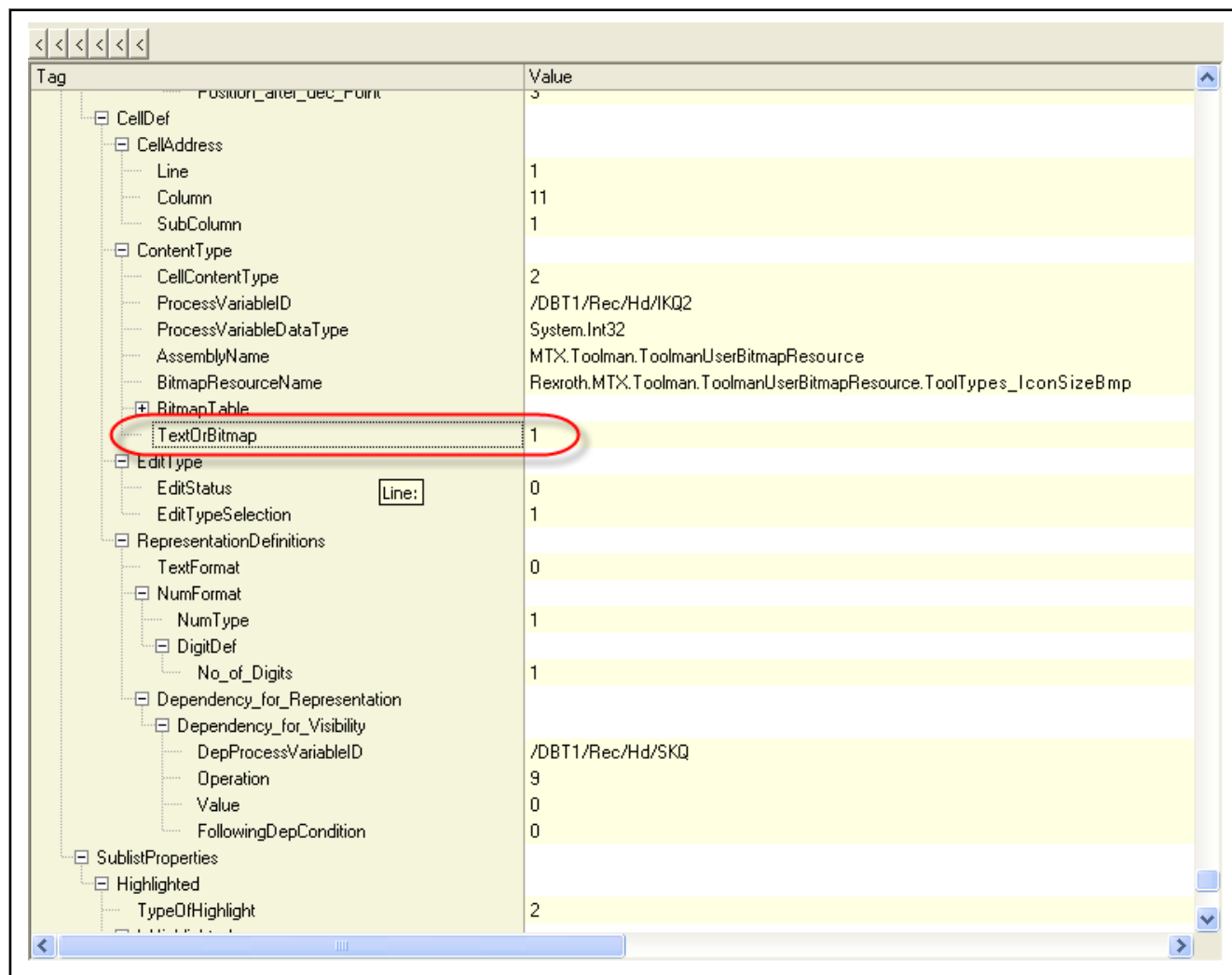


Fig. 7-110: Setting the display of the process variables as a bitmap

2. Define AssemblyName, BitmapResourceName and the bitmap table (see Q1 - Q2).

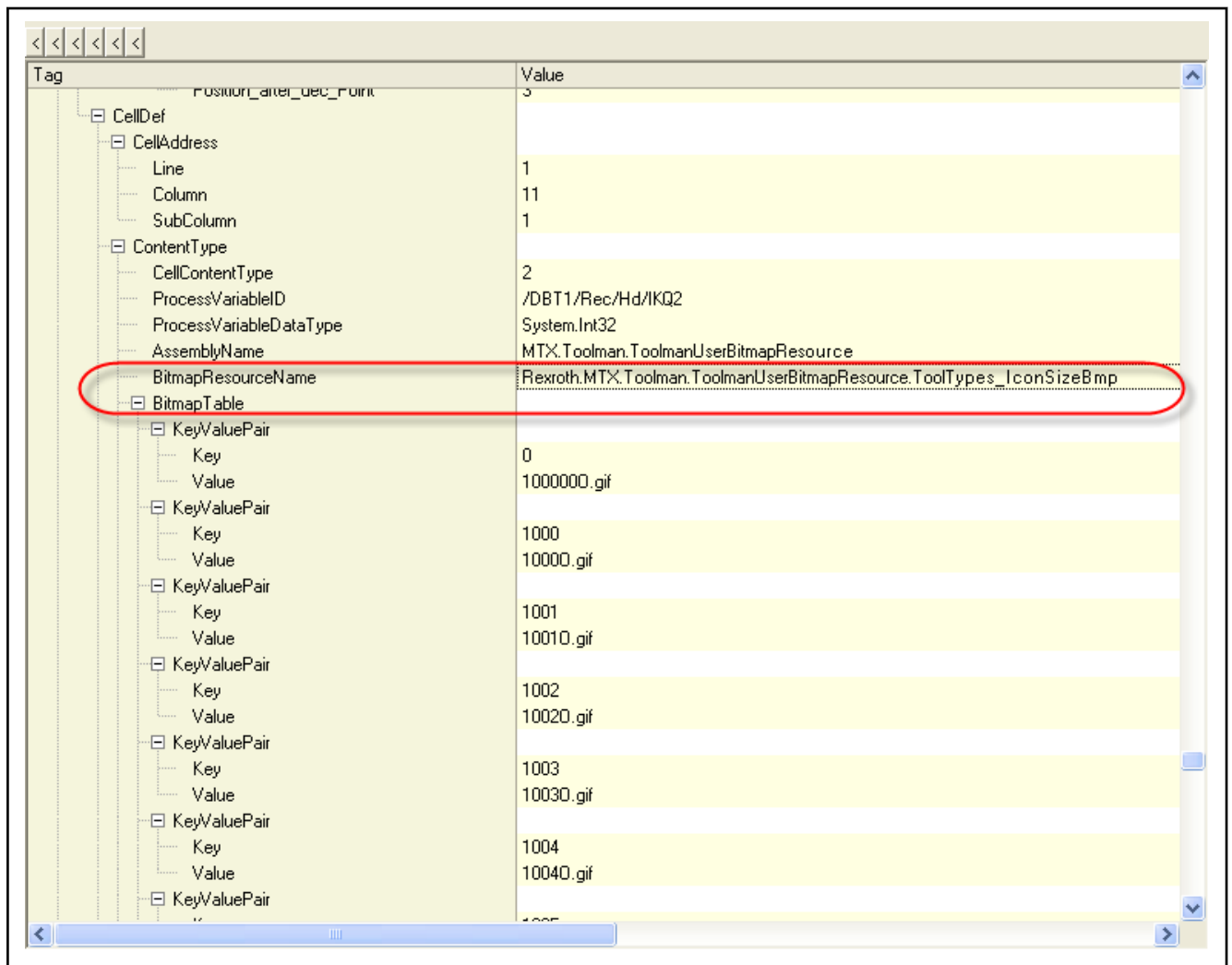


Fig. 7-111: Bitmap links

Setting process-dependent text display

It is also possible to display also texts from resource files depending on the value of a process variable in the tool list.

The texts can either be firm texts or can originate from a text resource DLL or from a user text file.

Settings in the configuration file:

1. Setting: Display of the process variables as a text.
2. Via **"TextTableUse"** it can be set if the text is to be a determined text (value = 0), a resource text (value = 1) or a user text (value = 2).
3. Definition of AssemblyName, TextResourceName (in the case of resource texts being used) and the text table (see Q 1 - Q2).

Control commissioning

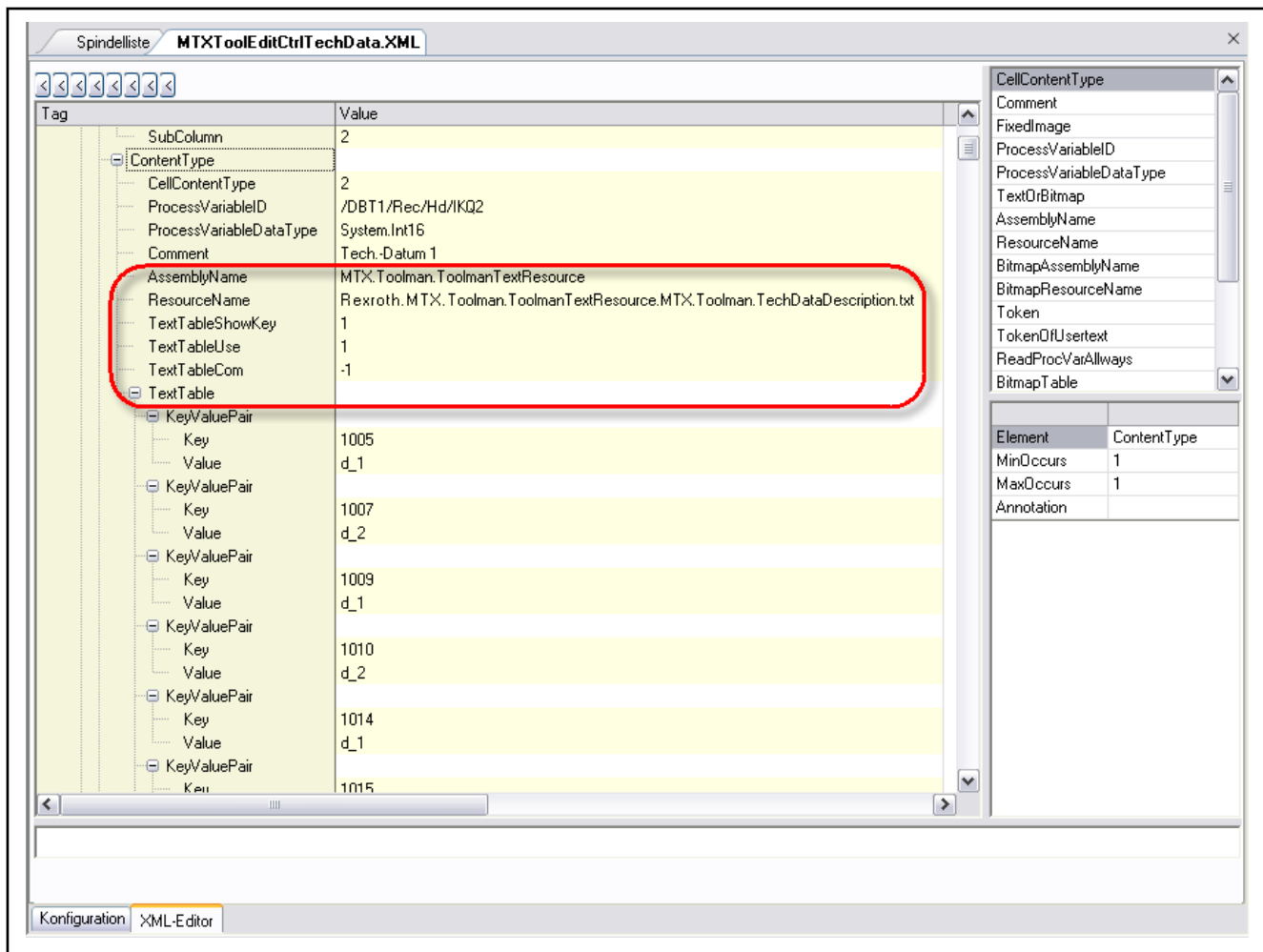


Fig. 7-112: Text references

4. Via "TextTableShowKey" it can be set how the display reacts in the case of the defined display conditions of the text table not being fulfilled. It can also be set if () is a resource text (value=1) or a user text (value=2).

- Value = 0: The cell is displayed as empty
- Value=1: The cell is displayed as if the process value = 0

Conditional editability of cell contents

During the editor/list configuration, it is possible to define the editability of a data element depending on an additional process variable.

The setting is made in the configuration file in the configuration node "EditDepend" (see P1 - P5).

Example:

Conditional editability

The editing of a selected tool datum is permitted depending on data element BQ2 - bit 7 (Tool locked).

BQ2 AND 64 != 0

(Operation 7: P AND V != 0)

FollowingDepCondition	1
Condition	
DepProcessVariableID	/DBT1/Rec/UD/Ed[2]/Limits/MinLimits/Rad
Operation	5
EditDepend	
DepProcessVariableID	/DBT1/Rec/Hd/BQ2
Operation	7
Value	64
Line:	
TextFormat	false
NumFormat	
NumType	2
Pos_dec_Point_Def	

Fig. 7-113: Example of conditional editability



The prerequisite for this function is that the **edit status = 3** in the edit type definition.

Highlighted definition of data blocks

It is possible to highlight tool data blocks in the tool list depending on a process variable. As many displays as desired can be implemented.

Example:

Highlighted definitions

BQ2 - bit 0 (tool active) → background color 1

BQ2 - bit 4 (tool worn) → background color 2

Control commissioning

Tag	Value
NumType	1
DigitDef	
Line:	
No_of_Digits	1
Dependency_for_Representation	
Dependency_for_Visibility	
DepProcessVariableID	/DBT1/Rec/Hd/SKQ
Operation	9
Value	0
FollowingDepCondition	0
SublistProperties	
Highlighted	
TypeOfHighlight	2
IsHighlighted	
DepProcessVariableID	/DBT1/Rec/Hd/BQ2
Operation	7
Value	1
IsHighlighted	
DepProcessVariableID	/DBT1/Rec/Hd/BQ2
Operation	7
Value	64
HighlightColor	
Red	33
Green	100
Blue	200
IsHighlighted	
DepProcessVariableID	/DBT1/Rec/Hd/BQ2
Operation	7
Value	128
HighlightColor	
Red	50
Green	150
Blue	250

- 1 Highlighted condition
- 2 Highlighted condition
- 3 Color setting for the background color of the 2nd definition
- 4 Highlighted condition
- 5 Color setting for the background color of the 3rd definition

Fig. 7-114: Example of several highlighted definitions



Color settings, including the background color for the 1st definition, are always located in the configuration node **Styles/Highlighted**; the foreground color that is defined there applies to all highlighted definitions.

Definition of value range checks during the entry of values

In addition to limit value monitoring in the PLC, it is possible to configure the tool list / tool editor in such a manner that only the entry editor carries out a value range check during entry.

If the value range is exceeded, an error message is displayed in the status bar.

As is the case for the check using the PLC, it is required that limit values be defined in the data block (see [chapter 7.10.4 "Configuring the database" on page 213](#)).

The setting for the check condition is made in the configuration file in configuration node **"Validation"** (see P1 - P5)

Also write:

instead of /DBT1/Rec/...

/DBT1/Rec[{0}]/...

Example:

Value Range Check during Entry in the Editor

(value of the limit $R_{\max}$ of tool edge 2) $\geq$ input value $\geq$ (value of the limit $R_{\min}$ of tool edge 2)

Value	15
Value2	3
Validation	
DepProcessVariableID	/DBT1/Rec/UD/Ed[2]/Limits/MaxLimits/Rad
Operation	6
FollowingDepCondition	1
Condition	
DepProcessVariableID	/DBT1/Rec/UD/Ed[2]/Limits/MinLimits/Rad
Operation	5
EditDepend	
DepProcessVariableID	/DBT1/Rec/Hd/BQ2
Operation	7

Fig. 7-115: Example of value range inspection during entry in the editor

Possible operations (generally applies to the DependencyType):

Explanation of abbreviations:

P Process variable (ProcVariableID)

V Value (Value)

V2 2nd value (Value 2)

Explanation for MN: V2
Value is preset by

MN = 0 Partial list

MN = 1 Subordinate line

MN = 2 Column

MN = 3 Subordinate column

Permitted operations:

The value is written if:

1	P==TRUE	Process variable log TRUE is
2	P<V	The process variable is lower than the specified value
3	P>V	The process variable is greater than the specified value
4	P==V	The process value is equal to the specified value
5	P≤V	The process value is lower than or equal to the specified value
6	P≥V	The process value is greater than or equal to the specified value
7	P∧V≠FALSE	The process variable is AND-logically linked with the specified value and the result is logic TRUE

8	$\sim P \wedge V \neq \text{FALSE}$	The negative process variables are AND-logically linked with the specified value and the result is logic TRUE
9	$\text{strlen}(P) < V$	The string length of the process variables is higher than the specified value (P of the string type, e.g. SKQ) is
10	$(P \wedge V) < V2$	Process variable is AND-logically linked with the specified first value and the result is lower than the second specified value
11	$(P \wedge V) > V2$	Process variable is AND-logically linked with the specified first value and the result is higher than the second specified value
12	$(P \wedge V) == V2$	Process variable is AND-logically linked with the specified first value and the result is equal to the second specified value
13	$(P \wedge V) \leq V2$	Process variable is AND-logically linked with the specified first value and the result is lower than or equal to the second specified value
14	$(P \wedge V) \geq V2$	Process variable is AND-logically linked with the specified first value and the result is higher than or equal to the second specified value
15	$(P \wedge V) < \text{MN}[V2]$	Process variable is AND-logically linked with the specified first value and the result is less than the value of the MN process variable specified by V2
16	$(P \wedge V) > \text{MN}[V2]$	Process variable is AND-logically linked with the specified first value and the result is higher than the value of the MN process variable specified by V2
17	$(P \wedge V) == \text{MN}[V2]$	Process variable is AND-logically linked with the specified first value and the result is equal to the value of the MN process variable specified by V2
18	$(P \wedge V) \leq \text{MN}[V2]$	Process variable is AND-logically linked with the specified first value and the result is lower than equal to the value of the MN process variable specified by V2
19	$(P \wedge V) \geq \text{MN}[V2]$	Process variable is AND-logically linked with the specified first value and the result is higher than equal to the value of the MN process variable specified by V2

Handling instruction: Configuring the list display

This handling instruction refers to the procedure described for configuring the display form of a tool list.

Configuring the list display

The following subtasks can be accomplished:

1. Adding/deleting a column
2. Adding/deleting a subcolumn
3. Changing a column or subcolumn title
4. Changing the column or subcolumn width
5. Setting the editing properties



See [chapter "ULC configurator" on page 183](#).

Documentation	MTX Functional Description	Tool list configuration

IW Engineering: Data transfer

1. Saving the configuration
2. Saving and, if necessary, activating the project

Documentation	IndraWorks HMI	Data transfer

Handling instruction: Configuring the list content

The following handling instruction describes the procedure to edit the displayed content of a list.

IW Engineering/Tool list configuration: Enter or Double-Click a List Configuration

1. Selecting of list configuration files
2. Selecting the F-key panel and configure it, if necessary
3. Selecting the right and left M-key panels and configure them, if necessary
4. Defining the list query

Documentation	MTX Functional Description	Tool list configuration

IW Engineering/Tool list configuration: Configuring the list definition

Documentation	MTX Functional Description	Data transfer

IW Engineering: Data transfer

1. Saving the configuration
2. Saving and, if necessary, activating the project

Documentation	IndraWorks HMI	Data transfer

Handling instruction: Creating a new list configuration

The following handling instruction describes the procedure to create a new list configuration.

IW Engineering/Tool list configuration: Pop-up or menu function "New"

A dialog to enter/select the list name is displayed. This name can be entered during the F/M-key configuration while entering the parameter **ListID** in the function **ToolList**.

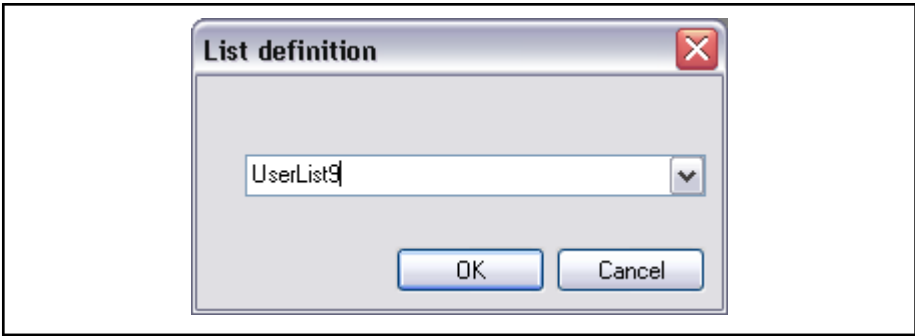


Fig. 7-116: Creating a new tool list definition

Documentation	MTX Functional Description	Tool list configuration

IW Engineering/Tool list configuration: Configuring the list display

		Instruction chapter "Handling instruction: Configuring the list display" on page 280
Instruction:	IndraWorks Commissioning	Configuring the list display

IW Engineering: Data transfer

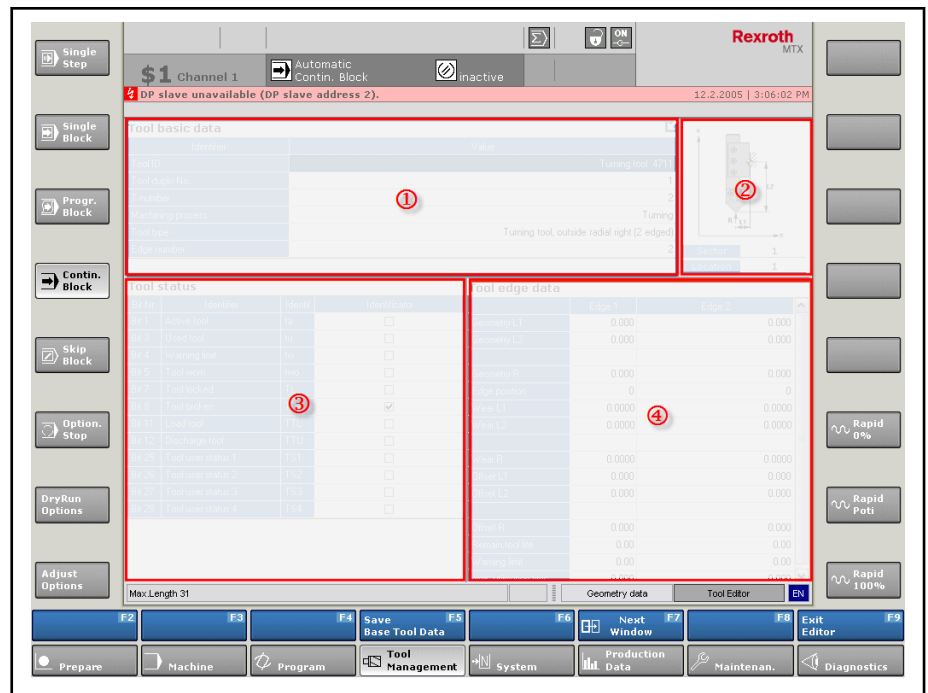
Executing an HMI download

Documentation	IndraWorks HMI	Data transfer

Configuration of tool editors

General information

Brief description	Like the NC main screens, the tool editors function according to the ACI principle (Active Container Interface). <i>This is to say:</i> • They can be freely configured • They contain several data containers which in turn can comprise several containers • The controls can also be configured • Navigation of the cursor, and thus editing, always takes place within the focused container only
Description	The various groups of tool data are shown using separate displays (e.g. tool edge data), so-called controls. When it is delivered, the Tool Editor has the following screen division.



- 1 Basic data container
- 2 Bitmap container
- 3 General tool data container
- 4 Tool edge data container

Fig. 7-117: Container distribution of the tool editor

The number and assignment of containers can be changed by the user. This information is stored in the frame configuration file "MTXToolEditor_Data.xml". The control configurations of the containers are assigned as follows in a standard project.

Container No. in the figure	Listcontrol	Name of the configuration file
1	Basic data control	MTXToolBasicDataConfig.XML
2	Bitmap control	Cannot be configured and focused
3	Tool status control	MTXToolEditCtrlToolStates.XML
3	Place status control	MTXToolEditCtrlPlaceStates.XML
3	Technology data control	MTXToolEditCtrlTechData.XML
3	Tool user data control	MTXToolEditCtrlToolBD1.XML
4	Tool edge data control	MTXToolEditConfigUsrData.XML
4	Geometry limit values control	MTXToolEditConfigLimitData.XML

Tab. 7-32: Overview on all editor configuration files

If there are several controls in one container (as for the default configuration), these can be selected via <F3><Next display>.

The bitmap container is an exception. It cannot be focused and the displayed tool screen is switched via the type of the respectively activated display control.

Display of a coordinate system in the bitmap container

It is possible to store the tool screen for a better illustration in a coordinate system.

This display is controlled via the system variable **SD.SysCoordSystem**.

If no automatic determination of the coordinate system is set (**SD.SysCoordSystem.Calc = 0**), the stored coordinate system can be set via the variable **SD.SysCoordSystem.Value**.

The coordinate system display is configured under the node **Tool Catalog ▶ Coordinate system**, see [chapter "Visualizing and configuring the tool catalog" on page 248](#).



Prerequisite for this function: the control device was switched on-line in IndraWorks Engineering.

As a rule, there are two groups of tool editors that can be managed separately in the project tree of Engineering Desktop under project node **"Editor Configuration"**:

Name		Method of operation	Icon	Example
Insert type	Offline editor:	Editor for inserting a tool into the tool list		DBT1Insert
Edit type	Online editor:	Editor for modifying a tool data block		DBT1Edit

Newly created tool editors are always assigned to one of the two editor groups during creation by selecting the menu function **New Screen: Edit type** or **New screen: Insert type**; assigned to one of the two editor groups upon creation and displayed in the tree with the appropriate symbol. However, this assignment does not yet apply to new editors from old projects. However, it can be subsequently performed later on via the **Assign to Edit Screens** or **Insert Screens**.

The distribution, selection and arrangement of the tool data displays of the existing tool editors can be changed using the "Configuration" dialog. New tool editors can be created and ones that are no longer needed can be deleted.

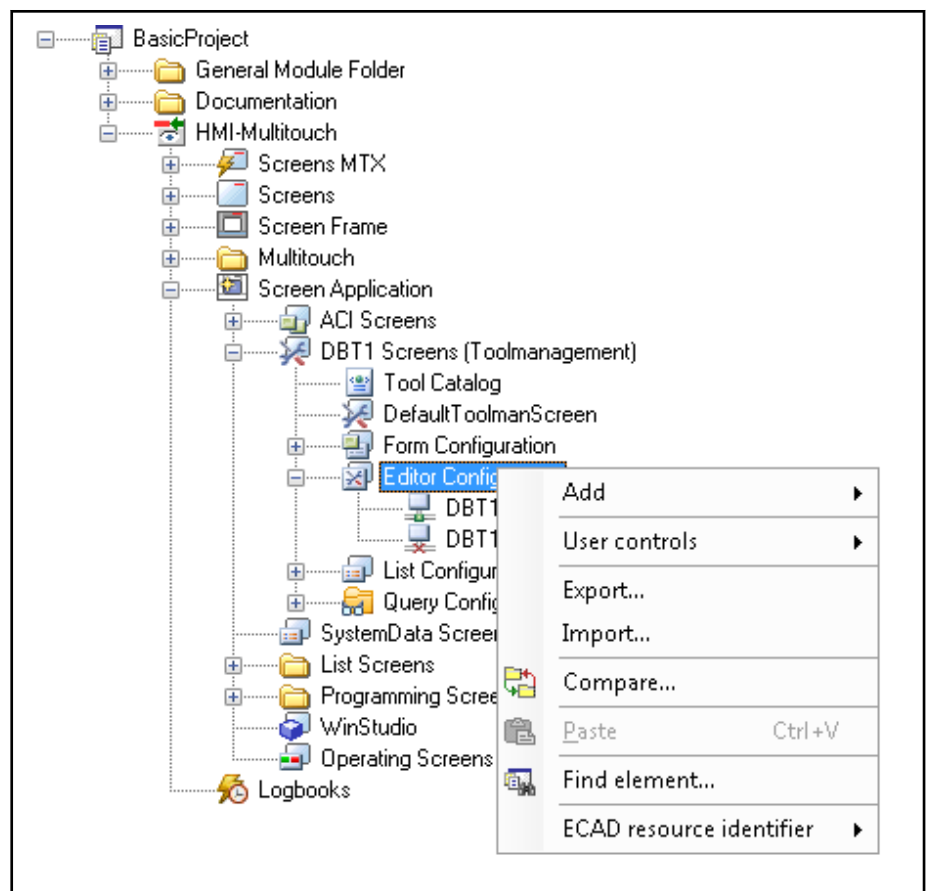


Fig. 7-118: Context menu: Editor configuration

The dialogs and functions to configure properties shared by all the tool editors as well as those to create a new tool editor, can be opened via the context menu of project node **"Editor Configuration"**.

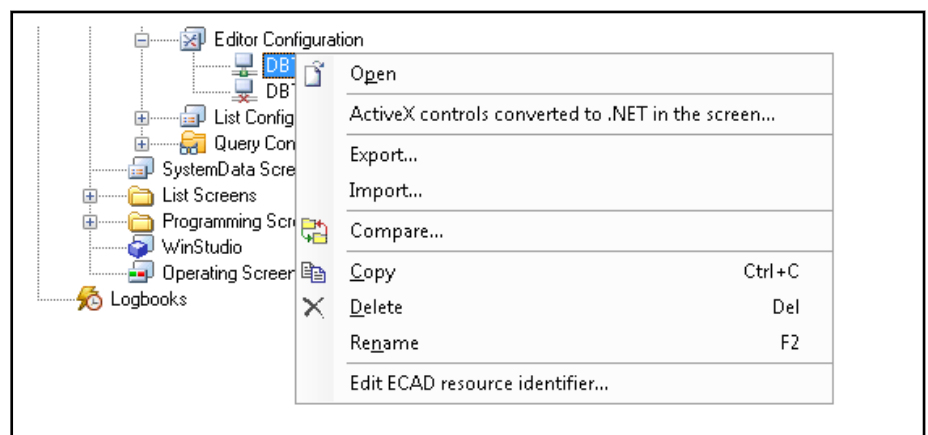


Fig. 7-119: Context menu: Configuration (tool editor)

Dialog "New Edit/Insert Screen"

A new editor can be created using the dialog **New screen**. The new editor is called in the context menu of project node **"Editor configuration"** with the entry **New screen: Edit type** or **New screen: Insert type**.

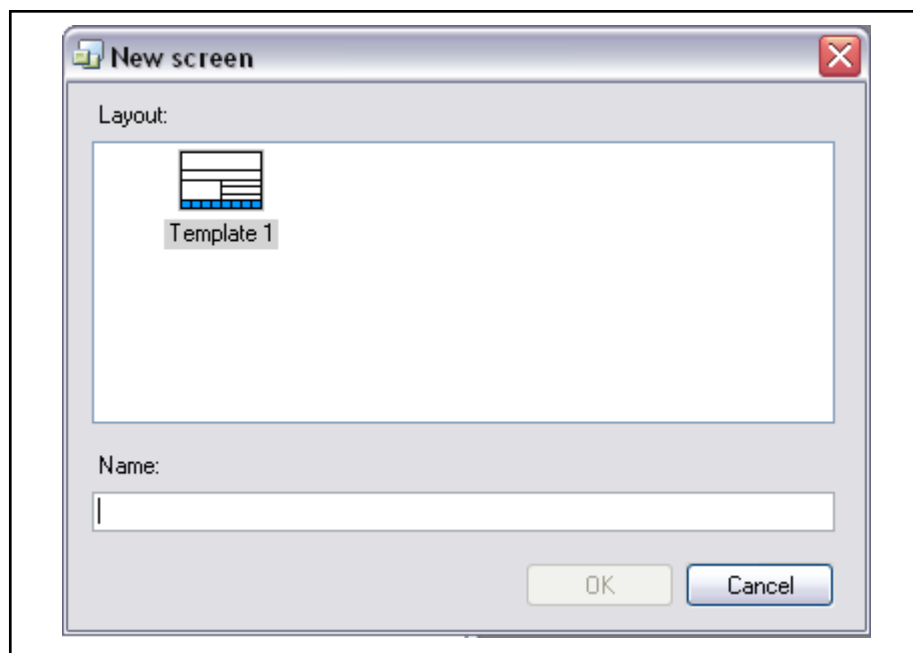


Fig. 7-120: Dialog: New screen (with layout templates)

To create a new screen, it is necessary to select a layout template and to enter a unique screen ID.

"Dataset screen editor"

The **"Dataset Screen Editor"** is called via "Open" in the context menu of the data set screen to be edited. The division of the editor into screen segments, along with their number and position, can be changed in this dialog.

For a general description on the configuration of ACI screens in the HMI, refer to the HMI manual **"DOK-IWORKS-HMI*Vxx****-APxx-EN-P"**.



In case of a tool editor configuration, the following panels can be used, depending on whether an online or an offline editor is to be configured:

For an online editor:

- F-key file: **MTX.Toolman.Editor.Edit.xml**
- *possible start F-keypad:*
 - **MTXToolEdit_BasicData_ControlPanel**
 - MTXToolEdit_BasicData_ControlPanel2 (to be used only internally)
 - **MTXToolEdit_ToolStates_ControlPanel**
 - **MTXToolEdit_UserData_ControlPanel**
 - MTXToolEdit_UserData_ControlPanel2 (to be used only internally)
 - **MTXToolEdit_BitMap_ControlPanel**

For an offline editor:

- F-key file: **MTX.Toolman.Editor.Insert.xml**
- *possible start F-keypad:*
 - **MTXToolInsert_BasicData_ControlPanel**
 - **MTXToolInsert_ToolStates_ControlPanel**
 - **MTXToolInsert_UserData_ControlPanel**
 - **MTXToolInsert_BitMap_ControlPanel**

Online editor:

F-keypad level	<F2>	<F3>	<F4>	<F5>	<F6>	<F7>	<F8>	<F9>
1. Level				Save basic data				

Tab. 7-33: MTXToolEdit_BasicData_ControlPanel

F-keypad level	<F2>	<F3>	<F4>	<F5>	<F6>	<F7>	<F8>	<F9>
1. Level		Next display						

Tab. 7-34: MTXToolEdit_ToolStates_ControlPanel

F-keypad level	<F2>	<F3>	<F4>	<F5>	<F6>	<F7>	<F8>	<F9>
1. Level		Next display						

Tab. 7-35: MTXToolEdit_UserData_ControlPanel

F-keypad level	<F2>	<F3>	<F4>	<F5>	<F6>	<F7>	<F8>	<F9>
1. Level							Switch level	

Tab. 7-36: MTXToolEdit_BitMap_ControlPanel

Offline editor:

Control commissioning

F-keypad level	<F2>	<F3>	<F4>	<F5>	<F6>	<F7>	<F8>	<F9>
1. Level				Save basic data				

Tab. 7-37: MTXToolInsert_BasicData_ControlPanel

F-keypad level	<F2>	<F3>	<F4>	<F5>	<F6>	<F7>	<F8>	<F9>
1. Level		Next display						

Tab. 7-38: MTXToolInsert_ToolStates_ControlPanel

F-keypad level	<F2>	<F3>	<F4>	<F5>	<F6>	<F7>	<F8>	<F9>
1. Level		Next display						

Tab. 7-39: MTXToolInsert_UserData_ControlPanel

F-keypad level	<F2>	<F3>	<F4>	<F5>	<F6>	<F7>	<F8>	<F9>
1. Level								

Tab. 7-40: MTXToolInsert_BitMap_ControlPanel

- If a control contains the implementation of the MTXACInterface.dll interface, the configuration parameter that can be entered in field "Instance" can be transferred to it. If the control can display different process data or if it has various types of displays, this configuration parameter can be used during the initial instancing of the control to determine which of these instances the control should use.

Configuring the tool editor controls

The following tool editor controls can be configured as the tool lists via the ULC configurator:

- MTXToolBasicDataConfig.XML
- MTXToolEditCtrlToolStates.XML
- MTXToolEditCtrlPlaceStates.XML
- MTXToolEditCtrlToolBD1.XML
- MTXToolEditConfigUsrData.XML
- MTXToolEditConfigLimitData.XML

As opposed to the list configuration, however, only the data of one tool are displayed.



There is no way to directly call the ULC configurator from the editor configuration. The method using **Form Configuration ► Container Controls** must be chosen.

Configuring the M-keypads in the tool editor

The tool editors can receive their own M-keypads specific to their type with the following fixedly determined panel names:

Editor type	M-keypad	M-panel name
Insert type	Right	ToolInsertEditorLocalRight
	Left	ToolInsertEditorLocalLeft
Edit type	Right	ToolEditorLocalRight
	Left	ToolEditorLocalLeft

Tab. 7-41: Names of the M-key panels of the tool editors

Said M-keypads can be configured by the user in the M-key configurator.

For this, all keys should be defined by means of the "TL_Editor" function using the following subfunctions in "CommandString":

Function name	CommandString	Description
TL_Editor	Edit_Prev_Tool	Search for the previous tool
	Edit_Next_Tool	Search for the next tool
	Save_BasicData_Tool-Type	Save the current data block as a basic data block
	Delete_BasicData_Tool-Type	Delete the basic data block

Tab. 7-42: Commands for the M-key functions in the tool editor



The use of other functions for the M-key configuration is possible; however, it is not recommended, since these usually do not result in a reasonable behavior of the tool editor (e.g. Delete tool).

Handling instruction: Configuring the tool editor controls

The configuration of the following tool editor controls takes place in the same way as tool list configuration, the only difference being that the data of only one tool are indicated in these lists:

- MTXToolBasicDataConfig.XML
- MTXToolEditCtrlToolStates.XML
- MTXToolEditCtrlPlaceStates.XML
- MTXToolEditCtrlToolBD1.XML
- MTXToolEditConfigUsrData.XML

This way, it is ensured that the entry **<ShowMultipleSublists>** = FALSE is set and that the entry **<PathOfMultiplicator>** remains empty.

IW Engineering: Data transfer

1. Save the project
2. Insert the icon

Defining the display of the coordinate system in the tool bitmap control

A coordinate system establishing an axis reference of the correction values is stored in the displayed tool bitmap.

This axis reference can be set via the system date **SD.SysCoordSystem.Value**.

Bitmap control

Coordinate system	Ordinate	Abscissa	SD.SysCoordSystem.Value
Without	-	-	0
	X	-	1
	Y	-	3
	Z	-	5
Fig. 7-121:			
	X	-	2
	Y	-	4
	Z	-	6
Fig. 7-122:			
	X	Z	10
	X	Y	20
	Y	X	30
	Y	Z	40
	Z	X	50
Fig. 7-123:			
	Z	Y	60
	X	Z	11
	X	Y	21
	Y	X	31
	Y	Z	41
Fig. 7-124:			
	Z	X	51
	Z	Y	61
	X	Z	12
	X	Y	22
	Y	X	32
Fig. 7-125:			
	Y	Z	42
	Z	X	52
	Z	Y	62
	X	Z	13
	X	Y	23
Fig. 7-126:			
	Y	X	33
	Y	Z	43
	Z	X	53
	Z	Y	63

Tab. 7-43: Coordinate system

The displayed coordinate system can also be determined automatically. But presettings have to be made once.

Specifications whether the value SD.SysCoordSystem.Value should be determined automatically:

SD.SysCoordSystem.Calc: 1 - Automatic value determination
0 - No automatic value determination

Specifications for the machine configuration:

Machine coordinate system:

SD.SysCoordSystem.horizontal: 1 - Abscissa in positive direction
0 - Abscissa in negative direction

SD.SysCoordSystem.vertical: 1 - Ordinate in positive direction
0 - Ordinate in negative direction

Master axis meaning:

SD.SysCoordSystem.X_axis_pref: 1 - X-axis exists and is preferably displayed
 0 - X-axis does not exist
 SD.SysCoordSystem.Y_axis_pref: 1 - Y-axis exists and is preferably displayed
 0 - Y-axis does not exist
 SD.SysCoordSystem.Z_axis_pref: 1 - Z-axis exists and is preferably displayed
 0 - Z-axis does not exist

Tab. 7-44: SD.SysCoordSystem

That means if the value of the variables SD.SysCoordSystem.Value should be determined automatically, 1 has to be assigned to SD.SysCoordSystem.Calc. it is only then required to set the machine configuration.

Configuring the user management

General information

MTX user management is classified into

- data-related user management and
- functional user management.

Data-relevant user management allows for assignment of data element-relevant read and write privileges to the individual user groups. As described above, this is realized via attribute definitions (L1, L2, L3, L3, L5) in the data block schema.



Presently, the functional user management cannot be configured by the user.

Handling instruction: Defining data-relevant user privileges

The following handling instructions describe the process to be followed when the user privileges for the individual data elements are to be modified.

IW Operation/Program: Edit the Schema File "dbt?ud.xsd"

1. Copy the file "dbt?ud.xsd", "dbt?sd.xsd" (?:= 1 [DBT1] or 2 [DBT2]) or "tool_ty.xsd" in control directory "\usrfep\schemas" or "\feprom\schemas" or "\root\schemes" to the mount directory (\mnt).
2. Edit user rights L1 to L5 of the respective nodes or elements in the schema file by means of the XSD editor (preferably with Altova SPY).
3. Save file.
4. Copy the file back to the control directory "\usrfep" or "\root".

Figure:		Image of data block schema
Documentation	MTX Functional Description	Edit schema file

NC: Apply new data structure

1. Complete IW Operation
2. Reset control

Documentation	MTX Functional Description	Apply new data structure

7.10.7 Interfaces

CPL interfaces

The MTX provides the following NC commands for accessing the tool database:



For the detailed syntax of the NC commands described below, please see the documentation "MTX Programming Manual". Here, only the names of the NC commands are mentioned, each illustrated by one example.

TCV Provides the latest programmed tool correction values, either as a total (D-correction + external correction memory) or as a single value.

Example: Reading of the L2 tool length of the last programmed ED-correction.

Program:

```
110 TCV(2,"E")
```

DCT Read and write access to tool correction values in any D-correction tables as well as to external correction values (ED-correction). While writing, incremental modifications can also be specified.

Example:

Inch access to the "L2" correction of data block 2 of the external tool correction.

Program:

```
110 DCT("L2" ,2 ,0 , "INCH")
```

DBSEA Search for data blocks within a database table.

The CPL function returns the header of the first data block that corresponds to the search condition. In this case, the variable returns a value of 1. The data block search starts with the data block defined using <StartKey1> and <StartKey2>. If one of the two starting keys has a value of -1, the search starts at the first data block of the database table.

Example:

Search for the tool with T-number 5 in the tool memory and save the data block that is found in structured variable "SV.Tool". Using DBSEA, only the "Header" data of a data block are read.

Program:

```
110 SV.Tool.Hd = DBSEA("/DBT1",-1,-1 , "IKQ3=5",RECFFOUND%,ERR%)
```

DBSEAX Searches for one or more data blocks in a tool database table.

The command provides a list of data block headers corresponding to the search criterion. The headers are stored in a system data array <HeaderArr> that is sorted according to K1 and K2. The parameter <ErgSize> is used to specify the maximum number of data block headers to be searched. The return value of DBSEAX provides the number of data blocks found.

DBTAB Reading or writing of data elements and data blocks

Example:

Reading the data block of sector 3, place 34 into the structured variable SV.Tool.

Program:

```
110 SV.Tool = DBTAB("/DBT1",3,34 ,ERR%)
```

DBTABX Can read a complete data set or a substructure of a tool database table into a CPL variable or write from the variable to the data set. Parameter <Mode>

can be used to control whether all data of the data block or only tool- or place-specific data are written.

Example:

1. Reading all data of the data block of sector 3, place 34 into the structured variable SV.Tool.
2. Writing all tool data of the data block of sector 3, location 34 of the structured variable "SV.Tool". No location-specific parts are applied.

Writing tool data

Program:

```
110 SV.Tool = DBTABX("DBT1.Rec",3,34)
120 DBTABX("DBT1.Rec",3,34,1,ERRNO) = SV.Tool
```

DBTABXL

Similar to DBTABX, but the data block is locked during read access; this lock is removed when the data block is written.

DBMOVE

Moving data blocks within a database table.

Example:

The contents of the data block (1,1) are moved to the data block (2,2).

Program:

```
130 DBMOVE("/DBT1",1,1,2,2)
```

DBLOAD

Via "DBLOAD", parts of a database table or a complete database table can be read from a file into the database.

Example:

The content of the data block (1,1) is read from the file "dbdaten.txt" into the database table.

Program:

```
140 DBLOAD("/DBT1",1,1,"/dbdaten.txt")
```

DBSAVE

Via "DBSAVE", parts of a database table or a complete database table can be saved to a file.

Example: The content of the data set (1,1) is saved to the "dbdaten.txt" file.

Program:

```
150 DBSAVE("/DBT1",1,1,"/dbdaten.txt")
```

PLC interfaces

General information

To access the tool database, the MTX offers the PLC function components and structures described in the following; these are summarized in PLC library "MT_MTX.lib".



For the detailed syntax of the NC command described below, please see the documentation "MTX PLC Interface". Only the names of the PLC blocks are mentioned here.

Function blocks

MT_DbData

The "MT_DbData" program component offers various functions for reading and changing tool data blocks.

MT_DbRecList

The program component "MT_DbRecList" offers the possibility to search for data blocks within a database table or to delete data blocks according to a list.

MT_DbLoad By means of this program component, a database table can be read in from an XML file in whole or in part.

MT_DbSave By means of program component "MT_DbSave", a database table can be read into a file.

Here, there are the options

- to delete the file first and then to save the data blocks

-or-

- to add the data blocks at the end of the file.

Structures

By means of the structures predefined in the PLC library "MT_MTX.lib", all data blocks and data elements of the tool database can be processed in the PLC user program. The names of the PLC structures follow the names of the data types defined in the scheme files.

The following structures currently exist:

- MT_DbRecListCond_t
- MT_DBT1Rec_t
- MT_DbT1Hd_t
- MT_DBT1Ud_t
- MT_Ed_t
- MT_EdLife_t
- MT_Geo_t
- MT_GeoExtended_t
- MT_TI_t
- MT_Wear_t

State upon delivery

Database

In the delivery state, the database tables are configured as follows:

Database size

- DBT1: 10 data blocks
- DBT2: 10 data blocks

Table division

DBT1:

- Sector 1: 5 places
- Sector 2: 5 places

DBT2:

- Sector 1: 5 places
- Sector 2: 5 places

Data schema

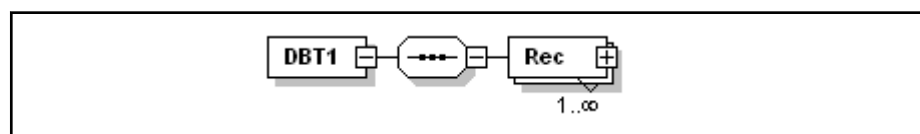


Fig. 7-127: Default configuration of the data schema (1)

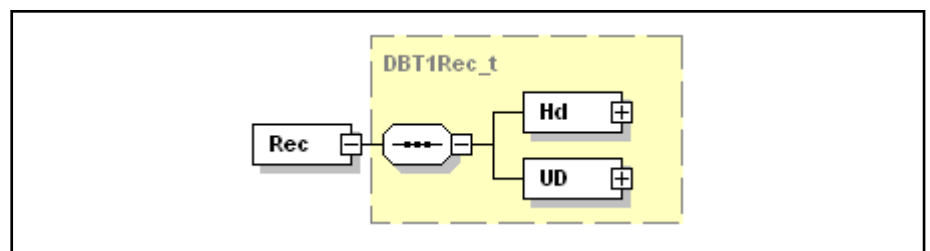


Fig. 7-128: Data scheme of default configuration (2)

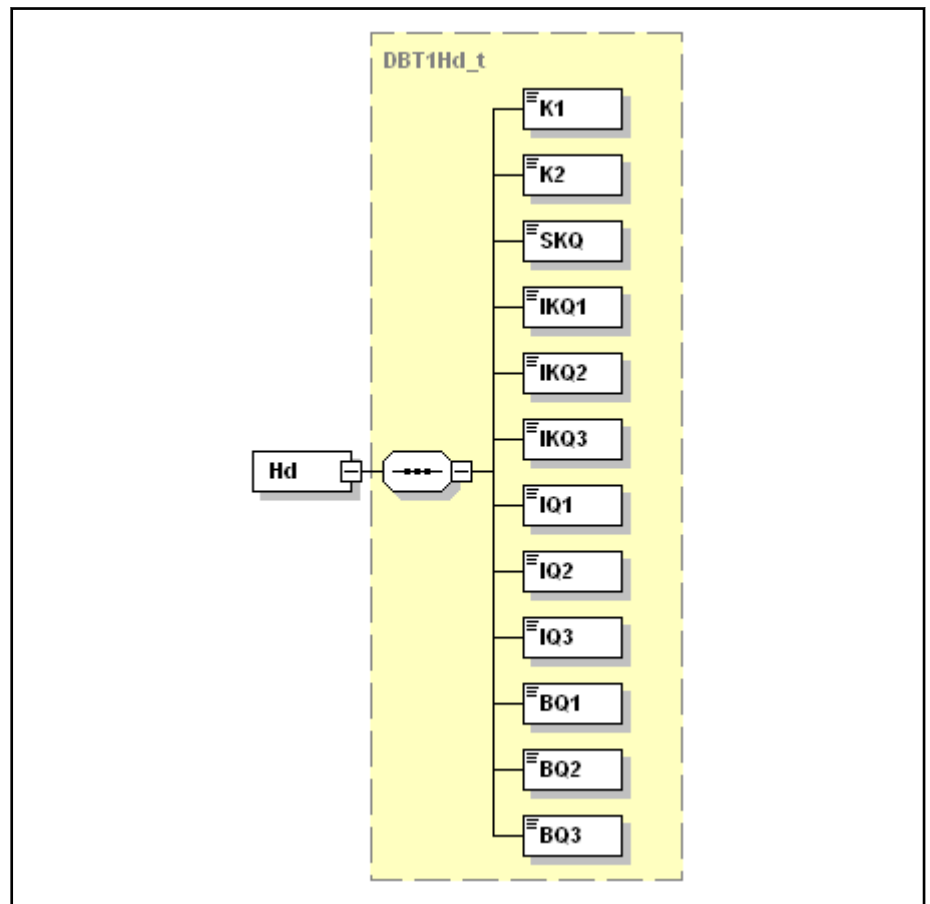


Fig. 7-129: Data scheme of default configuration (3)

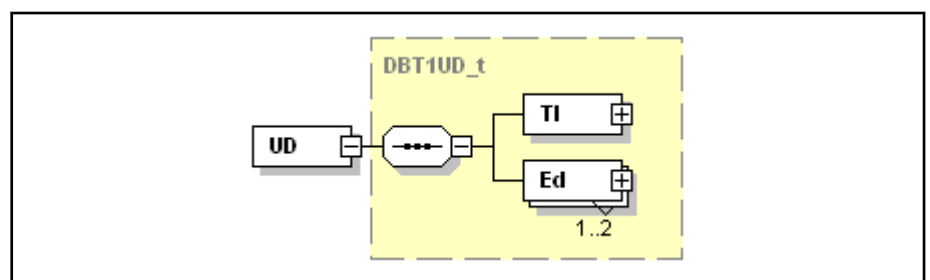


Fig. 7-130: Data scheme of default configuration (4)

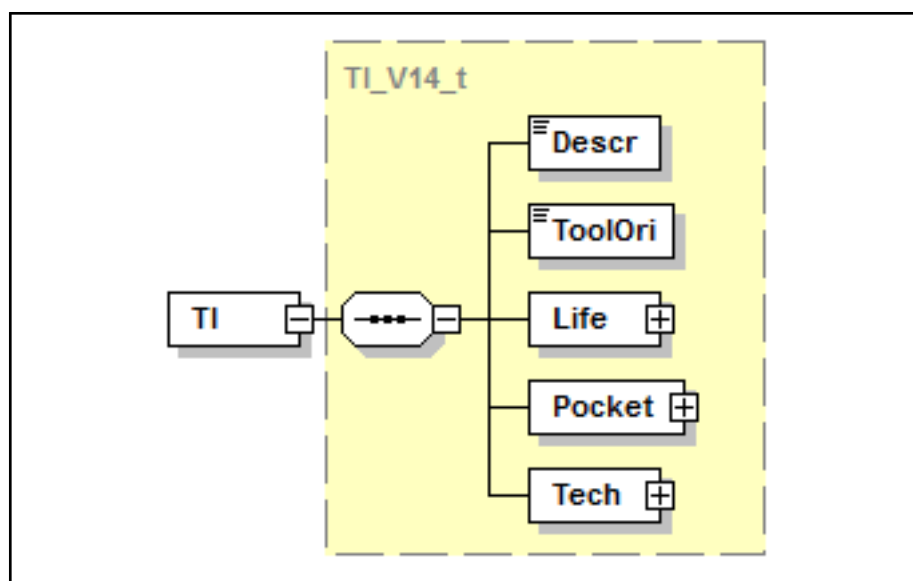


Fig. 7-131: Data scheme of default configuration (5)

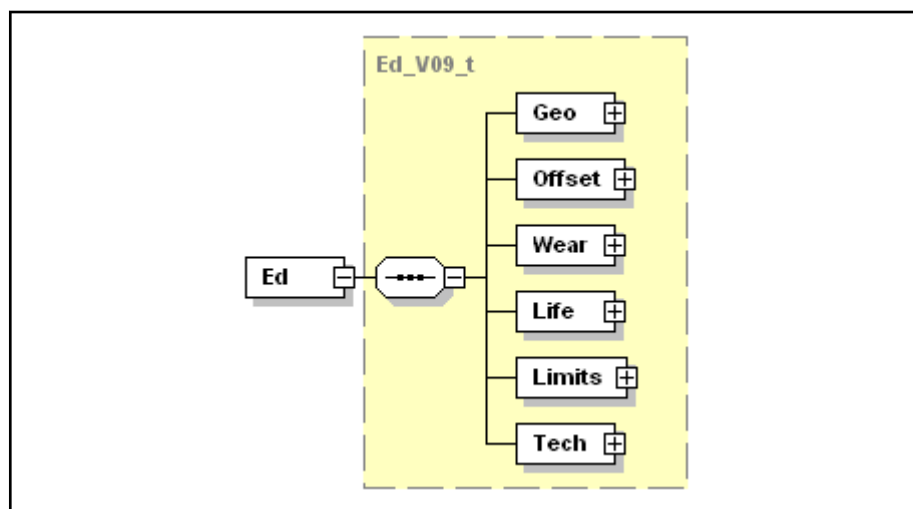


Fig. 7-132: Data scheme of default configuration (6)

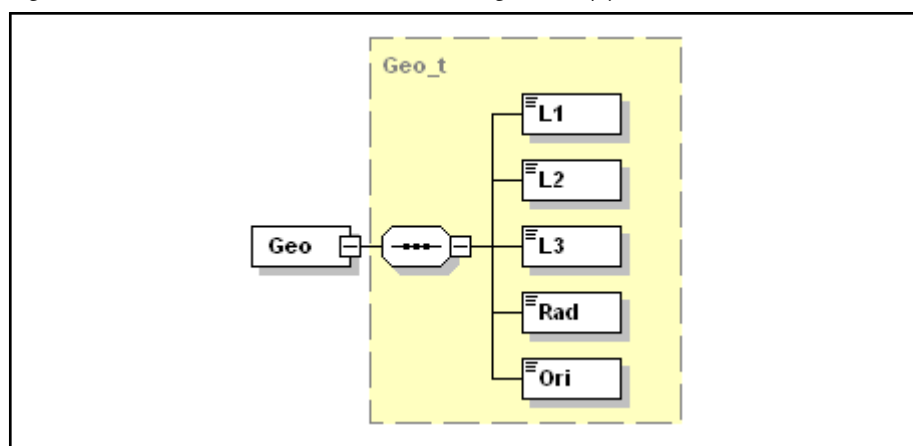


Fig. 7-133: Data scheme of default configuration (7)

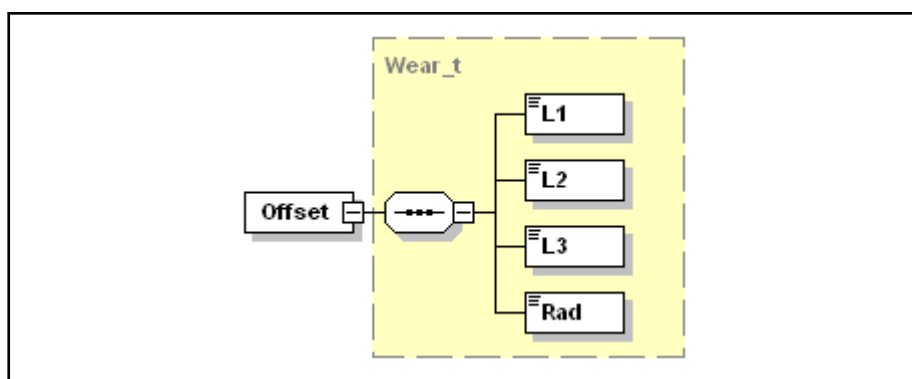


Fig. 7-134: Data scheme of default configuration (8)

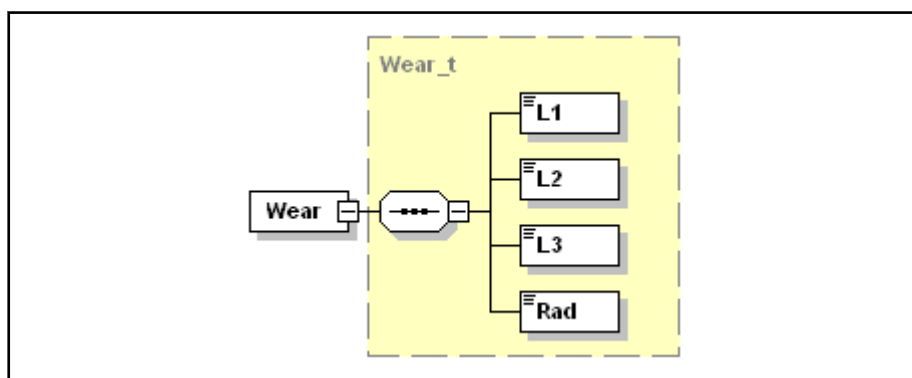


Fig. 7-135: Data scheme of default configuration (9)

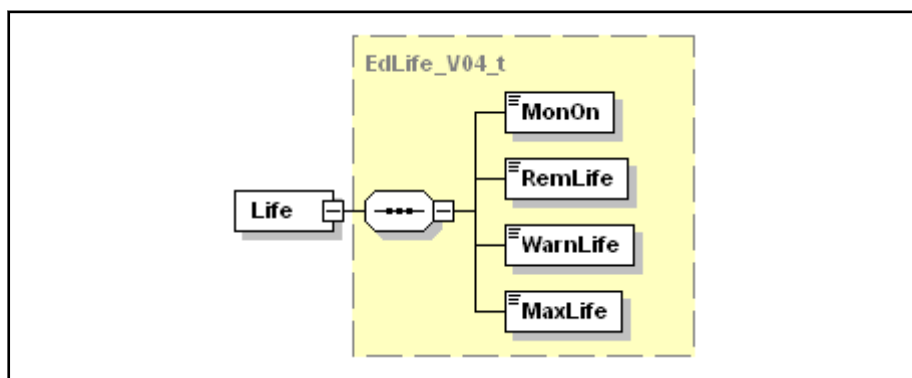


Fig. 7-136: Data scheme of default configuration (10)

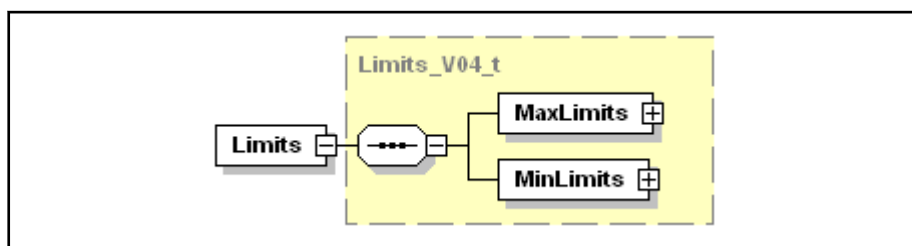


Fig. 7-137: Data scheme of default configuration (11)

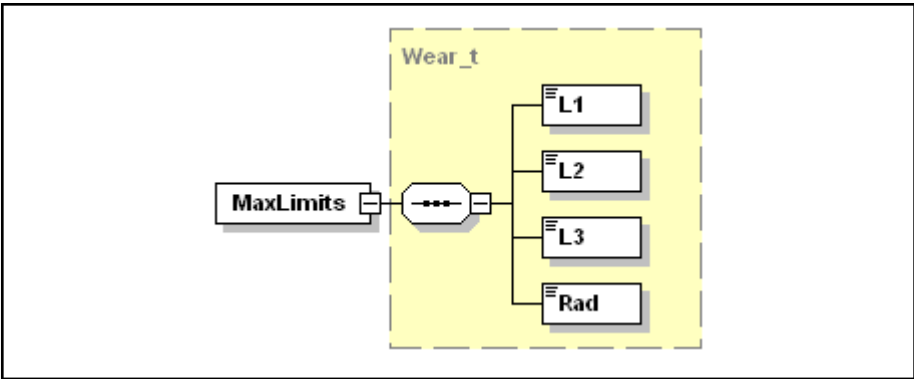


Fig. 7-138: Data scheme of default configuration (12)

Status bits

[chapter 10.3.1 "Status bits" on page 458](#)

Tool catalog

[chapter 10.3.2 "Tool catalog" on page 461](#)

Bitmap libraries

[chapter 10.3.3 "Bitmap libraries" on page 547](#)

User interface

Tool lists *The following lists are provided to the user in the supplied condition for direct use or to configure tool management:*

- Geometry list
- Wear list
- Tool life list
- Status list
- List of all worn tools (same layout as tool life list)

Geometry list	Column division:		Heading
	Sector		S
	Place		P
	Tool name		Tool name
	Duplo number		DN
	T-number		TN
	Status		Status
	Warning limit reached		tw
	Tool Worn		two
	Tool Locked		TL
	Tool edge number		SN
	Geometry		Geometry
	L1 value		L1
	L2 value		L2
	L3 value		L3
	Radius		R

Column division:	Heading
Edge position	O
Tool type	Type
Row division:	
One line per tool edge - irrelevant lines are hidden.	

Geometry data														
S	P	Tool name	DN	TN	Status			SN	Geometry					Type
					tw	two	TL		L1	L2	L3	R	O	
	1	Turning tool 4711	1	2	☐	☐	☐	1	0.000	0.000		0.000	0	
					2	0.000	0.000		0.000	0				
	2	Turning tool 4711	2	2	☐	☐	☐	1	0.000	0.000		0.000	0	
					2	0.000	0.000		0.000	0				
	1	C-drill 10	1	3	☐	☐	☐	1			0.000			
	2	C-drill 10	2	3	☐	☐	☐	1			0.000			
	3	Turn right 12.3	2	4	☐	☐	☐	1	0.000	0.000		0.000	0	
	4				☐	☐	☐							
	5				☐	☐	☐							
	6				☐	☐	☐							
	7	C-drill 10	3	3	☐	☐	☐	1			0.000			
	8	Turn right 12.3	1	4	☐	☐	☐	1	0.000	0.000		0.000	0	
	9				☐	☐	☐							
	10	angular Cut 23.7	1	6	☐	☐	☐	1			0.000	0.000		
	11				☐	☐	☐							
	12	Stepdrill 2-3-2	1	7	☐	☐	☐	1			0.000			
	13				☐	☐	☐	2			0.000			
	14				☐	☐	☐							
	15				☐	☐	☐							
	16				☐	☐	☐							
	17				☐	☐	☐							
	18				☐	☐	☐							

-10000000

<= Value <= 10000000

Geometry data

EN

Fig. 7-139: Geometry list

Wear list

Column division:	Heading
Sector	S
Place	P
Tool name	Tool name
Duplo number	DN
T-number	TN
Status	Status
Warning limit reached	tw
Tool Worn	two
Tool Locked	TL
Tool edge number	SN
Wear	Geometry
L1 value	L1
L2 value	L2
L3 value	L3
Radius	R
Row division:	
One line per tool edge - irrelevant lines are hidden.	

Control commissioning

Wear data												
S	P	Tool name	DN	TN	Status			SN	Wear			
					tw	two	TL		L1	L2	L3	R
1	1	Turning tool 4711	1	2	☐	☐	☐	1	0.000	0.000		0.000
								2	0.000	0.000		0.000
1	2	Turning tool 4711	2	2	☐	☐	☐	1	0.000	0.000		0.000
								2	0.000	0.000		0.000
2	1	C-drill 10	1	3	☐	☐	☐	1			0.000	
2	2	C-drill 10	2	3	☐	☐	☐	1			0.000	
2	3	Turn right 12.3	2	4	☐	☐	☐	1	0.000	0.000		0.000
2	4				☐	☐	☐					
2	5				☐	☐	☐					
2	6				☐	☐	☐					
2	7	C-drill 10	3	3	☐	☐	☐	1			0.000	
2	8	Turn right 12.3	1	4	☐	☐	☐	1	0.000	0.000		0.000
2	9				☐	☐	☐					
2	10	angular Cut 23.7	1	6	☐	☐	☐	1			0.000	0.000
2	11				☐	☐	☐					
2	12	Stepdrill 2-3-2	1	7	☐	☐	☐	1			0.000	
								2			0.000	
2	13				☐	☐	☐					
2	14				☐	☐	☐					
2	15				☐	☐	☐					
2	16				☐	☐	☐					
2	17				☐	☐	☐					
2	18				☐	☐	☐					

Fig. 7-140: Wear list

Tool life list

Column division:	Heading
Sector	S
Place	P
Tool name	Tool name
Duplo number	DN
T-number	TN
Status	Status
Warning limit reached	tw
Tool Worn	two
Tool Locked	TL
Tool edge number	SN
Monitoring status	Active
Tool life	Tool life[min] / no. of pieces[cyc]
Remaining tool life	Remaining
Warning limit	Warn. lim.
Maximum utilization time	Max. UT
Time unit	Unit
Row division:	
One line per tool edge - irrelevant lines are hidden.	

Monitoring data

S	P	Tool name	DN	TN	Status			SN	actv	Tool life [min] / Quantities [cyc]			Unit
					tw	two	TL			Rest	Warning Limit	max. UT	
1	1	Turning tool 4711	1	2	☐	☐	☐	1	☐	0.00	0.00	0.00	
								2	☐	0.00	0.00	0.00	
1	2	Turning tool 4711	2	2	☐	☐	☐	1	☐	0.00	0.00	0.00	
								2	☐	0.00	0.00	0.00	
2	1	C-drill 10	1	3	☐	☐	☐	1	☐	0.00	0.00	0.00	
2	2	C-drill 10	2	3	☐	☐	☐	1	☐	0.00	0.00	0.00	
2	3	Turn right 12.3	2	4	☐	☐	☐	1	☐	0.00	0.00	0.00	
2	4				☐	☐	☐		☐				
2	5				☐	☐	☐		☐				
2	6				☐	☐	☐		☐				
2	7	C-drill 10	3	3	☐	☐	☐	1	☐	0.00	0.00	0.00	
2	8	Turn right 12.3	1	4	☐	☐	☐	1	☐	0.00	0.00	0.00	
2	9				☐	☐	☐		☐				
2	10	angular Cut 23.7	1	6	☐	☐	☐	1	☐	0.00	0.00	0.00	
2	11				☐	☐	☐		☐				
2	12	Stepdrill 2-3-2	1	7	☐	☐	☐	1	☐	0.00	0.00	0.00	
								2	☐	0.00	0.00	0.00	
2	13				☐	☐	☐		☐				
2	14				☐	☐	☐		☐				
2	15				☐	☐	☐		☐				
2	16				☐	☐	☐		☐				
2	17				☐	☐	☐		☐				
2	18				☐	☐	☐		☐				

Monitoring data

EN

Fig. 7-141: Tool life list

Status list

Column division:	Heading
Sector	S
Place	P
Tool name	Tool name
Duplo number	DN
T-number	TN
Tool status	Tool status
Tool active	ta
Tool used	tu
Warning limit reached	tw
Tool worn	two
Tool locked	TL
Tool broken	TD
Tool loaded	TTL
Tool unloaded	TTU
Tool user status 1	TS1
Tool user status 2	TS2
Tool user status 3	TS3
Tool user status 4	TS4
Empty column	
Place status	Place status
Place blocked	PB
Location user status 1	PS1
Location user status 2	PS2

Control commissioning

Column division:	Heading
Location user status 3	PS3

Row division:

One line per tool edge - irrelevant lines are hidden.

Status data																				
S	P	Tool name	DN	TN	Tool status												Location status			
					ta	tu	tw	two	TL	TD	TTL	TTU	TS1	TS2	TS3	TS4	PB	PS1	PS2	PS3
1	1	Turning tool 4711	1	2	☐	☐	☐	☐	☐	☒	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐
1	2	Turning tool 4711	2	2	☐	☐	☐	☐	☐	☒	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐
2	1	C-drill 10	1	3	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐
2	2	C-drill 10	2	3	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐
2	3	Turn right 12.3	2	4	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐
2	4				☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐
2	5				☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐
2	6				☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐
2	7	C-drill 10	3	3	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐
2	8	Turn right 12.3	1	4	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐
2	9				☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐
2	10	angular Cut 23.7	1	6	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐
2	11				☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐
2	12	Stepdrill 2-3-2	1	7	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐
2	13				☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐
2	14				☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐
2	15				☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐
2	16				☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐
2	17				☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐
2	18				☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐

←

→

Status data

EN

Fig. 7-142: Status list

Tool editor The default configuration of the tool editor provides for division into 4 containers:

- *Container 1 comprises:*
 - Basic tool data control
- *Container 2 comprises:*
 - Place and tool status control
 - Tool data control
- *Container 3 comprises:*
 - Edge data control
 - Limit value data control
- *Container 4 comprises:*
 - Bitmap control (with dimensions)
 - Bitmap control (without dimensions) - only in Insert editor

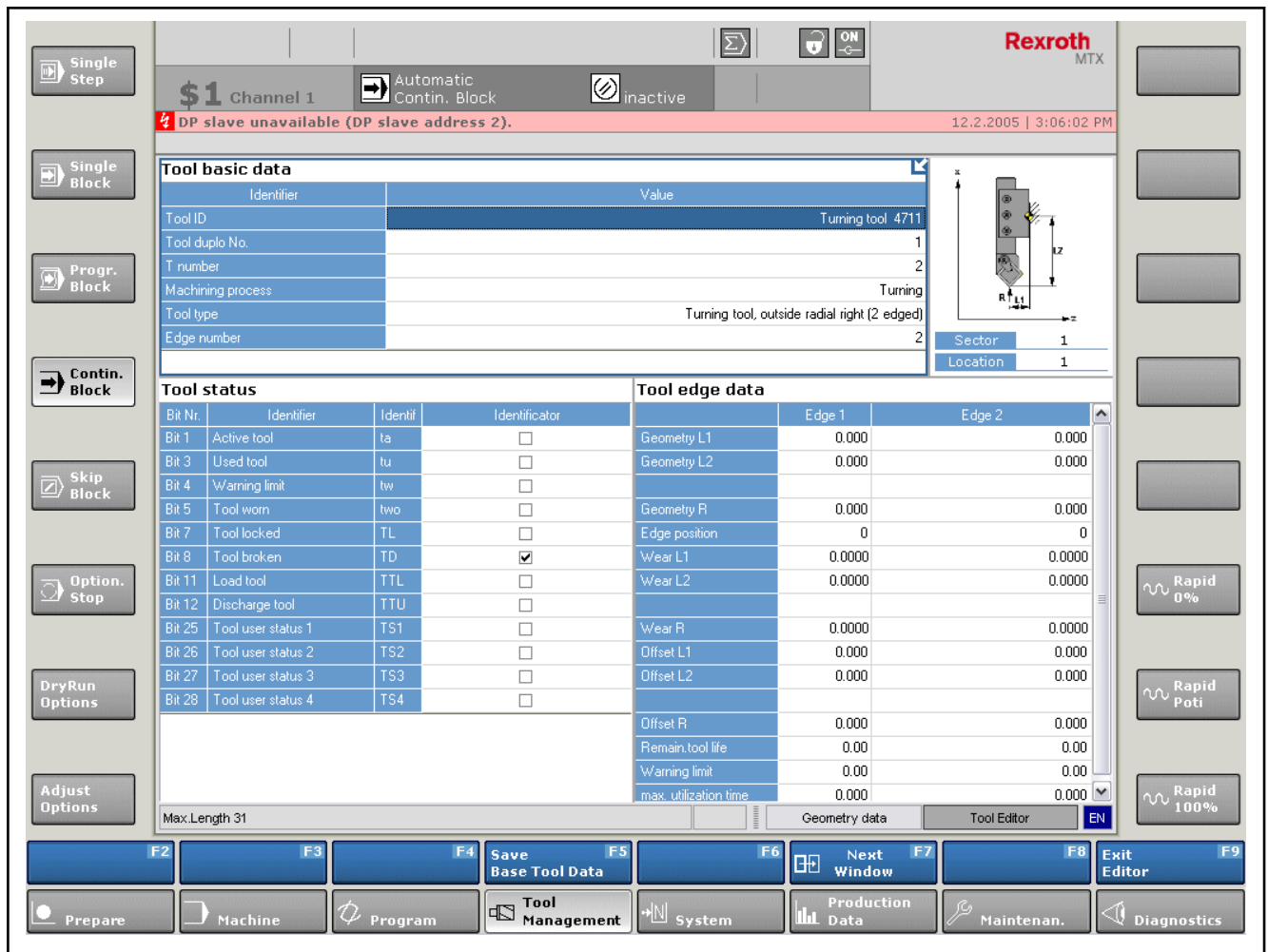


Fig. 7-143: Default tool editor configuration

7.10.8 Tool management standard tool change

General information

This document is intended for users wanting to use tool change programs at machine tools with the control MTX. The functionality of the tool change cycle TCH and how to create the functionality is explained in the following.

Due to different automatic tool changer systems such as pickup or turret tool changer, the entire sequence from the configuration of the tool database up to the tool change call in the machining program is to be displayed. The user-friendly user interface IndraWorks Operation provides several standard screens to setup the tools and to correct the tool data.



Basic knowledge about NC programming and mechanic knowledge of different automatic tool change systems are required for a better understanding.

The following topics are covered:

- Program sequence
- Different calls
- Integrate the physical tool change
- Commissioning

Operating interface

Displays in the operation area
"Tools"

By default, tool data is displayed and managed in the operating area OP5 "Tools" and OP3 "Machine".

To display and enter the tool data in a database, different list views such as wear, geometry, status and monitoring data are available for tools and tool locations in the operating area "Tools" (OP5).

Geometry data (all locations)													
S	P	Tool Name	DN	TN	Status			SN	Geometry				
					tw	two	TL		L1	L2	L3	R	O
1	1	End Mill D50	1	1	☐	☐	☐	1			100.000	25.000	
2	1	Slot mill Cutter D6	1	17	☐	☐	☐	1			100.000	3.000	
2	2	Plain mill cutter D50	1	2	☐	☐	☐	1			200.000	25.000	
2	3	Slot mill Cutter D10	1	20	☐	☐	☐	1			100.000	5.000	
2	4												
2	5	Center Drill	1	3	☐	☐	☐	1			40.000		
2	6	Tap M6	1	9	☐	☐	☐	1			100.000		
2	7	D-M-C D6	1	29	☐	☐	☐	1			65.000	3.000	
2	8	Twist Drill D5 118°	1	8	☐	☐	☒	1			10.000		
2	9												
2	10												
2	11												
2	12	Twist Drill D5 118°	2	6	☐	☐	☐	1			20.000		

Fig. 7-144: Geometry data list in the operation area "Tools"

Wear data (all locations)													
S	P	Tool Name	DN	TN	Status			SN	Wear				
					tw	two	TL		L1	L2	L3	R	
1	1	End Mill D50	1	1	☐	☒	☐	1			0.000	0.000	
2	1	Slot mill Cutter D6	1	17	☐	☐	☐	1			0.000	0.000	
2	2	Plain mill cutter D50	1	2	☐	☐	☐	1			0.000	0.000	
2	3	Slot mill Cutter D10	1	20	☐	☐	☐	1			0.000	0.000	
2	4												
2	5	Center Drill	1	3	☐	☐	☐	1			0.000		
2	6	Tap M6	1	9	☐	☐	☐	1			0.000		
2	7	D-M-C D6	1	29	☐	☐	☐	1			0.000	0.000	
2	8	Twist Drill D5 118°	1	8	☐	☐	☐	1			0.000		
2	9												
2	10												
2	11												
2	12	Twist Drill D5 118°	2	6	☐	☐	☐	1			0.000		

Fig. 7-145: Wear data list in the operating area "Tools"

Monitoring data (all locations)													
S	P	Tool Name	DN	TN	Status			SN	Active	Tool Life [min] / Quantities [cyc]			Unit
					tw	two	TL			Rem	Warning Limit	max. UT	
1	1	End Mill D50	1	1	☐	☐	☐	1	☐	0.00	0.00	0.00	min
2	1	Slot mill Cutter D6	1	17	☐	☐	☐	1	☐	0.00	0.00	0.00	min
2	2	Plain mill cutter D50	1	2	☐	☐	☐	1	☐	0.00	0.00	0.00	min
2	3	Slot mill Cutter D10	1	20	☐	☐	☐	1	☐	0.00	5.00	0.00	min
2	4												
2	5	Center Drill	1	3	☐	☐	☐	1	☐	0.00	0.00	0.00	min
2	6	Tap M6	1	9	☐	☐	☐	1	☐	0.00	0.00	0.00	min
2	7	D-M-C D6	1	29	☐	☐	☐	1	☐	0.00	0.00	0.00	min
2	8	Twist Drill D5 118*	1	8	☐	☐	☒	1	☐	0.00	0.00	0.00	min
2	9												
2	10												
2	11												
2	12	Twist Drill D5 118*	2	6	☐	☐	☐	1	☐	0.00	0.00	0.00	min

Fig. 7-146: Monitoring data list in the operation area "Tools"

Status data (all locations)																					
S	P	Tool Name	DN	TN	Tool status												Location status				
					ta	tu	tw	two	TL	TD	tl	ttu	TS1	TS2	TS3	TS4	PB	PS1	PS2	PS3	
1	1	End Mill D50	1	1	☒	☐	☐	☐	☒	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐		
2	1	Slot mill Cutter D6	1	17	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐		
2	2	Plain mill cutter D50	1	2	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐		
2	3	Slot mill Cutter D10	1	20	☐	☐	☐	☐	☐	☒	☐	☐	☐	☐	☐	☐	☐	☐	☐		
2	4																☐	☐	☐		
2	5	Center Drill	1	3	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐		
2	6	Tap M6	1	9	☐	☐	☐	☒	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐		
2	7	D-M-C D6	1	29	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐		
2	8	Twist Drill D5 118*	1	8	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐		
2	9																☐	☐	☐		
2	10																☐	☐	☐		
2	11																☐	☐	☐		
2	12	Twist Drill D5 118*	2	6	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐		
										ABS											
										Status data (all locations)											
										EN											

Fig. 7-147: Status data list in the operation area "Tools"

Description of color coding of tools in the list:

Line background color	Description
Light blue/dark blue	Cursor color/focus
Green	Tool active
Red	Tool broken/defective
Blue	Tool locked
Orange	Tool worn

Tab. 7-45: Description of color coding in the standard

The color coding can be changed by the user.

The tool editor is called via <F5>"Edit tool" or <F2>"Insert tool". All tool data and tool place data can be changed in the tool editor.

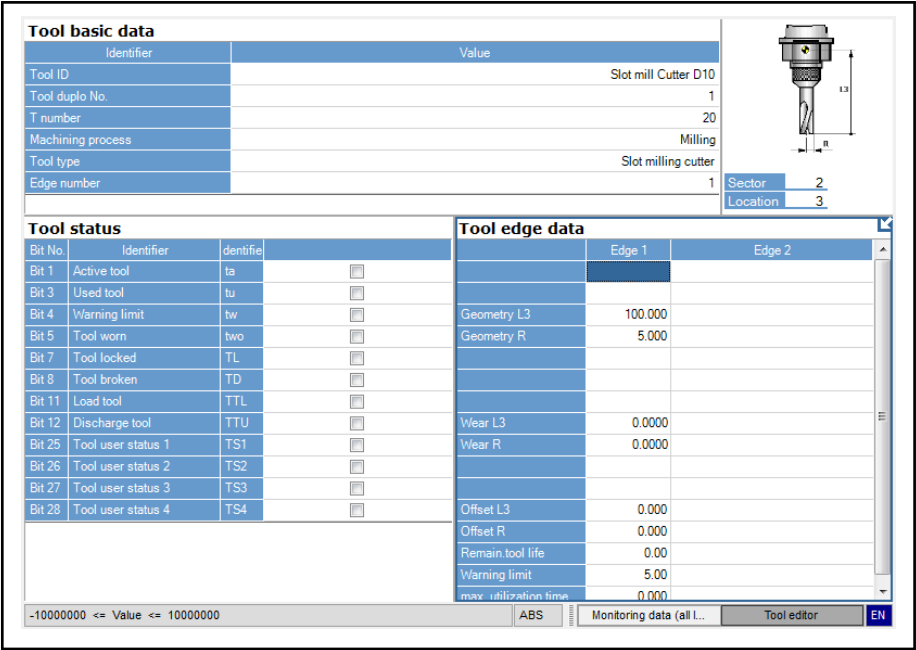


Fig. 7-148: Tool editor in operating area "Tools"

Displays in the operation area "Machine"

More standard screens for the diagnostics during setup of a workpiece show the data of the active tool and the active edge tabs, the active and preselected tool as well as the status, whether length or radius correction is activated or deactivated in the operating area "Machine" (OP3).

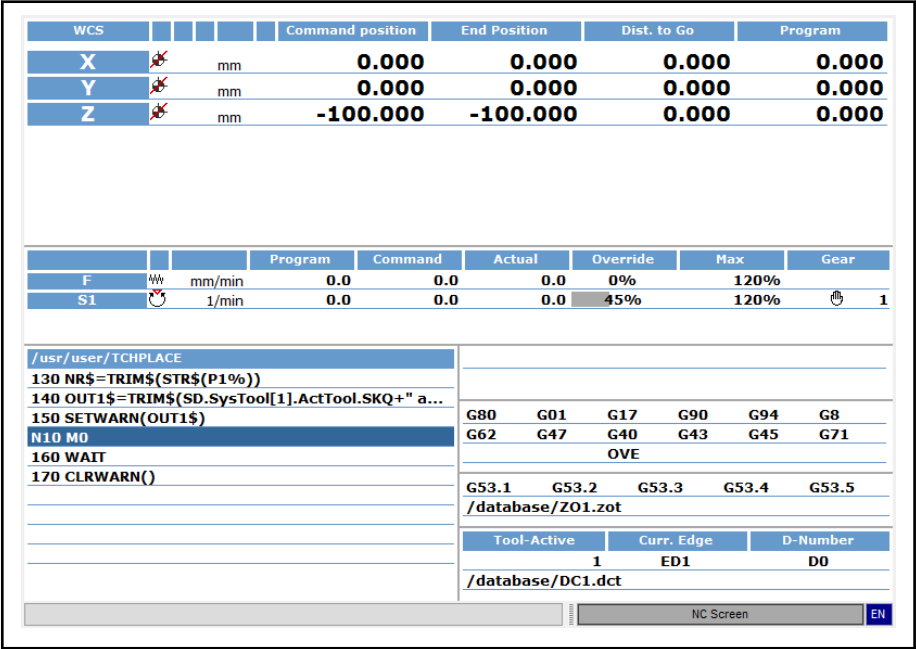


Fig. 7-149: Display of the active tool in the operating area "Machine"

ZOT bank		X	Y	Z
G53.1	Z01.zot	0.000	0.000	0.000
G53.2	--	0.000	0.000	0.000
G53.3	--	0.000	0.000	0.000
G53.4	--	0.000	0.000	0.000
G53.5	--	0.000	0.000	0.000
Total Σ	--	0.000	0.000	0.000

G47 G40		L1	L2	L3	R	Curr. Edge/D ...
Tool-Active	End Mill D50	0.000	0.000	100.000	25.000	ED1
Tool-presel.	Plain mill cutte...	--	--	--	--	--
D-Active	DC1.dct	0.000	0.000	0.000	0.000	D0
Total Σ	--	0.000	0.000	100.000	25.000	--

G47 G40		Phi [°]	Theta [°]	Psi [°]	Orient.	Type
Tool-Active	End Mill D50	0.000	0.000	0.000	0	End milling c...
Tool-presel.	Plain mill cutte...	--	--	--	--	Plain milling ...
D-Active	DC1.dct	--	--	--	0	--
Total Σ	--	0.000	0.000	0.000	--	--

NC Screen Corrections EN

Fig. 7-150: Display of all tool correction data in the operating area "Machine"

	L1	L2	L3	R	Orient.
ED01	0.000	0.000	100.000	25.000	0
ED02	0.000	0.000	0.000	0.000	0

NC Screen Corrections EN

Fig. 7-151: Display of all tool correction data tabs in the operating area "Machine"

Functional description of the tool change

System examples

Pickup automatic tool changer	
Manual - Change	Database with 2 sectors: 1. Sector with 1 spindle location 2. Sector with n magazine location Tool cabinet with n physical locations (out of which 1 is a transfer location)
Pickup changer	Database with 2 sectors: 1. Sector with 1 spindle location 2. Sector with n magazine location Magazine with n physical locations (out of which: 1 transfer location)
Pickup- Changer (Multispindle)	Database with 2 sectors: 1. Sector with 1 spindle location 2. Sector with n magazine location Magazine with 2 * n physical locations (out of which: 2 transfer locations) The tool geometries Tx and Tx* have to be identical.

Tab. 7-46: System example Pickup automatic tool changer

Turret automatic tool changer	
Turret changer	Database with 1 sector: with n magazine locations Turret magazine with n physical locations (out of which: 1 active location)

Tab. 7-47: System example Turret changer

- Properties

 - Fixed logical sequence
 - Physical process integrated and adjustable by simple NC subroutines
 - The change program is specific to the channel and can thus be used multiple times in a control
 - In Pickup mode, two database sectors are used for a channel in the tool management, in turret mode, one database sector is used
- Application

 - Manually operated tool changer
 - Automatic tool changer

- Integrated block pre-run
- Equipping the magazine is possible via spindle and magazine
- Optionally fixed-place coded tool magazine yes/no



In case of a fixed place-coded tool magazine FPC, the tool is always put down on the magazine place it has been picked up from in contrast to the free tool magazine. When equipping via the spindle, the first available position in the magazine is searched and positioned there.

This is a property of the tool magazine and not of the tool. Fixed-place coded tools also have this property.

Pickup sequence

Standard procedures during pickup tool change:

- **Case 1: Tool is already in the spindle**
 - The spindle tool is reactivated
- **Case 2: Spindle is empty (T0 active)**
 - The preselected tool is searched for
 - The tool is loaded
 - The tool data is activated
- **Case 3: T0 is programmed**
 - The active tool is replaced
 - The tool data is deactivated
- **Case 4: The new tool is unequal to the active tool**
 - The spindle tool is replaced
 - The tool data is deactivated
 - The preselected tool is searched for
 - The preselected tool is loaded
 - The tool data is activated

Turret sequence

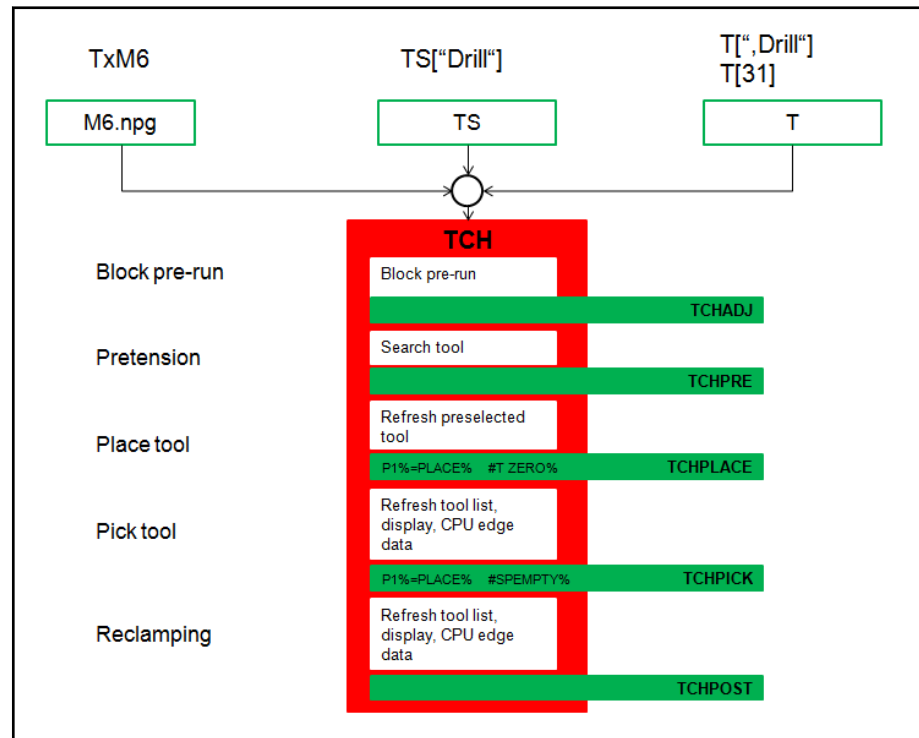
Standard procedures during turret tool change:

- **Case 1: Tool is already active**
 - The tool is reactivated
- **Case 2: No tool is active (T0 active)**
 - The preselected tool is searched for
 - The tool is loaded
 - The tool data is activated
- **Case 3: T0 is programmed**
 - The active tool is deactivated
- **Case 4: The new tool is unequal to the active tool**
 - The active tool is deactivated
 - The preselected tool is searched for
 - The preselected tool is activated

NC subroutine TCH

General information

The following graphic displays fixed logical tool changes, the integrated call of the sequences of the physical tool change as well as possible calls of the change program.



red Obligatory and cannot be changed

Green Adjustable (application-specific)

Fig. 7-152: Length of the tool change programs



The use of TCHPRE, TCHPOST and TCHADJ is optional.

TCHxxPICK and TCHxxPLACE are obligatory (xx: channel number).

A complete tool change or only a defined batch JOB (e.g. "Activate tool and refresh display") can be called.

The tool change can be used by several channels (simultaneously). That is why the tool change program uses channel-independent system variables **SD.SysTch[\$]** (\$ = channel number). (See [chapter "SD setup variables" on page 329](#))

Calling the tool change

Call via TCH

Call parameters of TCH

Optional position parameters	Description
P1%	Tool number
P2\$	Tool ID
P3%	Job number

Tab. 7-48: Significance of the TCH cycle position parameters

Syntax

TCH[<NO>] Call with tool number

or:

TCH[,"<ID>"] Call with tool identifier

or:

TCH[,"<JOBNO>"] Calling a job (see [chapter "NC subroutine TCH" on page 310](#))

If the tool number and the tool identifier are transmitted in a call, an error occurs. A corresponding message is displayed.

Examples:

Call with tool number:

Program:

```
N100 TCH[45] ;Replace tool with number 45
```

Call with tool identifier:

Program:

```
N100 TCH[,"Drill D10"] ;Tool "Drill D10" is replaced
```

Optional calls

For the provision of optional tool change calls, please refer to chapter [Settings: Optional Calls, page 336](#).

T is subroutine:**Syntax**

T[<NO>,"<ID>"]
]
Examples:

```
N100 T[45] ; Replace tool with number 45
N110 ... ; Machining
N200 T[,"Drill D10"] ; Replace "Drill D10"
N210 ... ; Machining
N ...
N300 M30
```

TS is subroutine:**Syntax**

TS["<ID>"]
Examples:

```
N200 TS["Drill D10"] ; Replace "Drill D10"
```

T is the auxiliary function, M6 is the subroutine:**Syntax**

T<NO> M6
Examples:

N100 T45 M6 ; Replace tool with number 45

Physical process

To realize different physical processes (e.g. axis identifier, axis positions, auxiliary functions etc.), the user can program these in the following 4 sub-routines.

Tool header TCHPRE

Task: Tool header

By specifying a program name in the **SD.SysTch[\$].SETUP.PREPRG** (\$ = channel number) variable, the user can integrate a subroutine as leader.

In the leader, axes that are not required can be traveled to a safe, collision-free positions or automatic coolant or probe switch offs can be programmed.



TCHPRE is not executed if the tool is already in the spindle.

Traversing motion to put the tool down TCHPLACE

Task: Put tool down

Steps:	Condition:	Variable assignment:

Bring place to put tool down to transfer position		P1%=MAGPLACE%

Preposition XY		

Move in Y		

Clamp magazine tool (WZ)		

Release tool spindle		

Retract Z		

Move out Y	In case of follow-on tool:	SD.SysTch[\$].EXTERNAL.TOOL_FOL=FALSE

Traversing motion to pick tool up TCHxxPICK (xx: Channel number)

Task: Pick tool up

Steps:	Condition:	Variable assignment:

Bring pick up location to transfer location		P1%=PREPLACE%

Position X,Y,Z in front of the magazine door	If the tool has not been put down before	

Move in Y	If the spindle is empty:	SD.SysTch[\$].EXTERNAL.SPEMPTY=TRUE

Pick up location z-axis		

Move out Y		

Tool trailer TCHPOST

Task: Tool trailer

By specifying a program name in the **SD.SysTch.SETUP.POSTPRG** variable, the user can integrate a subroutine as tool trailer.

In the tool trailer, states (e.g. positioning window, accelerations, etc.) that have been switched in the tool change program, can be programmed for machining.



TCHPOST is not executed, if no follow-on tool is programmed or if the tool is already in the spindle.

Examples for adjusting the physical process

SET is contained in folder SET1 to SETx and contains example programs and variable lists for different tasks, to facilitate commissioning of the system.

SET1 milling MANUAL with dialog *Tool header*

```
1000 GOTO .NOCOM
; -----
; Program name:   TCHPRE
; Version:        2.0.0.0
; Task :          Leader
; Note:
;               Is not executed if the tool
;               is already in the spindle
; Name:           JK
; TCH Version:    2.0.0.0
; Application:     Manually operated tool changer milling machine 2 sectors
; date:           01/10/14
; -----
1010 .NOCOM
```

Put tool down

```
100 GOTO .NOCOM
; -----
; Program name:   TCHPLACE
; Version:        2.0.0.0
; task:           Place tool:
; Steps variables:
; - Bring place to put tool down to transfer position P1%=MAGPLACE%
; - XY Pre-positioning
; - Move in Y
; - Clamp tool magazine
; - Release tool spindle
; - Retract Z
; - Move out Y if follow on tool SD.TCH.EXTERNAL.TOOL_FOL=FALSE
; Name:           JK
; TCH Version:    2.0.0.0
; Application:     Manually operated tool changer milling machine 2 sectors
; date:           01.10.14
; -----
110 .NOCOM Remove
120 DIM OUT1$(130) : DIM NR$(2)
130 NR$=TRIM$(STR$(P1%))
140 OUT1$=TRIM$(SD.SysTool[1].ActTool.SKQ+" from spindle and place in tool cabinet position:"+NR$+"")
150 SETWARN(OUT1$)
N10 M0
160 WAIT
170 CLRWARN()
```

Pick tool up

```
100 GOTO .NOCOM
; -----
; Program name:   TCHPICK
; Version:        2.0.0.0
; task:           Pick tool:
; steps variables:
; - Set pick up place to transfer position P1%=PREPLACE%
; - If previously the tool has not been placed down:
; - Position X,Y,Z in front of the tool magazine door
; - Move in Y SD.TCH.EXTERNAL.SPEMPTY=TRUE
; - Pick up position Z-axis
; - Move out Y
; Name:           JK
; TCH Version:    2.0.0.0
; Application:     Manually operated tool changer milling machine 2 sectors
; date:           01.10.14
; -----
110 .NOCOM
120 DIM OUT1$(130) : DIM NR$(2)
130 NR$=TRIM$(STR$(P1%))
140 Remove OUT1$=TRIM$(SD.SysTool[1].PreTool.SKQ+" from tool cabinet position:"+NR$+" and clamp in spindle")
```

Control commissioning

```

150 SETWARN(OUT1$)
N10 M0
160 WAIT
170 CLRWARN()

```

Tool trailer

```

1000 GOTO .NOCOM
; -----
; Program name:   TCHPOST
; Version:        2.0.0.0
; Task :          Trailer
;
; Note:
;               Is not executed
;               if no follow-on tool is programmed
;               or
;               if the tool is already in the spindle.
; Name:           JK
; TCH Version:    2.0.0.0
; Application:     Manually operated tool changer milling machine 2 sectors
; Date:           01/10/14
; -----
1010 .NOCOM

```

SET2 Milling AUTOMATIC without motions *Tool header*

```

1000 GOTO .NOCOM
; -----
; Program name:   TCHPOST
; Version:        2.0.0.0
; Task :          Trailer
;
; Note:
;               Is not executed
;               if no follow-on tool is programmed
;               or
;               if the tool is already in the spindle.
; Name:           JK
; TCH Version:    2.0.0.0
; Application:     Manually operated tool changer milling machine 2 sectors
; Date:           01/10/14
; -----
1010 .NOCOM

```

Put tool down

```

1000 GOTO .NOCOM
; -----
; Program name:   TCHPLACE
; Version:        2.0.0.0
; task:           Place tool:
; Steps variables:
; - Bring place to put tool down to transfer position P1%=MAGPLACE%
; - XY Pre-positioning
; - Move in Y
; - Clamp tool magazine
; - Release tool spindle
; - Retract Z
; - Move out Y if follow on tool SD.TCH.EXTERNAL.TOOL_FOL=FALSE
; Name:           JK
; TCH Version:    2.0.0.0
; Application:     Simulator milling machine 2 sectors
; date:           01/10/14
; -----
1010 .NOCOM
1020 DUMMY%=P1%

```

Pick tool up

```

1000 GOTO .NOCOM
; -----
; Program name:   TCHPICK
; Version:        2.0.0.0
; task:           Pick tool:
; steps variables:
; - Set pick up place to transfer position P1%=PREPLACE%
; - If previously the tool has not been placed down:
; - Position X,Y,Z in front of the tool magazine door

```

```
; - Move in Y SD.TCH.EXTERNAL.SPEMPTY=TRUE
; - Pick up position Z-axis
; - Move out Y
; Name: JK
; TCH Version: 2.0.0.0
; Application: Simulator milling machine 2 sectors
; date: 01/10/14
; -----
1010 .NOCOM
1020 DUMMY%=P1%
```

Tool trailer

```
1000 GOTO .NOCOM
; -----
; Program name: TCHPOST
; Version: 2.0.0.0
; Task: Trailer
; Note:
; Is not executed
; if no follow-on tool is programmed
; or
; if the tool is already in the spindle.
; Name: JK
; TCH Version: 2.0.0.0
; Application: Simulator milling machine 2 sectors
; Date: 01/10/14
; -----
1010 .NOCOM
```

SET3 Milling AUTOMATIC with motions

Tool header

```
1000 GOTO .NOCOM
; -----
; Program name: TCHPRE
; Version: 2.0.0.0
; Task: Leader
; Note:
; Is not executed
; if the tool is already in the spindle.
; Name: JK
; TCH Version: 2.0.0.0
; Application: Milling machine 2 sectors
; Date: 01/10/14
; -----
1010 .NOCOM
```

Put tool down

```
1000 GOTO .NOCOM
; -----
; Program name: TCHPLACE
; Version: 2.0.0.0
; task: Place tool:
; Steps variables:
; - Bring place to put tool down to transfer position P1%=MAGPLACE%
; - XY Pre-positioning
; - Move in Y
; - Clamp tool magazine
; - Release tool spindle
; - Retract Z
; - Move out Y if follow on tool SD.TCH.EXTERNAL.TOOL_FOL=FALSE
; Name: JK
; TCH Version: 2.0.0.0
; Application: Milling machine 2 sectors multi-spindle
; date: 01/10/14
; -----
1010 .NOCOM
N100 M19 S0 ; Spindle orientation at 0 degrees
N110 G76 G0 Z-100 ; pick position Z-axis
N120 G76 G0 X-300 Y0 ; Position X-, Y-axis in front of tool magazine door
N130 M206 ; Signal Start tool change at PLC
N140 M110 ; Open tool magazine door
N150 G76 G0 MAG[P1%] ; approach empty magazine position
N160 M224 ; Open MAG gripper
N170 WAIT
N180 G76 G0 Y200 ; Transfer position Y-axis
N190 M225 ; Close MAG gripper
```

Control commissioning

```

N200 M212 ; Open tool clamp
N210 G76 G0 Z0 ; Via tool position Z-axis
;-----
; If no follow-on tool is programmed, then:
1020 IF SD.TCH.EXTERNAL.TOOL_FOL=FALSE THEN
N220 M213 ; Close tool clamp
1030 WAIT
N230 G76 G0 Y0 ; Retract Y-axis
N240 M111 ; Close magazine cover
1030 ENDIF
;-----

```

Pick tool up

```

1000 GOTO .NOCOM
; -----
; Program name:  TCHPICK
; Version:       2.0.0.0
; task:         Pick tool:
; steps variables:
; - Set pick up place to transfer position P1%=PREPLACE%
; - If previously the tool has not been placed down:
; - Position X,Y,Z in front of the tool magazine door
; - Move in Y SD.TCH.EXTERNAL.SPEMPTY=TRUE
; - Pick up position Z-axis
; - Move out Y
; Name:         JK
; TCH Version:  2.0.0.0
; Application:  Milling machine 2 sectors and multi-spindle
; date:        01.10.14
; -----
1010 .NOCOM
N100 G76 G0 MAG[P1%] ; Position MAG axis
;-----
; If spindle is empty, then:
1010 IF SD.TCH.EXTERNAL.SP EMPTY=TRUE THEN
N110 G76 G0 X-300 Y0 Z0 ; Position X,Y,Z-axis in front of tool magazine door
N120 M213 ; Close tool clamp
N130 M19 S0 ; Spindle orientation
N140 M110 ; Open magazine door
N150 M206 ; Signal Start tool change at PLC
N160 WAIT
N170 G76 G0 Y200 ; Transfer position Y-axis
N180 M212 ; Tool clamp OPEN
1110 ENDIF
;-----
N190 G76 G0 Z-100 ; Pick position Z-axis
N200 M213 ; Close tool clamp
N210 M224 ; Open MAG gripper
1120 WAIT
N220 G76 G0 Y0 ; Position Y-axis in front of tool magazine door
N230 M225 ; Close MAG gripper
N240 M111 ; Close tool magazine door
;-----

```

Tool trailer

```

1000 GOTO .NOCOM
; -----
; Program name:  TCHPOST
; Version:       2.0.0.0
; Task:         Trailer
; Note:
;             Is not executed
;             if no follow-on tool is programmed
;             or
;             if the tool is already in the spindle.
; Name:         JK
; TCH Version:  2.0.0.0
; Application:  Milling machine 2 sectors
; Date:        01/10/14
; -----
1010 .NOCOM

```

**SET4 Turning AUTOMATIC with
motions***Tool header*

```

1000 GOTO .NOCOM
; -----

```

```

; Program name:   TCHPRE
; Version:       2.0.0.0
; Task:         Leader
;
;       Note:
;       Is not executed
;       if the tool is already in machining position.
; Name:         JK
; TCH Version:   2.0.0.0
; Application:   Turning machine 1 sector
; <date: 01/10/14
; -----
1010 .NOCOM

```

Put tool down

```

1000 GOTO .NOCOM
; -----
; Program name:   TCHPLACE
; Version:       2.0.0.0
; Task:         Place tool:
; Steps variables:
; - Bring place to put tool down to transfer position P1%=MAGPLACE%
; - XY Pre-positioning
; - Move in Y
; - Clamp tool magazine
; - Release tool spindle
; - Retract Z
; - Move out Y if follow on tool SD.SysTch.EXTERNAL.TOOL_FOL=FALSE
; Name:         JK
; TCH Version:   2.0.0.0
; Application:   Turning machine 1 sector
; date:         09.10.14
; -----
1010 .NOCOM
1 IF P1%=NUL THEN : SETERR("TCHPLACE: Error P1%") : ENDIF
;*****; Synchronize REM data transfer
1 WAIT

```

Pick tool up

```

1000 GOTO .NOCOM
; -----
; Program name:   TCHPICK
; Version:       2.0.0.0
; Task:         Pick tool:
; steps variables:
; - Set pick up place to transfer position P1%=PREPLACE%
; - If previously the tool has not been placed down:
; - Position X,Y,Z in front of the tool magazine door
; - Move in Y SD.SysTch.EXTERNAL.SPEMPTY=TRUE
; - Pick up position Z-axis
; - Move out Y
; Name:         JK
; TCH Version:   2.0.0.0
; Application:   Turning machine 1 sector
; date:         09/10/14
; -----
1010 .NOCOM
;*****
;unlock revolver
;*****
M305
TUR UNL1
WAIT
WAX(TUR)
WAIT

;*****
N100 G76 G0 TUR[P1%] ; Position MAG axis
;*****
;lock revolver
;*****
RAX(TUR)
WAIT
TUR LOC1
WAIT

;*****; Synchronize REM data transfer
1 WAIT

```

Control commissioning

Tool trailer

```

1000 GOTO .NOCOM
; -----
; Program name:   TCHPOST
; Version:       2.0.0.0
; Task:         Trailer
;
;      Note:
;      Is not executed
;      if no follow-on tool is programmed
;      or
;      if the tool is already in machining position.
; Name:         JK
; TCH Version:   2.0.0.0
; Application:   Turning machine 1 sector
; Date:         01/10/14
; -----
1010 .NOCOM

```

TCH at block pre-run**Tool adjustment**

During the calculation run in the block pre-run, the tool lengths in the respective tool are provided in the TCH program. The subroutines TCHxxPICK and TCHxxPLACE (xx: channel number) are not called.

Example:

```

TCH[1]
M8
G54 X...
...
G0 X100 Y200 Z300 M5
;Standard target position calculation run
TCH[2]
M8
;Target block
G55 X10 Y20 Z30...

```

Thus, the target position for the adjustment is the position programmed before the tool change call. If this target position would be approached, the position would be incorrect for the already adjusted tool (T2).

To avoid this, the target position calculation run has to be programmed in a subroutine called during the calculation.

**Adjusting the block pre-run
TCHADJ**

In the **TCHADJ** subroutine, a machine-fixed position is specified for the calculation run.

The name of the subroutine has to be entered in **SD.SysTch[\$].SETUP.ADJM6PRG** (see [chapter "SD setup variables" on page 329](#)).

TCHADJ:

```

G76 X-300 Y-200 Z0 ; Fixed position in the machine for TCH calculation run

```

Thus, the target position was moved during the calculation run to the tool change position in the machine by the user.

Now, the target position for the already adjusted tool (T2) is correct.

Example:

```

TCH[1]
M8
G54 X...
...
G0 X100 Y200 Z300 M5      ;Standard target position calculation run)
TCH[2]                   ;User target position calculation run
M8                       ;Target block
G55 X10 Y20 Z30...

```



The block pre-run variants apply to all tool change calls (T, TS and T with M6), as TCH is always used as central subroutine.

Calling TCH jobs

General information

Jobs are sequences integrated in the TCH that can be started by the NC or the PLC.

NC syntax:

TCH(,JobNo)

PLC:

Set variables and start ASUP

The following jobs are available:

Job 1:	Activate tool on location when releasing the turret disk
Job 2:	Activate tool on location when clamping the turret disk
Job 3:	In the function leave contour and return to contour, the changed geometry data of the tool have to be activated

NC call

In the function "leave contour and return to contour", the changed geometry data of the tool have to be activated (re-entry ASUP).

The name of the subroutine has to be entered in **SD.SysReentry.NameOfAsup**.

Reactivate tool data.

Example TCH_REENTRY:

```
;
;
; TCH_REENTRY
; 08/10/2014 JK
; Task:      Applying the tool geometry data of the active tool
;            for the function path and return to contour
; Process:   Is called automatically during leaving and returning to contour
;            if: SD.SysReentry.NameOfAsup=TCH_REENTRY
;
; 100 DIM WZL$(4) : DIM WZR$(4) : DIM WZ3R$(4)
110 WZL$=NCF("G47") : WZR$=NCF("G40") : WZ3R$=NCF("G140")
N10 G40 G48 G140
120 SETWARN("Tool length and radius are reactivated")
N20 M0
N30 TCH(,3)
N40 [WZL$][WZR$][WZ3R$]
130 CLRWARN()
```

PLC call

Homing of turret disk

Transfer turret position at reference position of the turret disk to activate an available tool or to deactivate corrections if no tool is available.

Sequence:

1. PLC writes
 - SD.SysTch[\$].INTERNAL.PLC.MagPos=1
and
 - SD.SysTch[\$].INTERNAL.PLC.JobNo=2
2. PLS starts TCHASUP

Control commissioning

Jogging of turret disk

Jogging of the turret disk in manual mode at lathe from location 4 to location 10.

Sequence:

Activate tool on location when releasing the turret disk

1. PLC writes
 - SD.SysTch[\$].INTERNAL.PLC.MagPos=4
and
 - SD.SysTch[\$].INTERNAL.PLC.JobNo=1
2. PLS starts TCHASUP
Activate tool on location when clamping the turret disk
3. PLC writes
 - SD.SysTch[\$].INTERNAL.PLC.MagPos=10
and
 - SD.SysTch[\$].INTERNAL.PLC.JobNo=2
4. PLS starts TCHASUP

Example TCHASUP:

```

;
;
; TCHASUP ---
; 08.11.2014 JK
; Task:
; Jog turret in manual mode
; Process: 1. PLC writes
; SD.TCH[1].INTERNAL.PLC.MagPos
; and
; SD.SysTch[1].INTERNAL.PLC.JobNo
; 2. PLC starts TCHASUP
;
; Is started when releasing the turret disk JobNo=1 (deactivating)
; and restarted when locking the turret disk JobNo=2 (activating)
;
;
100 WAIT(,200)
110 CASE SD.TCH.INTERNAL.PLC.JobNo OF
120 LABEL 1
;130 PRN#(0,"deactivating")
N30 TCH(,,1)
N40 RAX(TUR)
N50 WAIT
    150 LABEL 2
;160 PRN#(0,"activating")
N80 TCH(,,2)
180 OTHERWISE
190 SETERR("Error SD.TCH.INTERNAL.PLC.JobNo")
200 ENDCASE
;*****
;210 PRN#(0,"Done")
;*****

```

Commissioning

General information

The system is commissioned in the steps explained in detail in this chapter:

Commissioning steps

1. Configuring the database
2. Definition of the SD setup variable
3. Provide and if required, adjust NC subroutines
4. Select optional calls

Configuring the database

Setting the size and structure of the tool data memory

The size and database division is set in the "/usrfep/dbt1prms.dat" file. The database is divided into sectors and places. A sector represents physical machine units in which the tools can be contained.



The content of the tool data set is preconfigured by default and can be adjusted, if required (see [chapter 7.10.4 "Configuring the database" on page 213](#)).

The following steps are required if the database has to be adjusted:

Configuring the database

These steps are carried out in IndraWorks Engineering.

1. Copy "dbt1prms.dat" from FEPROM to "usr\user"
2. Make changes in the copied "dbt1prms.dat" (see the following program listing)
3. Copy the changed "dbt1prms.dat" from "mnt" to "usrfep"
If there is another file with the same name, rename the file.
4. Create tar file (archiving of control data)
5. Subsequently, reimport Tar file (restoring the control data)

Now, the database size changes are applied.

Example: Pickup

Sector 1 is a tool spindle in which one tool can be loaded.

Sector 2 is a chain magazine in which 20 tools can be loaded.

dbt1prms.dat

```
; -----
;           Configuration file for database table DBT1
; -----

; general structure:
; P0 P1
; |  |
; +---+--- 2 int values
;
; P0 describes the type of object and must not be modified.
;
; ";" indicates a comment line.
; In a data line no comment is allowed.

;
; *****
;           Configuration of database table
; *****

; For modifying the number of data records in a database table all
; records are deleted first. Afterwards they are newly-created
; corresponding sector configuration.
; This is only done during startup of NC.

; Control parameter
; P0: 0
; P1: Controls creation of data records in database table
;      0 Normal operation
;        Only when database table is empty, the records are created.
;      6 Deletes all existing records and creates new records
;        corresponding sector configuration.
0 0

; Sector configuration
```

Control commissioning

```
; P0: 1-99 (equals sector 1 - 99)
; P1: Number of places within sector
1 12
```

Example: Turret Sector 1 is a circular magazine in which 12 tools can be loaded.
dbt1prms.dat

```
; -----
;          Configuration file for database table DBT1
; -----

; general structure:
; P0 P1
; |  |
; +---+--- 2 int values
;
; P0 describes the type of object and must not be modified.
;
; ";" indicates a comment line.
; In a data line no comment is allowed.

;
; *****
;          Configuration of database table
; *****

; For modifying the number of data records in a database table all
; records are deleted first. Afterwards they are newly-created
; corresponding sector configuration.
; This is only done during startup of NC.

; Control parameter
; P0: 0
; P1: Controls creation of data records in database table
;      0 Normal operation
;      Only when database table is empty, the records are created.
;      6 Deletes all existing records and creates new records
;      corresponding sector configuration.
0 0

; Sector configuration
; P0: 1-99 (equals sector 1 - 99)
; P1: Number of places within sector
1 12
```

Setting the maximum number of edges

The default tool data block was defined for up to two cutting edges.

To change this number, proceed with the following configuration steps:

1. Adjustment of the tool data scheme
2. Adjustment of system data
3. Adjustment of the tool list configuration
4. Adjustment of the PLC

Settings in the tool data schema

The number of tool edges can be changed using the following entry "maxOccurs" in the scheme file "dbt1ud.xsd":

Configuration of the tool data scheme

The maximum number of edges is changed from 2 to 3. Standard tool data set as example.

1. Search the file "dbt1ud.xsd" in the following directories using the following sequence:
 1. \Root\schemas
 2. \usrfep\schemas
 3. \feprom\schemas

2. If the file found is located either in the directory "\usrfep\schemas" or "\feprom\schemas", copy it to "usr\user".
3. Change the file "dbt1ud.xsd" (refer to the following program listing)

Example:

Number of tool edges = 3:

Program:

```
<xs:complexType name="DBT1UD_t">
  <xs:sequence>
    <xs:element name="Tl" type="Tl_t"/>
    <xs:element name="Ed" type="Ed_t" maxOccurs="3"/>
  </xs:sequence>
</xs:complexType>
```

4. Copy the changed file "dbt1ud.xsd" from "usr\user" to "usrfep\schemas". If there is another file with the same name, rename the file. If the directory "schemas" does not yet exist, create it.
 5. Create tar file (archiving of control data)
 6. Subsequently, reimport Tar file (restoring the control data)
- Now, the database changes are applied with regard to the maximum number of edges.

Setting in system data

To correctly display the active tool corrections in the operating area "Machine", modify the following system setting in the file **SDDat.xml**.

Configuring the system data

The maximum number of edges is changed from 2 to 3. Standard system data definition as example.

1. Search the file "SDDat.xml" in the following directories using the following sequence:
 1. \Root
 2. \usrfep
 3. \feprom
2. If the file found is located either in the directory "\usrfep\schemas" or "\feprom\schemas", copy it to "usr\user".
3. Open the file "SDDat.xml"
4. Make changes in the file "SDDat.xml" (refer to the following program listing)

Example:

Number of tool edges = 3:

Program:

```
...
<SysMaxEd>
  <MaxEd>3</MaxEd>
</SysMaxEd>
...
```

5. Copy the changed file "SDDat.xml" from "usr\user" to "usrfep"
6. Create tar file (archiving of control data)
7. Subsequently, reimport Tar file (restoring the control data)

Now, the system data changes are applied with regard to the maximum number of edges.

Adjustment of the tool list configuration

The required additional edges have to be considered by adding a respective number of lines in the tool lists displaying edge data.

There are two different methods.

Potential solutions 1:

If the standard tool data set is used with a the standard configuration, the easiest option is to use the predefined list configurations in the project template to show tools with up to 16 edges.

Inserting and activating pre-defined list configurations

1. Select "Add" from the context menu **DBT1 screens (tool management) ► Form configuration ► List controls**. Then, right-click on **Add....**

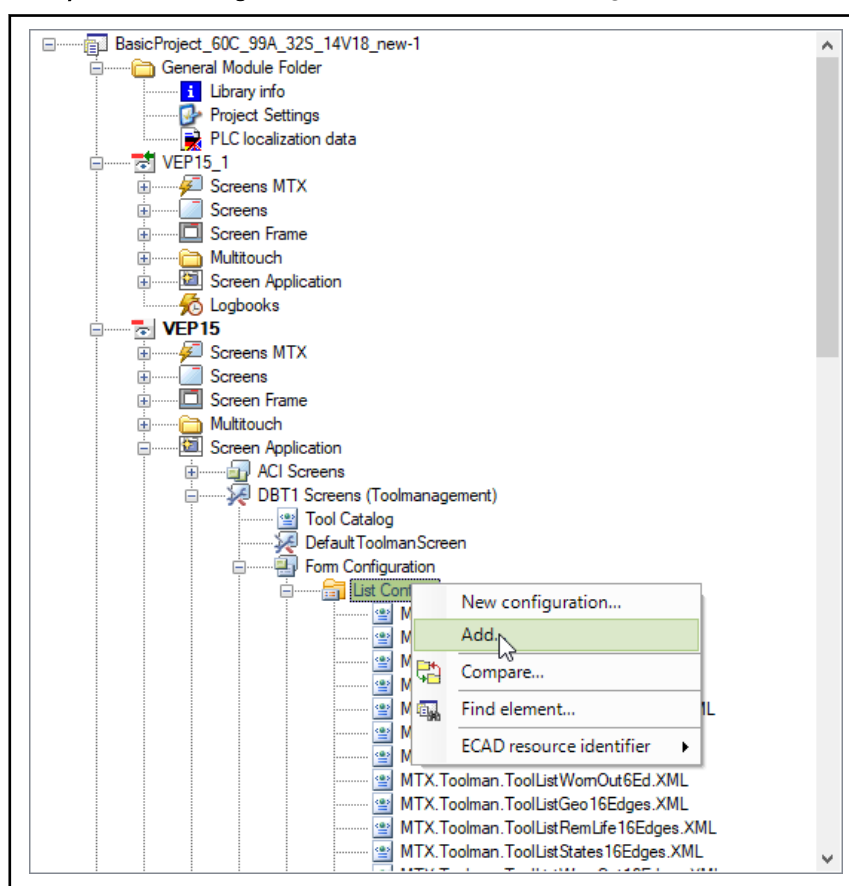


Fig. 7-153: "Add" command of a list configuration

A dialog is displayed to select the configuration file.

2. The following configuration files are subsequently selected and added:
 - MTX.Toolman.ToolListRemLife16Edges.XML
 - MTX.Toolman.ToolListGeo16Edges.XML
 - MTX.Toolman.ToolListWornOut16Edges.XML
3. Press <Ctrl+S> to save
4. In the list configuration, the entry "Tool lists configuration file" is replaced by a respective new list configuration for each configured list after **DBT1 screens (tool management) ► List configuration ► Open**.

5. To open the M-key configurator for the "MtxToolList_L" file, go to **Screen frame ► M-panels ► MtxToolList_L ► Open**
 6. Modify the "CommandString" entry to change the key configuration. Select the respectively matching newly added configuration file.
 7. Press <Ctrl+S> to save
- The configuration is completed.



To apply the changes, go to **Visualization data ► Transmit and enable...** on the node of the visualization device in the user interface.

Potential solutions 2:

If own list configurations are used, e.g. for a modified tool data set, they have to be adjusted.

Adjustment of the tool list display with regard to an extendable maximum number of edges

*Duplication of the tool list configuration in the standard project under **DBT1 screens (tool management) ► Form configuration ► List controls***

1. Open the tool list configuration
2. Inserting a respective number of rows (below). The number depends on the future increase of the maximum number of edges

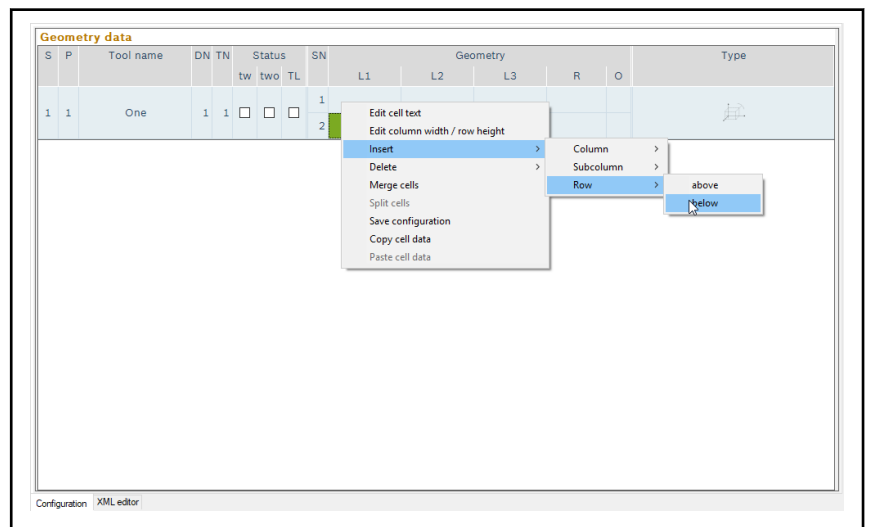


Fig. 7-154: Configuration step when increasing the maximum number of edges from 2 to 6

3. Merge cells of the columns without edge-dependent data (e.g. header columns)
4. Adjust the columns of all further contents respectively. In cells of row 2, it is recommended to use the function "Copy cell data" column by column and to use the function "Paste cell data" in the undefined new line. Then, adapt the process variable definition in the cell configuration, e.g. in the columns "L1", "L2", "L3", "R" and "O" with regard to the edge number

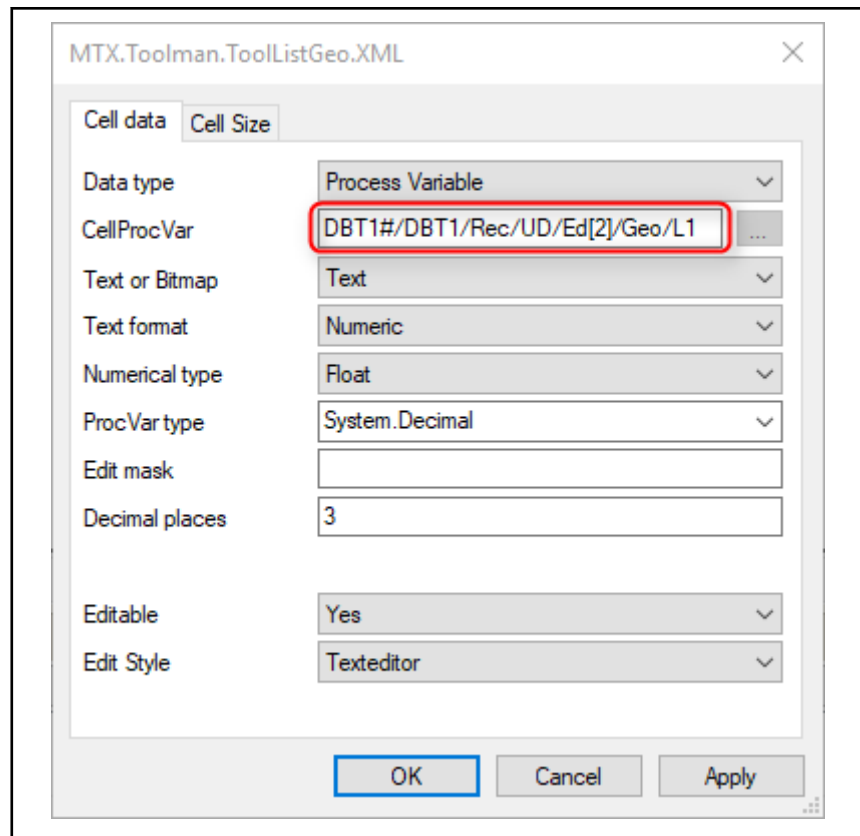


Fig. 7-155: Dialog to adjust the process variable definition

Example of a process variable definition

Program:

DBT1#/DBT1/Rec/UD/Ed[3]/Life/RemLife

The fixed comments to display the edge number have to be adapted as well as in the newly added rows in the "SN" column.

The maximum number of rows in the list corresponds to the maximum number of edges possible.

5. Open der site "XML editor" in the list configurator
6. Change the value for the node **ListControlDefinition** ► **SubListDefinitions** ► **RowDefinition** ► **SubRowDefinitions(n)** ► **SubRowHeight** to = 32 (row height of the already available rows). Repeat this for each new row/node.

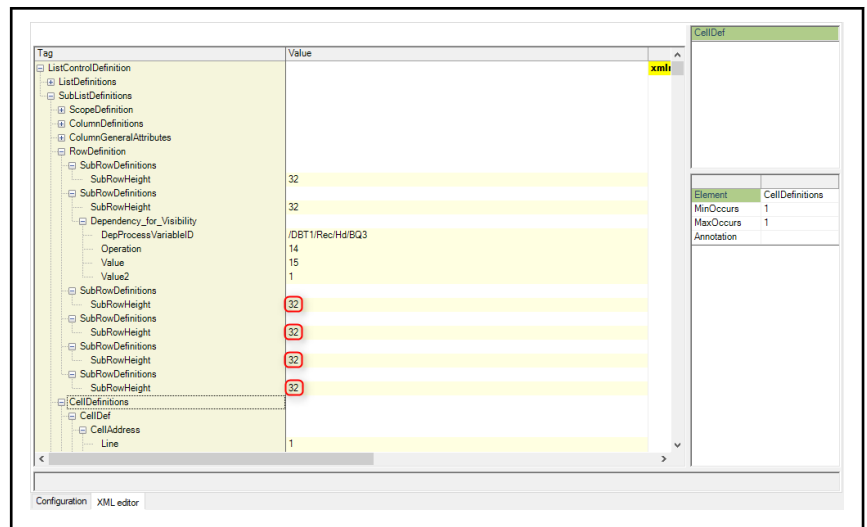


Fig. 7-156: Result of configuration step 6 after increasing the maximum number of edges from 2 to 6

7. Copy the subnode "Dependency_for_Visibility" via **ListControlDefinition** ▶ **SubListDefinitions** ▶ **RowDefinition** ▶ **SubRowDefinitions(2)** ▶ **Dependency_for_Visibility** ▶ **Copy node** and insert this subnode below the new node "SubRowDefinitions(n)" via **ListControlDefinition** ▶ **SubListDefinitions** ▶ **RowDefinition** ▶ **SubRowDefinitions(n)** ▶ **Dependency_for_Visibility** ▶ **Insert node**. Repeat this for each new row/node.

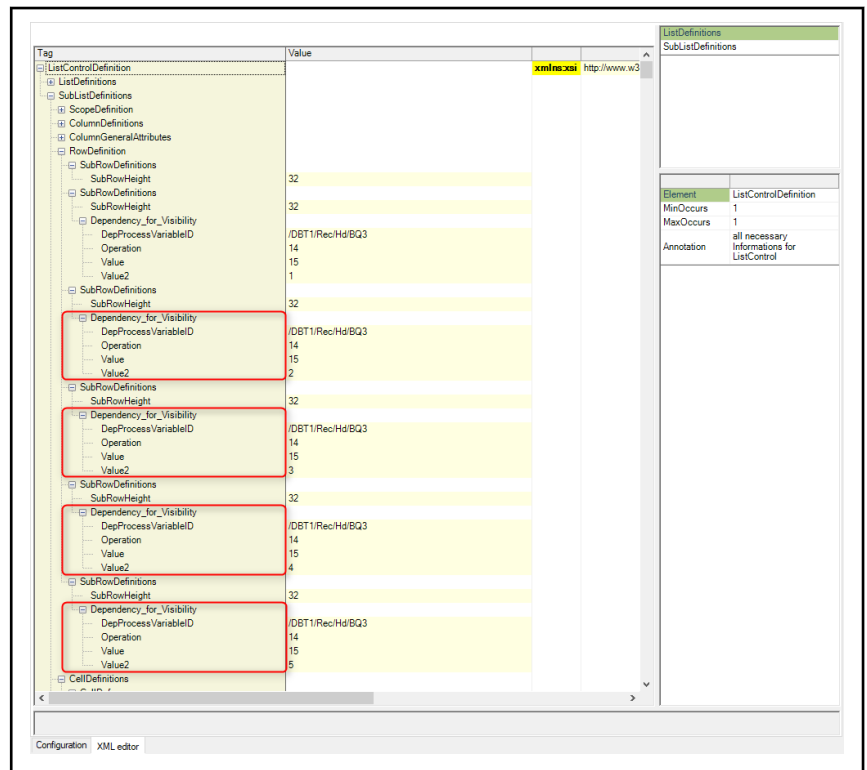


Fig. 7-157: Result of configuration step 7 after increasing the maximum number of edges from 2 to 6

8. Change the value for the node "Value2" to = n-1. n is the node index of "SubRowDefinitions". Repeat this for each new row/node.

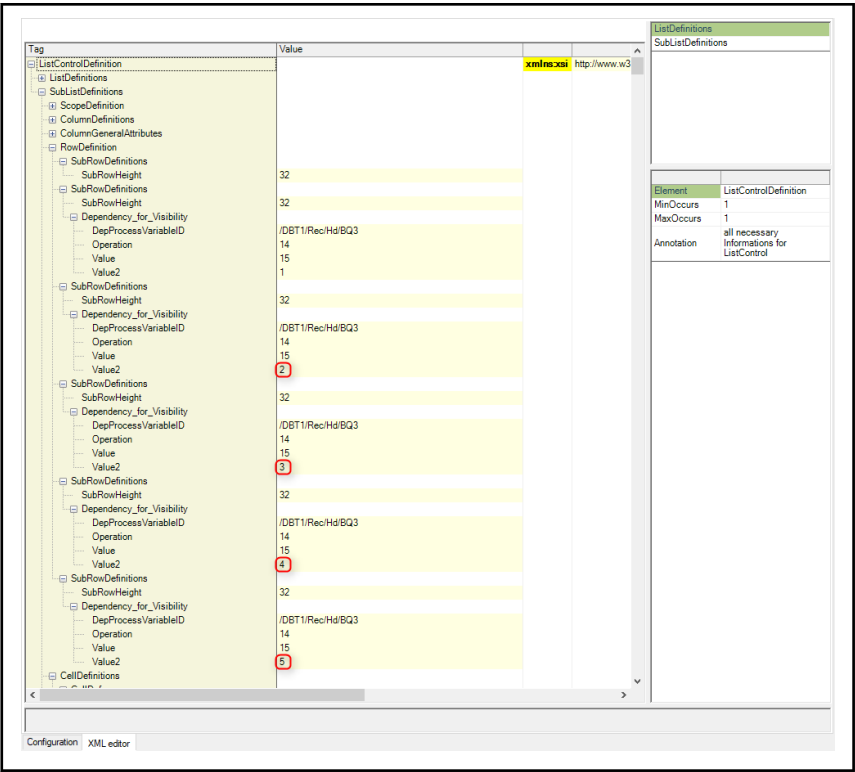


Fig. 7-158: Result of configuration step 8 after increasing the maximum number of edges from 2 to 6

The configuration is completed.

Geometry data														
S	P	Tool name	DN	TN	Status			SN	Geometry					Type
					tw	two	TL		L1	L2	L3	R	O	
1	1	One	1	1	☐	☐	☐	1						
								2						
								3						
								4						
								5						
								6						

Configuration

XML editor

Fig. 7-159: Result of configuration adjustments after increasing the maximum number of edges from 2 to 6. Geometry list as example



To apply the changes, go to **Visualization data ▶ Transmit and enable...** on the node of the visualization device in the user interface.

Setting in the PLC

If the function block "MT_TmToolLifeControl" monitors the tool life in the PLC, the changeable PLC constant "MT_MAX_EDGE" has to be adjusted corre-

spondingly. This PLC constant is part of the "MT_Base" library and it is in the "Constants/MT_ToolManagement_Const" folder.

SD setup variables

The variable elements of the structured variables of type TCH_t, required for the program TCH is summarized in the scheme "sdtch.xsd" as individual SD variable type. The SD.SysTch variable is channel-dependent and has the substructures SETUP, EXTERNAL and INTERNAL.

The SETUP substructure is used to

- select the automatic tool changer type, the number of spindles and the position offset for multiple spindle machines
- to assign the sectors for spindles and magazines
- select the magazine type FPC
- define the 3 search conditions for the tools as well as the program names for tool header and tool trailer

Name	Type	Length	Description
SD.SysTch[\$].SETUP.MODE	INTEGER		Automatic tool changer type:
			1=Pickup
			2= Turret
SD.SysTch[\$].SETUP.SPINDLES	INTEGER		Number of spindles
SD.SysTch[\$].SETUP.SPOFFSET	INTEGER		Position offset multiple-spindle
SD.SysTch[\$].SETUP.SPISEC	INTEGER		Sector number spindle
SD.SysTch[\$].SETUP.SPIPLA	INTEGER		Place number of the spindle
SD.SysTch[\$].SETUP.MAGSEC	INTEGER		Sector number magazine
SD.SysTch[\$].SETUP.MAGFIX	BOOL		FPC magazine
SD.SysTch[\$].SETUP.SEARCH1	STRING	21	Search condition 1
SD.SysTch[\$].SETUP.SEARCH2	STRING	21	Search condition 2
SD.SysTch[\$].SETUP.SEARCH3	STRING	21	Search condition 3
SD.SysTch[\$].SETUP.PREPRG	STRING	16	Program name of tool header (optional)
SD.SysTch[\$].SETUP.POSTPRG	STRING	16	Program name of tool trailer (optional)
SD.SysTch[\$].SETUP.ADJM6PRG	STRING	16	Program name Block pre-run (optional)

\$: Channel number

Tab. 7-49: *SETUP system data for tool change*

The substructure EXTERNAL is provided by the TCH program to structure program sequences in the TCHxxPICK and TCHxxPLACE (xx: channel number) user programs.

Name	Type	Description
SD.SysTch[\$].EXTERNAL.TOOL_FOL	BOOL	Follow-on tool
SD.SysTch[\$].EXTERNAL.SP_EMPTY	BOOL	Spindle is empty

\$: Channel number

Tab. 7-50: *EXTERNAL system data for tool change*

Control commissioning

Variable Editor (/mnt/Toolmanagement_en/Set1 Pickup No_Axis_2SHandChanger/Set1_Milling.var)

Name	Value	Comment
SD.SysTch[1].SETUP.MODE	1	1= Milling 2=Turning
SD.SysTch[1].SETUP.SPINDLES	1	Number of spindles MODE Milling
SD.SysTch[1].SETUP.SPOFFSET	0	Place offset when number of spindles > 1
SD.SysTch[1].SETUP.SPISEC	1	Database sector spindle tool
SD.SysTch[1].SETUP.SPIPLA	1	Database Place spindle tool
SD.SysTch[1].SETUP.MAGSEC	2	Database Sector tool magazine
SD.SysTch[1].SETUP.MAGFIX	true	Fixed bin location coded Magazine
SD.SysTch[1].SETUP.SEARCH1	0:208	Tool search condition1
SD.SysTch[1].SETUP.SEARCH2	0:220	Tool search condition2
SD.SysTch[1].SETUP.SEARCH3	16:208	Tool search condition3
SD.SysTch[1].SETUP.PREPRG		Program name leader
SD.SysTch[1].SETUP.POSTPRG		Program name trailer
SD.SysTch[1].SETUP.ADJM6PRG		Program name Block search

13 / 13 NC File System Variable Editor EN

Fig. 7-160: Example setup for milling machine "single-spindle machine"

Variable Editor (/mnt/Toolmanagement_en/Set3 Pickup XYZS_MAG_2S/Set3_MillingMS.var)

Name	Value	Comment
SD.SysTch[1].SETUP.MODE	1	1= Milling 2=Turning
SD.SysTch[1].SETUP.SPINDLES	2	Number of spindles MODE Milling
SD.SysTch[1].SETUP.SPOFFSET	4	Place offset when number of spindles > 1
SD.SysTch[1].SETUP.SPISEC	1	Database sector spindle tool
SD.SysTch[1].SETUP.SPIPLA	1	Database Place spindle tool
SD.SysTch[1].SETUP.MAGSEC	2	Database Sector tool magazine
SD.SysTch[1].SETUP.MAGFIX	true	Fixed bin location coded Magazine
SD.SysTch[1].SETUP.SEARCH1	0:208	Tool search condition1
SD.SysTch[1].SETUP.SEARCH2	0:220	Tool search condition2
SD.SysTch[1].SETUP.SEARCH3	16:208	Tool search condition3
SD.SysTch[1].SETUP.PREPRG	TCHPRE	Program name leader
SD.SysTch[1].SETUP.POSTPRG	TCHPOST	Program name trailer
SD.SysTch[1].SETUP.ADJM6PRG	TCHADJ	Program name Block search

1 / 13 NC File System Variable Editor EN

Fig. 7-161: Example setup for milling machine "multi-spindle machine"

Variable Editor (/mnt/Toolmanagement_en/Set4 Turret/Set4_Turning.var)		
Name	Value	Comment
SD.SysTch[1].SETUP.MODE	2	1= Milling 2=Turning
SD.SysTch[1].SETUP.SPINDLES	0	Number of spindles MODE Milling
SD.SysTch[1].SETUP.SPOFFSET	0	Place offset when number of spindles > 1
SD.SysTch[1].SETUP.SPISEC	0	Database sector spindle tool
SD.SysTch[1].SETUP.SPIPLA	0	Database Place spindle tool
SD.SysTch[1].SETUP.MAGSEC	2	Database Sector tool magazine
SD.SysTch[1].SETUP.MAGFIX	true	Fixed bin location coded Magazine
SD.SysTch[1].SETUP.SEARCH1	0:208	Tool search condition1
SD.SysTch[1].SETUP.SEARCH2	0:220	Tool search condition2
SD.SysTch[1].SETUP.SEARCH3	16:208	Tool search condition3
SD.SysTch[1].SETUP.PREPRG		Program name leader
SD.SysTch[1].SETUP.POSTPRG		Program name trailer
SD.SysTch[1].SETUP.ADJM6PRG		Program name Block search

1 / 13
NC File System
Variable Editor
EN

Fig. 7-162: Example setup for lathe



3 scans always take place. Thus, search condition 1, search condition 2 and search condition 3 must be specified.

Adjusting the search condition

The search condition 1 to 3 are defined by using the tool status bit BQ2. This is done using two numbers, separated by a colon (e.g. 16:208).

Description of the numbers:

- Number after the colon: Total number of bit significance that has to be taken into consideration in the search condition
- Number after the colon: Represents a binary mask for the number after the colon

The description and the significances are contained in the following list.

Tool status:

Bit no.	Name	Brief descr.	Description
1	ActiveTool	ta	Tool active
2	PrimaryTool	tp	Primary tool
3	UsedTool	tu	Used tool
4	WarnLimit	tw	Warning limit reached
5	WornOut	two	Tool worn
6	PlaceCoded	TPC	Fixed place-coded tool
7	Locked	TL	Tool locked
8	Defective	TD	Tool defective
9	Ignore	ti	Ignore tool
10	Measured	tm	Probe tool
11	ToLoad	ttil	Tool loaded

Bit no.	Name	Brief descr.	Description
12	ToUnload	ttu	Tool unloaded
13	MasterTool	mt	Master tool
14	ToolChange	tc	Tool change
15	SisterTool	ts	Alternate tool
16	HeavyTool	TH	Heavy tool
17	NonEditable	tne	Tool cannot be edited
25	UserStatus 1	TS1	Tool user status 1
26	UserStatus 2	TS2	Tool user status 2
27	UserStatus 3	TS3	Tool user status 3
28	UserStatus 4	TS4	Tool user status 4
29	UserStatus 5	TS5	Tool user status 5
30	UserStatus 6	TS6	Tool user status 6
31	UserStatus 7	TS7	Tool user status 7
32	UserStatus 8	TS8	Tool user status 8

Tab. 7-51: Tool status (BQ2)

Example 1 Searching for new tool that is not worn, locked or broken.

Calculation of the number after the colon

The following bits have to be considered:

Bit 5	Tool worn	2^4	16
Bit 7	Tool locked	2^6	64
Bit 8	Tool broken	2^7	128
Total number of significan- ces:			208

Calculation of the number in front of the colon

The following bits have to be considered:

Bit 5	Tool worn	2^4	FALSE	0
Bit 7	Tool locked	2^6	FALSE	0
Bit 8	Tool broken	2^7	FALSE	0
Total number of significan- ces:				0

The result for the entry in SD.SysTch[\$].SETUP.SEARCHx is **0:208**.

If the search does not return a new tool, residual machining can take place with an already worn tool (see example 2).

Example 2 The already worn tool (with the properties worn, not locked and not broken) is to be loaded again.

Calculation of the number after the colon

The following bits have to be considered:

Bit 5	Tool worn	2 ⁴	16
Bit 7	Tool locked	2 ⁶	64
Bit 8	Tool broken	2 ⁷	128
Total number of significan- ces:			208

Calculation of the number in front of the colon

The following bits have to be considered:

Bit 5	Tool worn	2 ⁴	TRUE	16
Bit 7	Tool locked	2 ⁶	FALSE	0
Bit 8	Tool broken	2 ⁷	FALSE	0
Total number of significan- ces:				16

The result for the entry in SD.SysTch[\$].SETUP.SEARCHx is **16:208**.

If the search does not return a new tool, residual machining can take place with an already worn tool.

Activating the SD variables

Creating and activating a new variable list

1. Generate a variable list under <OP3>"Program" using the variable structure SD.SysTch[\$].SETUP.
In the example, SD.SysTch[1] is used for channel1.
2. Enter the calculated values.
3. Subsequently, write these values with <F7>"Write var. values" to the control.

Optionally, use the variable list of the example project or of the templates.

This is recommended as the variable list contains supporting comments.

Providing the NC subroutines

The subroutines and variable lists are contained in the example project "Basic project with VAM and VPP".



The folders "SET1..." to "SET4..." contain example programs and variable lists for different tasks, to facilitate commissioning of the system.

The missing programs are required for the tool change and are to be copied to a user-defined directory (e.g. "/usr/mtb").

The directory has to be specified in the following machine parameter:

Macoda no.:	Name:	ID:
3080 00001	Search	NCO/FileOrg/ SrchPathSubProg

Control commissioning

Source	File name	Task	Note
For all variants:			
General information	M6.npg	Subroutine for call via M6	Setting Macoda parameters 3090 00003/4 is required
	TS	Subroutine for call via TS	
	T	Subroutine for call via T	
	Toolchange_TCH.npg	Example program for a tool change call with TCH	
	Toolchange_TSx.npg	Example program for a tool change call with TS	
	Toolchange_TxM6.npg	Example program for a tool change call with M6	
	TCHASUP	Job 1 (only for turning) Activate tool on place	Register TCHASUP in Ch\$Ini.prg with ASPSET
		Job 2 (only for turning) Activate tool on place	
	TCH_REENTRY	Job 3 Reactivate tool data	User name in SD.SysReentry.NameOfAsup defined
	TCHADJ	Block pre-run (optional)	User name in SD.SysTch[\$].SET-UP.ADJM6PRG defined
And additionally for manually operated tool changer milling machine 2 sectors:			
Set1 Pickup No_Ax- is_2SHand- Changer	TCHPICK	Traversing motion Pick tool up	Program name fixed Renaming to "TCH\$PICK" required
	TCHPLACE	Traversing motions Put tool down	Program name fixed Renaming to "TCH\$PLACE" required
	TCHPOST	Tool trailer (optional)	User name in SD.SysTch[\$].SET-UP.POSTPRG defined
	TCHPRE	Tool header (optional)	User name in SD.SysTch[\$].SETUP.PREPRG defined
	Set1_Fräsen.var	SD initialization	
Or for Simulator milling machine 2 sectors:			

Source	File name	Task	Note
Set2 Pickup No_Axis_2S	TCHPICK	Traversing motion Pick tool up	Program name fixed Renaming to "TCH\$PICK" required
	TCHPLACE	Traversing motions Put tool down	Program name fixed Renaming to "TCH\$PLACE" required
	TCHPOST	Tool trailer (optional)	User name in SD.SysTch[\$].SET- UP.POSTPRG defined
	TCHPRE	Tool header (optional)	User name in SD.SysTch[\$].SETUP.PREPRG defined
	Set2_Fräsen.var	SD initialization	
Or for milling machine 2 sectors and multi-spindle machine:			
Set3 Pickup XYZS_MAG_2S	TCHPICK	Traversing motion Pick tool up	Program name fixed Renaming to "TCH\$PICK" required
	TCHPLACE	Traversing motions Put tool down	Program name fixed Renaming to "TCH\$PLACE" required
	TCHPOST	Tool trailer (optional)	User name in SD.SysTch[\$].SET- UP.POSTPRG defined
	TCHPRE	Tool header (optional)	User name in SD.SysTch[\$].SETUP.PREPRG defined
	Set3_FräsenMS.var	SD initialization for multiple spindle machine	
	Set3_Fräsen.var	SD initialization for single-spindle machine	
Or for lathe 1 sector:			

Source	File name	Task	Note
Set4 Turret	TCHPICK	Traversing motion Pick tool up	Program name fixed Renaming to "TCH\$PICK" required
	TCHPLACE	Traversing motions Put tool down	Program name fixed Renaming to "TCH\$PLACE" required
	TCHPOST	Tool trailer (optional)	User name in SD.SysTch[\$].SET-UP.POSTPRG defined
	TCHPRE	Tool header (optional)	User name in SD.SysTch[\$].SETUP.PREPRG defined
	Set4_Drehen.var	SD initialization	

\$: Channel number

Remaining tool time counting and monitoring

The remaining tool life time is counted and monitored in the PLC.

Thus, implement the "MT_TmToolLifeControl" function block into the PLC program.

The function block calculates the tool life of the active edge of the active tool and displays the state of wear. The function block is provided in the "MT_BaseAdditional.library".

For the description of the function block "MT_TmToolLifeControl", refer to the documentation "IndraMotion MTX 14VRS PLC Interface", (DOK-MTX***-PLC*INT*V14-CORS-EN-P), chapter "[MT_BaseAdditional.library](#)".

Settings: Optional calls

T[<NO>,"<ID>"]

T is subroutine

Copy the provided subroutine "T" to the search path.



T cannot be parameterized as auxiliary function.

TS["<ID>"]

TS is subroutine

Copy the provided subroutine "TS" to the search path.

T<NO> M6

T is auxiliary function, M6 is subroutine



T has to be parameterized as auxiliary function.

Copy the provided subroutine "M6.npg" to the search path.

The following parameters have to be adjusted, if required.

1. Non-modal M-functions

To call an NC program using an M-function, a program name has to be assigned to the M-function.

Macoda no.:	Name:	ID:	Value
3090 00003	M-code number	NCP/SubProg/NonModalMFunc/MFunc /MFuncCode	6
3090 00004	Subroutine name	NCP/SubProg/NonModalMFunc/MFunc /MFuncSubProg	M6.npg

2. Auxiliary class definitions

The "T" auxiliary function is made known to the system.

1. Select a free class.
2. Editing parameters

Macoda no.:	Name:	ID:	Value
3010 00040	Enable auxiliary function	AUXF/ClassDef/Class[1..30]/EnableAux-Func	Yes
3010 00030	Name	AUXF/ClassDef/Class[1..30]/AuxFunc-Name	T
3010 00040	Channel dependency	AUXF/ClassDef/Class[1..30]/AuxFuncCh-Dep	Channel-depend-ent auxiliary function
3010 00040	Type	AUXF/ClassDef/Class[1..30]	Int
3010 00090	Acknowledgement requirement	AUXF/ClassDef/Class[1..30]/Int/AcknInt	Test "No" Machine "Yes"
3010 00080	Index	AUXF/ClassDef/Class[1..30]/Int/IndInt	0

3. Increase the number of integer auxiliary functions by 1.

Macoda no.:	Name:	ID:	Value
3010 00010	Integer auxiliary function	AUXF/NofAuxFunc/ChDep/ChDepInt	Current value +1

4. Enter the address.

Macoda no.:	Name:	ID:	Value
2060 00012	Address of integer auxiliary functions	AUXF/PlcIf/ChDep/ChIf/AddrChDepInt	Address value

3. Auxiliary group definition

The "T" auxiliary function is made known to the system.

Control commissioning

Macoda no.:	Name:	ID:	Value
3010 01000	Channel Depend- ency	GrChDep	channel-dependent auxiliary function
3010 01000	Auxiliary Function 1	GrAuxFunc[1]	T

Notes to create a user-defined tool change

When creating user-defined tool change programs, note that several settings have to be selected to guarantee a correct tool information display (cf. [chapter 7.10.8 "Tool management standard tool change" on page 303](#)). These settings are also realized in the previously described standard tool change program (TCH).

Relevant system data The following system data are described when activating or preselecting a tool:

Name	Type	Length	Description
SD.SysTool[\$].ActToolValid	BOOLEAN		Important for tool life count: 0: Data record of the SD.SysTool[\$].ActTool structure is invalid or no tool is active 1: Data record of the SD.SysTool[\$].ActTool structure is valid or a tool is active
SD.SysTool[\$].ActEd	INTEGER		Number of the active edge
SD.SysTool[\$].ActTool.K1	SHORT		Sector number of the active tool
SD.SysTool[\$].ActTool.K2	SHORT		Place number of the active tool
SD.SysTool[\$].ActTool.SKQ	STRING	32	ID of the active tool
SD.SysTool[\$].ActTool.IKQ1	INTEGER		Duplo no of the active tool
SD.SysTool[\$].ActTool.IKQ2	INTEGER		Type no. of the active tool
SD.SysTool[\$].ActTool.IKQ3	INTEGER		T.-no of the active tool
SD.SysTool[\$].ActTool.IQ1	INTEGER		IQ1- Value of the active tool
SD.SysTool[\$].ActTool.IQ2	INTEGER		IQ2- Value of the active tool
SD.SysTool[\$].ActTool.IQ3	INTEGER		IQ3- Value of the active tool
SD.SysTool[\$].ActTool.BQ1	INTEGER		Status of the active tool location
SD.SysTool[\$].ActTool.BQ2	INTEGER		Status of the active tool
SD.SysTool[\$].ActTool.BQ3	INTEGER		BQ3- Value of the active tool

Written by the NC program (TCH)

Written by the system

Data record header of the active tool

Type: DBT1Rec_t

Name	Type	Length	Description
SD.SysTool[\$].PreTool.K1	SHORT		Sector number of the preselected tool
SD.SysTool[\$].PreTool.K2	SHORT		Location number of the preselected tool
SD.SysTool[\$].PreTool.SKQ	STRING	32	ID of the preselected tool
SD.SysTool[\$].PreTool.IKQ1	INTEGER		Duplo no of the preselected tool
SD.SysTool[\$].PreTool.IKQ2	INTEGER		Type no. of the preselected tool
SD.SysTool[\$].PreTool.IKQ3	INTEGER		T-no of the preselected tool
SD.SysTool[\$].PreTool.IQ1	INTEGER		IQ1 value of the preselected tool
SD.SysTool[\$].PreTool.IQ2	INTEGER		IQ2 value of the preselected tool
SD.SysTool[\$].PreTool.IQ3	INTEGER		IQ3 value of the preselected tool
SD.SysTool[\$].PreTool.BQ1	INTEGER		Status of the preselected tool location
SD.SysTool[\$].PreTool.BQ2	INTEGER		Status of the preselected tool
SD.SysTool[\$].PreTool.BQ3	INTEGER		BQ3 value of the preselected tool

Data record header of the preselected tool
Type: DBT1Rec_t

\$: Channel number

Tab. 7-52: System data for tool displays



If the channel number is not written by the NC program, the number is automatically set by the calling channel.

This system data is only relevant for displaying in the basic screens of the "Machine" operating area if *System data* was defined as *Tool source* in the properties of the *Tool display* and the *Tool correction display*.

The following system data are to be written during the block pre-run:

Control commissioning

Name	Type	Length	Description	
SD.SysToolSRun[\$].Act-Tool.K1	SHORT		Sector number of the active tool	Data record header of the active tool Type: DBT1Rec_t
SD.SysToolSRun[\$].Act-Tool.K2	SHORT		Place number of the active tool	
SD.SysToolSRun[\$].Act-Tool.SKQ	STRING	32	ID of the active tool	
SD.SysToolSRun[\$].Act-Tool.IKQ1	INTEGER		Duplo no of the active tool	
SD.SysToolSRun[\$].Act-Tool.IKQ2	INTEGER		Type no. of the active tool	
SD.SysToolSRun[\$].Act-Tool.IKQ3	INTEGER		T.-no of the active tool	
SD.SysToolSRun[\$].Act-Tool.IQ1	INTEGER		IQ1- Value of the active tool	
SD.SysToolSRun[\$].Act-Tool.IQ2	INTEGER		IQ2- Value of the active tool	
SD.SysToolSRun[\$].Act-Tool.IQ3	INTEGER		IQ3- Value of the active tool	
SD.SysToolSRun[\$].Act-Tool.BQ1	INTEGER		Status of the active tool location	
SD.SysToolSRun[\$].Act-Tool.BQ2	INTEGER		Status of the active tool	
SD.SysToolSRun[\$].Act-Tool.BQ3	INTEGER		BQ3- Value of the active tool	

Name	Type	Length	Description	
SD.SysToolSRun[\$].Pre-Tool.K1	SHORT		Sector number of the preselected tool	Data record header of the preselected tool Type: DBT1Rec_t
SD.SysToolSRun[\$].Pre-Tool.K2	SHORT		Location number of the preselected tool	
SD.SysToolSRun[\$].Pre-Tool.SKQ	STRING	32	ID of the preselected tool	
SD.SysToolSRun[\$].Pre-Tool.IKQ1	INTEGER		Duplo no of the preselected tool	
SD.SysToolSRun[\$].Pre-Tool.IKQ2	INTEGER		Type no. of the preselected tool	
SD.SysToolSRun[\$].Pre-Tool.IKQ3	INTEGER		T-no of the preselected tool	
SD.SysToolSRun[\$].Pre-Tool.IQ1	INTEGER		IQ1 value of the preselected tool	
SD.SysToolSRun[\$].Pre-Tool.IQ2	INTEGER		IQ2 value of the preselected tool	
SD.SysToolSRun[\$].Pre-Tool.IQ3	INTEGER		IQ3 value of the preselected tool	
SD.SysToolSRun[\$].Pre-Tool.BQ1	INTEGER		Status of the preselected tool location	
SD.SysToolSRun[\$].Pre-Tool.BQ2	INTEGER		Status of the preselected tool	
SD.SysToolSRun[\$].Pre-Tool.BQ3	INTEGER		BQ3 value of the preselected tool	

\$: Channel number

Tab. 7-53: System data for block pre-run



If the channel number is not written by the NC program, the number is automatically set by the calling channel.

Identifying the active tool

To identify the active tool in the tool list, set the tool status bit "ta" for the active tool and reset it for the deactivated tool.

7.11 Motion

7.11.1 NC file system (MTX Navigator)

General information

Characteristics of the MTX Navigator

The MTX Navigator is a directory and file navigator. It visualizes the directory and file structure of the NC control with reference to the Windows Explorer.

This navigator is provided in IndraWorks Operation as well as in IndraWorks Engineering to navigate in the NC file system.

Tree + list

In the "Tree+List" characteristic, the directory structure is shown as tree on the left. The subdirectories and files are listed on the right.

This is used in IndraWorks Operation as well as in IndraWorks Engineering.

Control commissioning

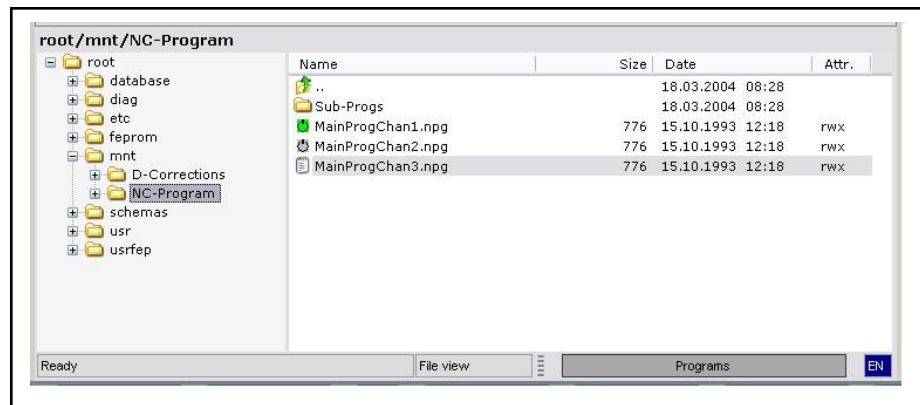


Fig. 7-163: MTX Navigator in the characteristic "Tree + List" of IndraWorks Operation

List The subdirectories and the files are listed in the "List". This representation is only used in the "program selection" ([chapter "Context-dependent functions of the MTX Navigator" on page 343](#)) in IndraWorks Operation.

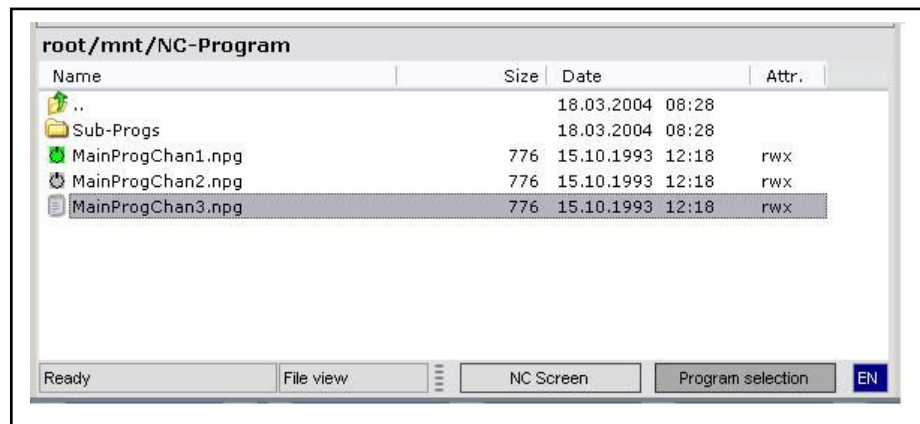


Fig. 7-164: MTX Navigator in the characteristic "List" of IndraWorks Operation

Object types Basically every following object type named in the tree as well as in the list of the MTX Navigator can be provided with its individual context menu and individual assignments of the function keys and the menu bar. This causes a change in the assignment of the function keypad and the characteristics of the menu bar or the context menu when changing between tree and list or selecting another object type.

The following object types are distinguished:

- the folder "root" in the tree  root (Root = the root directory)
- all other folders in the tree
- the return flag in the list  ..
- all other folders in the list
- all files of the list
- the list

Status bar In the status bar, the currently selected object (directory or file) is visualized.

If the focus is on a directory in the tree, the following is additionally displayed:

- the number of the objects (directories and files) and
- the sum of the sizes of all files

The sum is located directly below the selected directory.

The status bar in IndraWorks Operation is - as shown in [fig. 7-163 " MTX Navigator in the characteristic Tree + List of IndraWorks Operation" on page 342](#) - located directly below the navigator. In IndraWorks Engineering, the general status bar of IndraWorks Engineering is used.



In IndraWorks Engineering, the status display is deleted automatically after 10 seconds.

Context-dependent functions of the MTX Navigator

The MTX Navigator in IndraWorks Operation is used in two different operating areas:

- In the operating area "Program"
- In the operating area "Machine"

Additionally, the navigator is also used in IndraWorks Engineering.

According to the context, the navigator provides different functions.

Context program

The instance of this context is always activated as soon as the operating area "Program" is called in IndraWorks Operation. In this case, the navigator is the basic application for this operating area.

In this context, the navigator provides all functions for file editing.

The characteristic of the navigator is always "Tree+List" and cannot be switched.

Context: Program selection

The instance of this context is called and started with the operation mode "Automatic" via the "F2 Program Selection" function in the operating area "Machine" of IndraWorks Operation. This function is not provided in other operation modes.

In this context, the MTX Navigator provides - apart from the general functions (see [chapter "General functions" on page 345](#)) - the functionalities for the program selection.

By default, the MTX Navigator is started in this context with the characteristic "List". However, the user can switch between both characteristics "Tree+List" and "List".

Context Engineering

The instance of this context is called in IndraWorks Engineering via the "NC File System" node below the "Motion" node.

In this context, the navigator provides all functions for file editing.

The characteristic of the navigator is always "Tree+List" and cannot be switched.

Operation

General information

This section describes the different operating possibilities of the MTX Navigator.

The operation in different contexts assumes different input media. The interface is partially operated via mouse (operating areas "Program" and in IndraWorks Engineering) and partially via keyboard (operating area "Machine"). However, an operation with the keyboard is always possible. In certain areas, operation with the mouse is easier.

Context menu

For entries with an own context menu, right-click or press <Shift>+<F10> to display it.

Within the context menu, the user can highlight an entry with the mouse, the cursor buttons or by entering the underlined or the first letter and can execute

it via <Enter> or the mouse. Via <ESC> or by clicking outside the context menu, it can be closed again without executing any action.

Menu bar The menu bar can contain individual entries with regard to the currently focused object type of the MTX Navigator. The entries can be selected and activated via mouse or after pressing <F10> or <Alt>+<Letter> as for the context menus.

F-key bar The function key bar is only available in IndraWorks Operation. The function key bar is located at the lower margin of the user interface and consists of buttons which represent the function keys <F2> to <F9>. These buttons can be triggered via left-click button or by pressing the corresponding function key. The assignment of the individual buttons in the MTX Navigator depends on the currently focused object type.

Toggle between tree and list

Switch the entry focus between tree and list either by pressing <Tab> or selecting the desired object with the mouse. The switch is important, since keyboard entries always refer to the currently focused entry. To navigate in the list and if the focus is currently on the tree, only press <Tab> to focus in the list. The tree entry is then focused again with the <Tab> key.

Navigating in the tree

Keyboard or mouse can be used to navigate in the tree. Individual folders can be directly clicked on for highlighting. Click on the plus or minus sign at the left side of a folder or double-click on a folder to open or close the folder.

The cursor keys are intended to navigate via keyboard, whereas the folder currently highlighted can be expanded with <CursorRight> and collapsed with <CursorLeft>.

Folders are placed one position upwards or downwards by the cursor keys <CursorUp> and <CursorDown>.

Additionally, it can be jumped directly to the folders by entering the first sign of the node name. Ambiguities result in a jump to the next appropriate folder.

Use the keys <PgUp>, <PgDn>, <Pos1> and <End> to jump one screen page upwards or downwards or to the first or last folder in the tree.

When pressing <Backspace>, the highlighting moves to the respective superordinate folder in case the latter exists.

Navigating in the list

In the list, entries (folder, files or return label) are selected via the keys <CursorUp> and <CursorDown> or by the mouse. Use the keys <CursorLeft> and <CursorRight> to shift the list section horizontally.

Press "Enter" or double-click to trigger the predefined command of the currently highlighted entry:

- Entry = Return marker ⇒ it is switched to the next higher level
- Entry = Folder ⇒ the directory and its content is displayed
- Entry = File the function assigned to the file type (e.g. open editor) is carried out.

Change to the next higher level as shown in the tree view via <Backspace>.

Press <Ctrl>+<A> to highlight all entries of the list (multiple selection). Use the keys <Shift> and <Ctrl> together with the <Space> key, to select or deselect areas of entries or several individual entries.

Press <PgUp>, <PgDn>, <Pos1> and <End> to jump one screen page up or down or to the first or last entry in the list.

The columns of the list can be maximized or minimized by the mouse. A column can be "hidden" by completely pushing together the column with the mouse. To re-open these hidden columns later on, move the mouse from the right to the corresponding column separator and zoom out the column to the right while keeping the left mouse button pressed.

Sorting column and direction can only be changed by clicking one or multiple times on the corresponding column head. A little triangle highlights the current sorting column and indicates the selected sorting direction, whereas the apex of the triangle shows in the direction of the ascending values.



Despite the sorting, the order of the following groupings is always maintained:

- The return label is always at the top.
- Afterwards, all directories are shown.
- The last group is the group of the files.

The MTX Navigator saves all column widths as well as the sorting column and the sorting direction. This data is always saved on the hard disk when the navigator is left and exited.

Functions

General functions



General functions do not depend on the context!

Placing file filter

Use the filter dialog to exclude certain data types from the display in the list. Thus, the selection of the files can be limited.

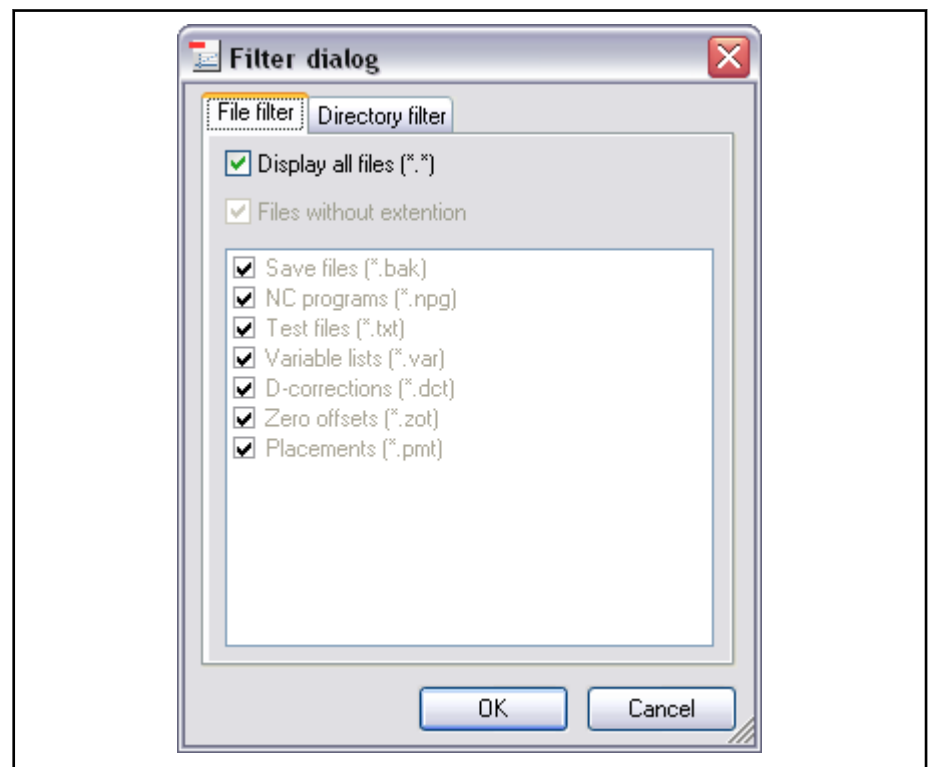


Fig. 7-165: Filter dialog for files

Control commissioning

Only files which correspond to the file types selected in the dialog are shown. All other files are not displayed. In the example above, only NC programs are displayed.

Adding file filters The list in the filter dialog for files can be extended by the "Add filter" function. This function is called via the context menu (right mouse button) of the list.

Deleting file filters Entries from the list can be deleted in the file filter dialog. Note that only the elements added by the user can be deleted. The extension specified by the installation cannot be deleted.

This function is called via the context menu (right mouse button) of the file filter list.

Setting editor Use this function to enable a user-defined editor for a certain file extension to be used when opening a file with this extension. By default, an internal ASCII editor is used unless a special editor was assigned.

This function is called via the context menu (right mouse button) of the file filter list.



This function is only available in IndraWorks Engineering.

Setting directory filter Use the filter dialog to exclude certain default directories of the control from the display in the tree. Thus, the selection of the directories can be limited.

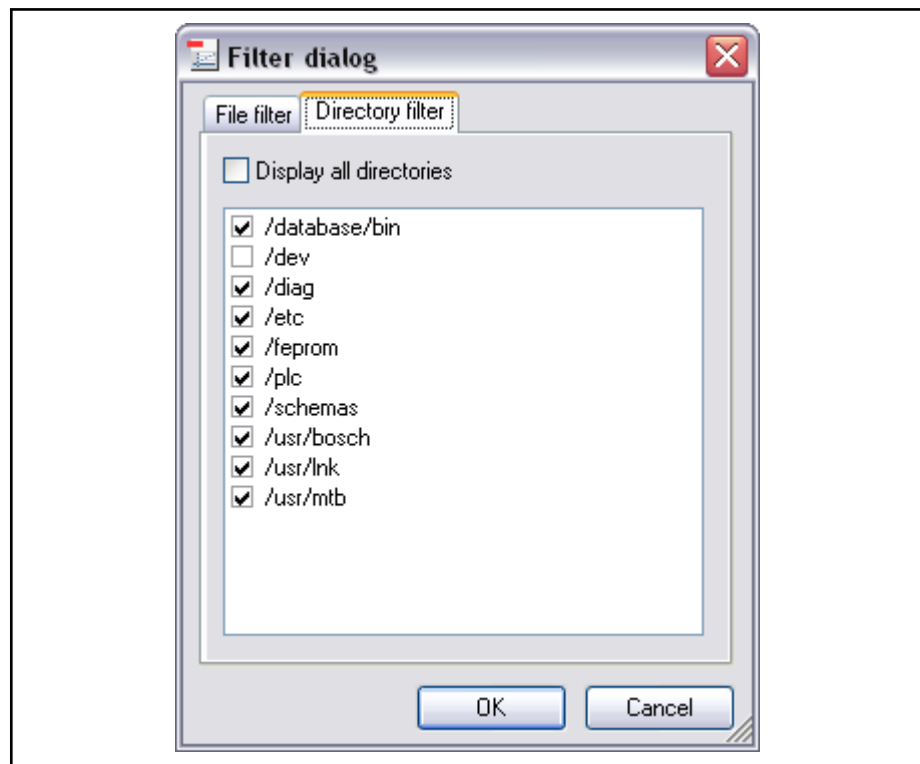


Fig. 7-166: Filter dialog for directories

All connected directories and the directories selected in the directory dialog (including mount directories) are always shown.

Adding directory filter The list of the directories in the standard can be extended using the function "Add directory filter". In this case, it is possible to include subdirectories in the list as well. Furthermore, individual directories can be deleted from the list via the context menu.

Properties of the directory The "Properties" dialog shows the following directory information:

- Directory name, can be changed

- Location of the directory
- File system in which the directory is located
- Assigned memory of the file system
- Free memory of the file system
- Memory capacity of the file system
- Creation date of the directory
- Directory attributes

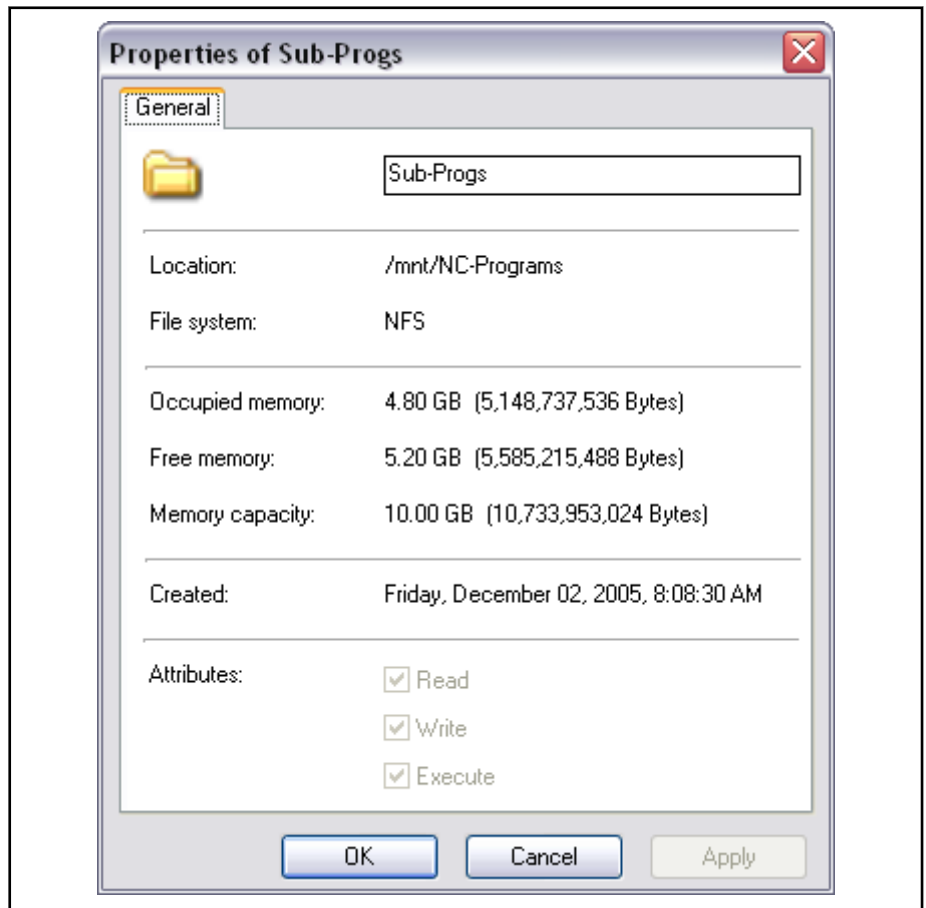


Fig. 7-167: Properties dialog of the directory



The attributes of the directory cannot be changed on the mount directory!

Properties of a file

The Properties dialog shows the following file information:

- File name, can be changed
- Location of the file
- File system in which the file is located
- File size
- Creation date, change date and date of last access
- File attributes

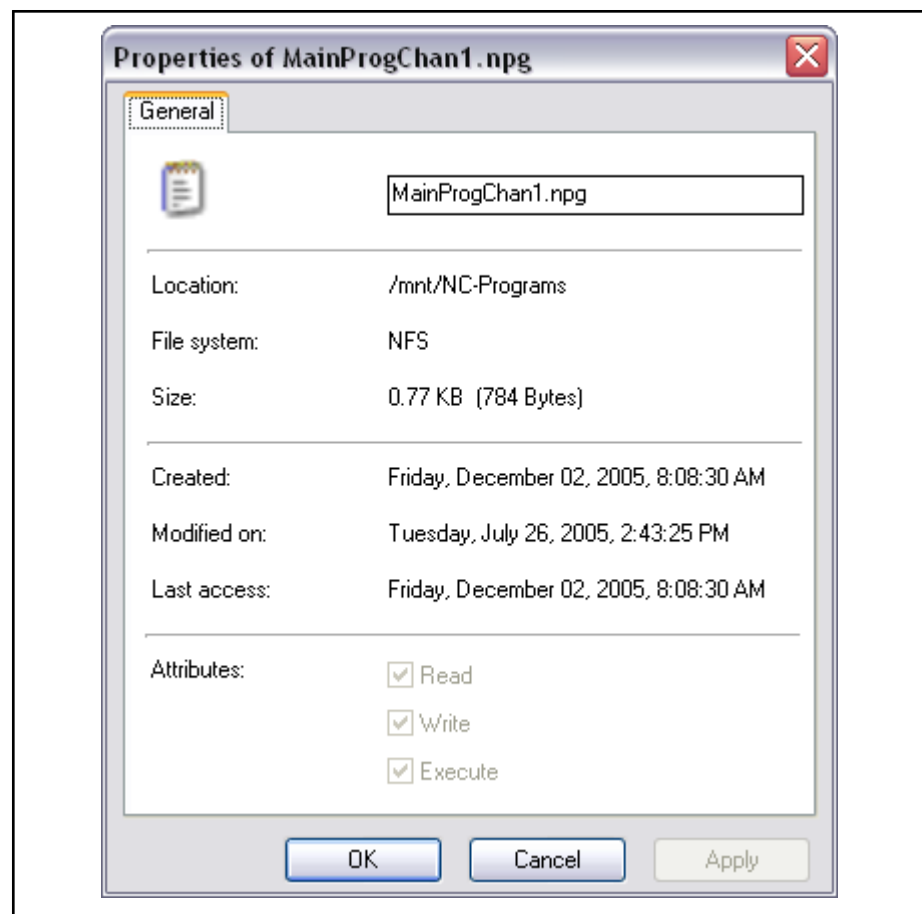


Fig. 7-168: "Properties" dialog of the file



The attributes of a file cannot be changed on the mount directory!

Importing directories/files

The user can use the "Import" dialog to import directories and/or files from the Windows file system into the file system of the NC kernel. The import function is only activated if the focus is on a directory (also the "root" directory). It is not significant whether it is in the tree or in the list. If several objects are selected in the list, the import function is deactivated. If the focus is on the "empty" list, the import function is activated.

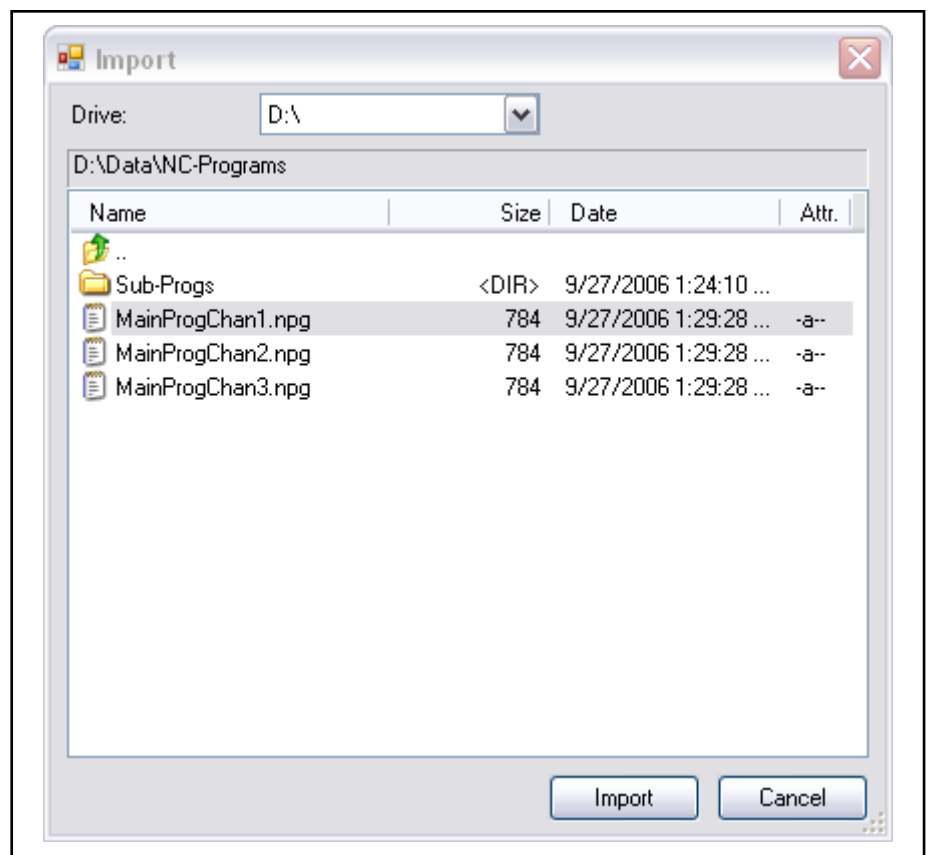


Fig. 7-169: "Import" dialog

- Upon the first call after the MTX Navigator start, "C:" is set as default drive. The last selected directory is placed as default directory for every following call (as long as the user interface was not closed)
- Several files and/or directories can be highlighted and imported. After pressing the "Import" button, all selected objects (files/directories and subdirectories and files) are copied to the selected directory in the MTX Navigator.
- If a file with the same name already exists in the target directory, the user is asked whether he would like to overwrite or cancel the copying process for this file.
- The dialog can be operated with the keyboard (i.e. also without a mouse)!

Exporting directories/files

With the export function, directories and/or files are copied from the file system of the NC kernel to a Windows directory selected by the user.

The export function is active under the following conditions:

- Not the node "root" is selected in the tree, but another directory
- In the list, one or several file(s) / directory/is/are selected



Fig. 7-170: "Export" dialog

In this dialog, the user selects the target directory in which the selected files/directories are copied from the MTX Navigator.

If the element (file/directory) to be exported already exists in the target directory, it is overwritten without prompt.

Furthermore, the dialog provides the possibility to create a new subfolder which can immediately be selected as target directory.

Program selection functions

Link NC program	If this function is selected in the function key bar, the selected NC program is automatically linked in case of a program selection.
Show tree	This function is used to switch the view of the MTX Navigator between Tree +List.
Cancel selection	This function closes the program selection without an NC program being selected.
Confirm selection	This function executes a program selection with the selected NC program. The selected NC program is visualized in the NC block display (operating screen machine).



All selected NC programs are marked with an individual symbol in the list of the MTX Navigator.

- Selected NC programs in the current channel contain a "green" symbol. 
- Selected NC programs in non-selected channels contain a "gray" symbol. 

Directory functions

These are functions which can only be used for directories (folders). These functions can only be called if a folder object is focused in the tree or in the list.



The directory functions are only active in the context "Program" or in IndraWorks Engineering.

New -> Directory

Use this function to create a new subdirectory in the currently selected directory. When this function is activated, a dialog is opened. Enter the name of the new subdirectory into this dialog. The maximum length of a directory name within the directory system of the NC kernel is limited to 30 characters. On the mount directory, the maximum length for a directory corresponds to the Windows conventions.



Fig. 7-171: Dialog: Create new directory

- Close the dialog by clicking on **OK** and create the directory.
- "Cancel" closes the dialog and does not create a new directory.
- If no name is entered and **OK** is pressed, an error message is output.
- If the name already exists in the directory, an error message is shown.
- If the directory name has more than 30 characters and the file is in the directory system of the NC kernel, an error message is shown.

Deleting

The current directory and all contained subdirectories and files are deleted. The deletion has to be confirmed by the user:

Yes The directory is deleted

No Deletion is canceled

It is possible to highlight several directories and/or files in the list (multiple selection) and delete them afterwards.

During deletion, a status box is shown visualizing the progress at deletion.



The root directory cannot be deleted!

Renaming

Use this function to rename the currently selected directory. When this function is activated, a dialog is opened. The user has to enter the name of the new directory into this dialog. The default assignment in the input field is the current directory name. The maximum length of a name in the directory system of the NC kernel is limited to 30 characters. On the mount directory, the maximum length for a directory corresponds to the Windows conventions.



Fig. 7-172: Dialog: Rename directory

- **OK** closes the dialog and renames the directory.

- Click on **Cancel** to close the dialog. The directory is not renamed.
- If no name is entered and **OK** is pressed, an error message is output.
- If the name entered already exists in the directory, an error message is shown.
- If the directory name has more than 30 characters and this directory is in the directory system of the NC kernel, an error message is shown.



The root directory cannot be renamed!

In case of multiple selection, directories cannot be renamed.

Copying

Use this function to save the currently selected directory in the clipboard. This function is required for the "Insert (Directory)" function. Only if an element of the "Directory" type is in the clipboard, the function "Insert (Directory)" is activated and can be selected by the user.

With each new copying, the current content of the clipboard is overwritten!



- The "root" directory cannot be copied!
- The return label cannot be copied!
- A multiple selection of directories in the list is possible.

Inserting

This function requires an element of the type "Directory" in the clipboard. Only in this case, this function is active and can be executed by the user.

When the function is selected, the program determines the source directory from the clipboard and copies the latter with all its subdirectories and files to the currently selected target directory.

If a subdirectory with the same name of the directory to be copied already exists in the target directory, the user has to decide whether the current directory is to be overwritten or whether the insertion is to be canceled using a dialog.

When inserting a directory with more than 30 characters from the Windows file system into the file system of the NC kernel, an error message is output.

Cutting

Basic function like "Copy"!

In addition, the source directory is deleted after insertion into another directory via the function "Insert (Directory)".

File functions

These are functions which can only be used for files. These functions are only activated if one or several files are selected in the list. For multiple selection, some functions are not activated!



File functions are only activated in the context "Program" or in IndraWorks Engineering.

New

This allows to create a new file of a certain type.

The following types are provided:

- NC programs
- NC variable list
- D-corrections
- Zero offsets
- Placements
- Text file

After selecting one of these types, the corresponding editor is started automatically. It is used to create and edit the file.

Deleting This function deletes the currently selected file. The deletion has to be confirmed by the user:

Yes The file is deleted.

No Deletion is canceled.



When deleting, a multiple selection is possible. In this case, all selected files are deleted!

Selected, i.e. active NC programs cannot be deleted.

Editing

When operating this function, the data type of the selected file calls the corresponding editor and loads the selected file.

If the data type of the selected files is unknown, the NC program editor is started in IndraWorks Operation and a simple ASCII editor is started in IndraWorks Engineering. This editor is part of the MTX Navigator.



In case of multiple selection, files cannot be edited.

Renaming

This function renames the currently selected file. When this function is activated, a dialog is opened. The user has to enter the new file name into this dialog. The default assignment in the input field is the current file name. The maximum length of a name is limited to 30 characters. On the mount directory, the maximum length for a file corresponds to the Windows conventions.



Fig. 7-173: Dialog "Rename file"

- Close the dialog by clicking on **OK** and rename the file.
- Click on **Cancel** to close the dialog. The file is not renamed.
- If no name is entered and **OK** is pressed, an error message is output.
- If the name entered already exists in the directory, an error message is shown.
- If the file name has more than 30 characters and this file is in the directory system of the NC kernel, an error message is shown.



- In case of multiple file selection, the "Rename" function is disabled.
- Selected - that means active - NC programs may not be renamed.

Copying

This function is used to save the currently selected file in the clipboard. This function is required for the function "Insert (File)". Only if an element of type "File" is in the clipboard, the function "Insert (File)" is activated and can be selected by the user.

With each new copying, the current content of the clipboard is overwritten!



- A multiple selection of files in the list is possible.
- A mixed multiple selection of directories and files in the list is also possible.

Inserting This function requires an element of the type "File" in the clipboard. Only in this case, this function is active and can be executed by the user.

When the function is selected, the program determines the source file from the clipboard and copies the latter to the currently selected target directory.

If a file with the same name of the file to be copied already exists in the target directory, the file name is prefixed by the text "Copy of" if source and target directory are identical. If both directories are different, the user is asked whether the file is to be overwritten in the target directory.

Cutting Basic function like "Copy"!

In addition, the source file is deleted after being inserted into another directory using the function "Insert (File)"!

Search functions

Find The "Search" function can be used to search for files in the file system of the NC kernel and in all mounted directories of the Windows file system or on a Compact Flash (CF) card of a controller-based control.

This function is only activated if the focus is on a directory (folder) of the tree (also the "root" directory) or the list. If several objects are selected in the list, the search function is deactivated.



The search function can be used in IndraWorks Engineering and in both IndraWorks Operation contexts.

If the "Search" function is called in a directory (folder), the following dialog is displayed:



Fig. 7-174: "Search File" dialog

To search for certain files, the following settings can be made:

- Entry of a file name combined with a placeholder "*"
 - *.* - Search for all files (default setting)

- **<FileName>.<ext>** - Search for a certain file
- ***.<ext>** - Searches for all files with a certain extension
- **<Filename>.*** - Searches for files with a certain name and any extension
- **<Filename>*.*** - Searches for files starting with a certain character and with any extension
- **<Filename>*.<ext>** - Searches for files starting with a certain character and with a certain extension
- **Entering a path**
In the "Find in" field, the path of the focused directory is applied when opening the dialog. This search path can be manually adjusted by the user. It has to be observed that a valid (existing) path is entered. If the path is invalid, searching is aborted with an error.
- **Search Subfolder**
This checkbox can be used to limit the file search to the specified directory or to include all subdirectories (default setting).
- **Include mount directories**
Select this checkbox only if the selected directory is root ("/"), since only "root" can be provided with one or several mount directories and subdirectories.

If the search is started, all files with the set specifications are searched. While searching, a progress bar is displayed. At the end of the search, the new node "Search Results" is newly created and automatically selected in the tree. All found files are shown in the list with their entire path. These are "links" to the original files.

Functions of the "Search Results" node

Generally, the search result remains until a new search is started or explicitly completed by the user. The advantage is that it can be switched any number of times between the individual directories of the NC file system and the search result without losing the search result as it is common for other browsers. The disadvantage is that updates such as modifications or deletion of original files made after the search cannot be seen in the search result list.

Repeat search This function starts the previous search again without displaying the search dialog. This function is especially useful if the original files were modified after the search but the search result should be up-to-date.

Complete search This function deletes the search result list (list of links to the files found) and the "Search Results" node.

Functions of the elements (links) in the search result list

Go to This function switches the directory to the original file and focuses it. The "Search Results" node remains.

Available file functions *The number of available file functions on the elements of the search result list is limited to the following functionalities:*

- **Delete**, to delete the original file as well as the link from the search list
- **Export**, to export the original file
- **Properties**, to visualize the properties of the original file.

All other file functions are not activated for elements of the search result list.

7.11.2 NC configuration (machine parameters)

The NC is configured via machine parameters.

Double-click on the "NC configuration" node to start a tool structurally representing the number of parameters and displaying a corresponding online help for each highlighted parameter.



The configuration of the machine parameters is described in detail in the "Machine Parameters" manual.

7.11.3 Mount directories

General information

The MTX control operates with a control-internal file system. The control can access files and directories outside the control via mount directories.

The following targets can be mounted:

- Directories of the flash card of a control
- USB stick in a PC
- Directories on the hard disk of a PC
- Network drive of a PC

The "mount directories" are used for the backup as well as for the general archiving and data exchange. The NC user interface can directly access up to ten mount directories.

Configuring mount directories

It is mounted via the "Network File Service" (NFS). A "Bosch Rexroth NFS client" runs in the MTX control.

To access data of a PC (USB stick, directories, network drives), the "Bosch Rexroth NFS server" has to run on this PC. This server is installed when installing IndraWorks on the PC.

To access directories of the flash card in the control, a "Bosch Rexroth NFS server" is additionally running on the control.

*The **mount configuration of the NCS client** is executed in IndraWorks Engineering:*

1. Going online with the control
2. Open the context menu of the control node and select the menu item ("Configure Device" \ "Configure Mount Directories...")
3. The mount configuration is executed in the following dialog. Use the three keys on the top right to create a new configuration and to edit or delete a configuration selected in the list.

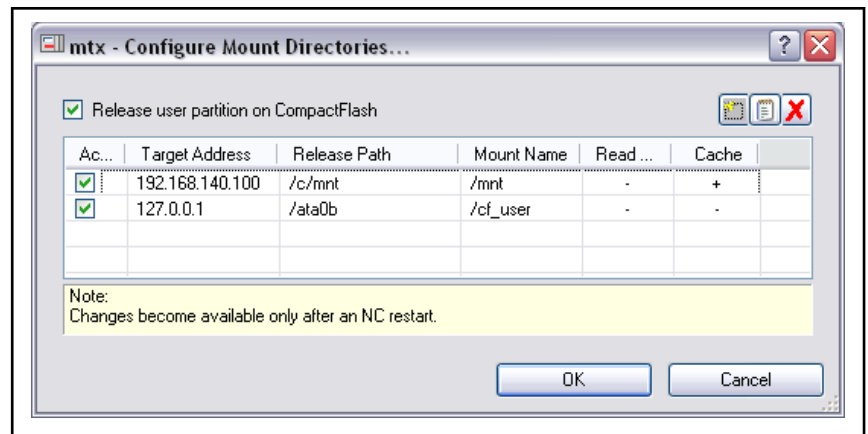


Fig. 7-175: Mount configuration dialog

Column meaning:

- Active: An existing mount configuration can be enabled via the checkbox in this column.
 - Target address: IP address of the device on which the directory to be mounted is located [the "Bosch Rexroth NFS server" has to run on this device (PC)].
 - Release path: Path to the directory to be mounted
 - Mount name: Name of the mounted directories as it is to be displayed in the file system of the control
 - Read-only: A plus sign (+) indicates that the access authorization of the control to the mounted directory is read-only. A minus sign (-) indicates that the access authorization of the control to the mounted directory is read and write.
 - Cache: Directory to be cached (only cache directory possible)
4. Confirm the dialog with **OK** and the configuration to the "Bosch Rexroth NFS Client" is transmitted to the control. To apply the configuration, restart the control.
 5. Press **OK** to transfer the configuration to the "Bosch Rexroth NFS Server" as well. This is only possible if the server is running on the same PC than IndraWorks Engineering. If this is not the case, configure the NFS server manually on the PC on which the directories to be mounted are located.

Open the "Export.us" file on the PC for editing purposes (storage "Export.us": C:\Documents and Settings\Rexroth\IndraWorks\MTX or C:\Documents and Settings\Rexroth\IndraWorks\MTX).

The "Export.us" file contains the directory to the path to be mounted as well as the IP addresses of the controls which are permitted to access this directory. The "localhost" entry indicates the access from the MTX Emulation.

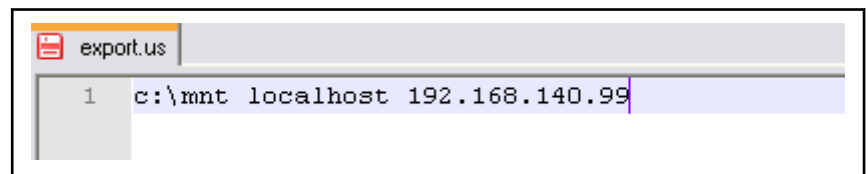


Fig. 7-176: NFS server, example configuration Export.us

After saving the file, the new configuration has to be declared to the "Bosch Rexroth NFS server". Go to the task bar for the corresponding

symbol. Right-click on this symbol to open the context menu. Execute "Reload Export.us" to declare the new configuration to the NFS server.



Fig. 7-177: "Bosch Rexroth NFS server" in the task bar



Fig. 7-178: Context menu of the "Bosch Rexroth NFS server" in the task bar

This completes the configuration of the mount directories.



The control tries to mount the configured directories at startup. If the corresponding "Bosch Rexroth NFS server" or the directors does not exist, the control startup can take longer.

7.12 Sercos

7.12.1 Axis commissioning

Commissioning tools

NC Configurator - Machine parameters

General information

The Engineering Desktop is the main tool for modifying and configuring data and settings of the MTX control system. When the Engineering Desktop was started using the "IndraWorks Engineering" icon, the following screen is displayed.

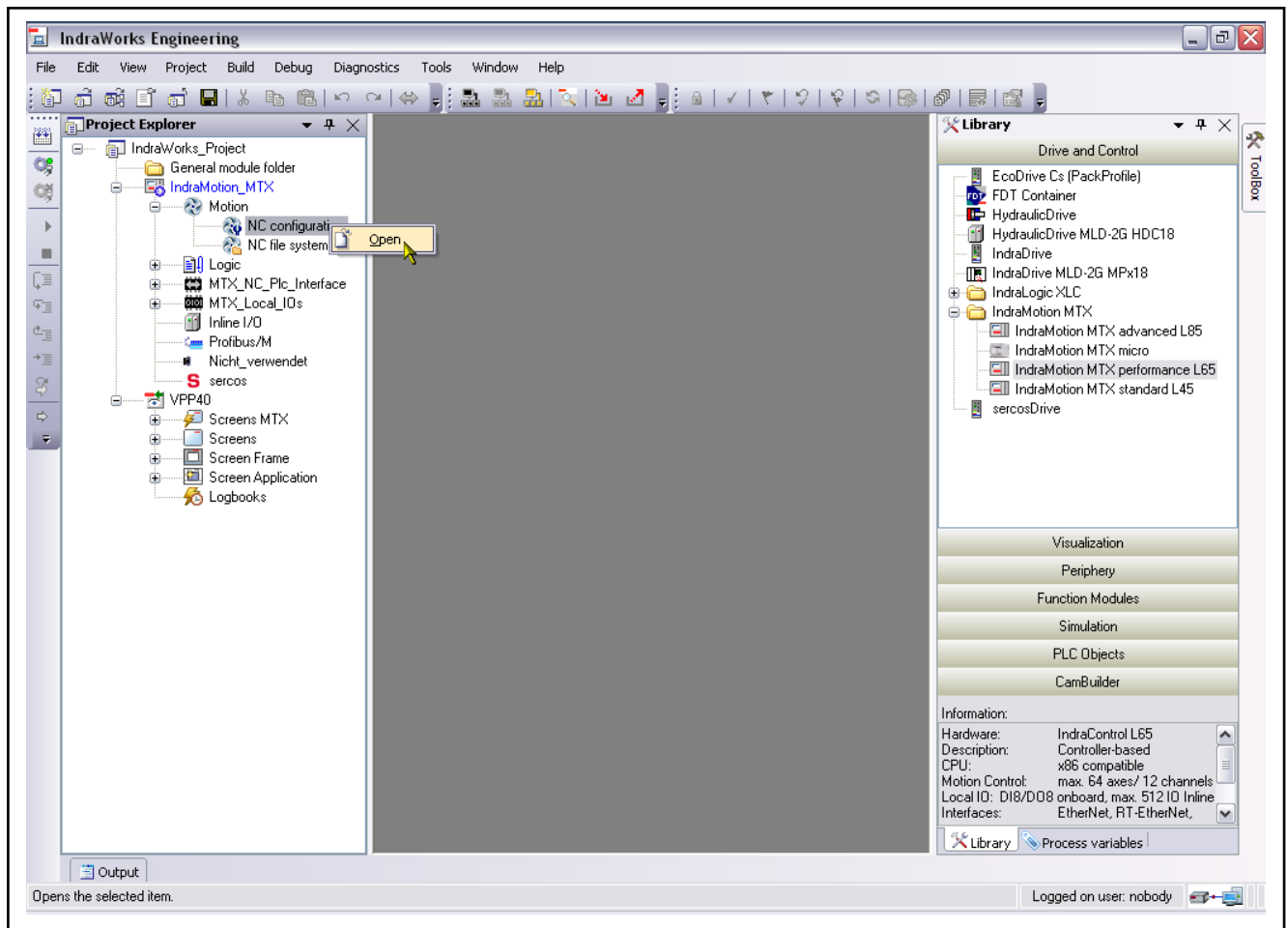


Fig. 7-179: Engineering desktop

The Project Explorer (left column) is used to navigate through the individual topic areas.

The NC configuration is used to commission the NC axes. The configurator is started via the context menu of the "NC Configuration" node.

Control commissioning

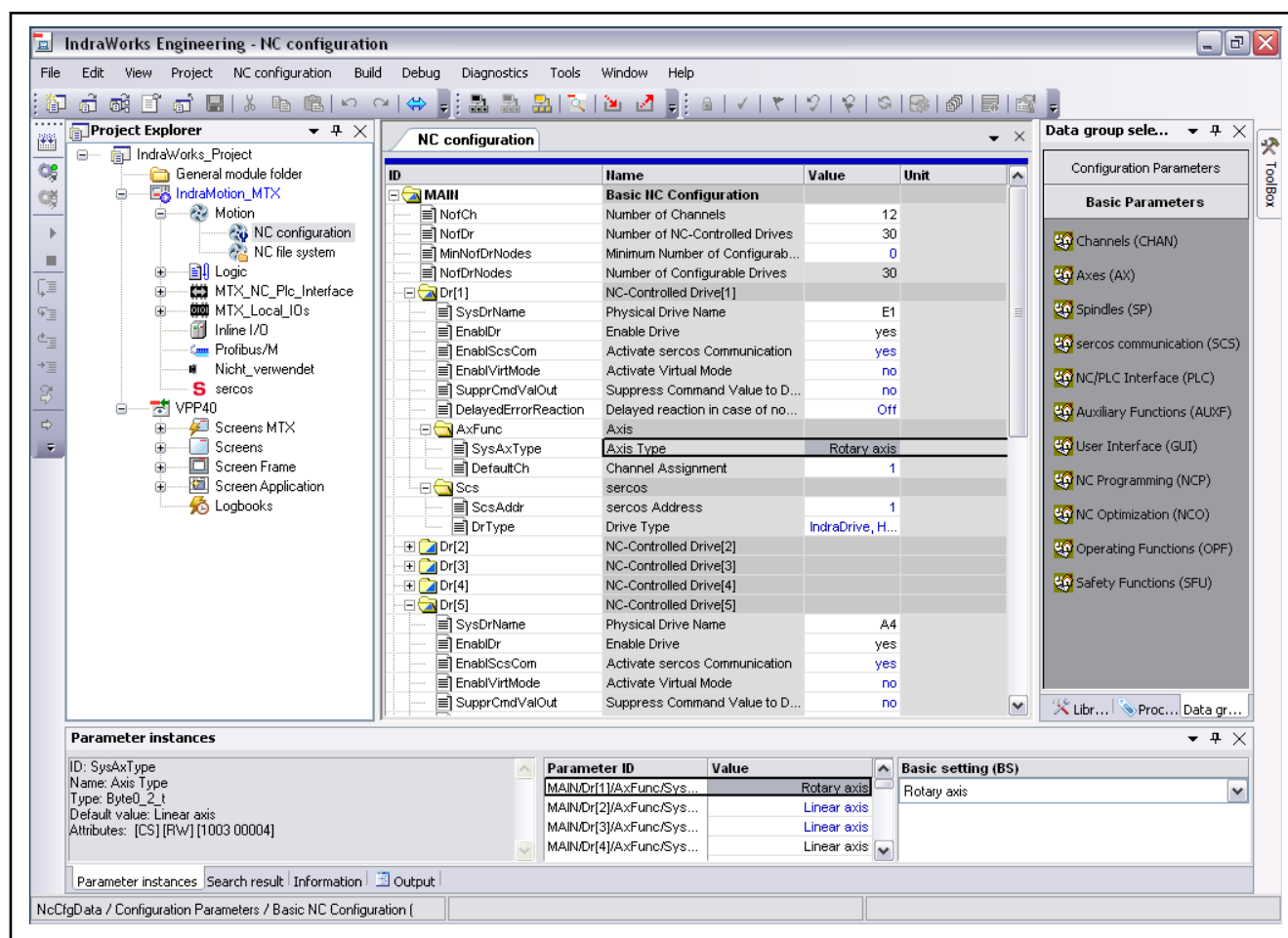


Fig. 7-180: "NC configuration" node

The desktop is divided into several sections:

- Project Explorer (left column)
- "Edit parameter" window (centered)
- Data group selection (right column)

Selecting parameters

The section to be edited can be selected in the "Data group selection" column under "Basic Parameters".

Important to commission the drives are:

- Main configuration (MAIN)
- Axes (AX)
- Spindles (SP)
- Sercos communication (SCS)

The optional parameters can be selected in the section "Setup Configuration Data". These parameter ranges cover specific topics, such as "electronic couplings", "measuring" and "traversing to fixed stop". They are only displayed in "Optional Parameters" after their selection.

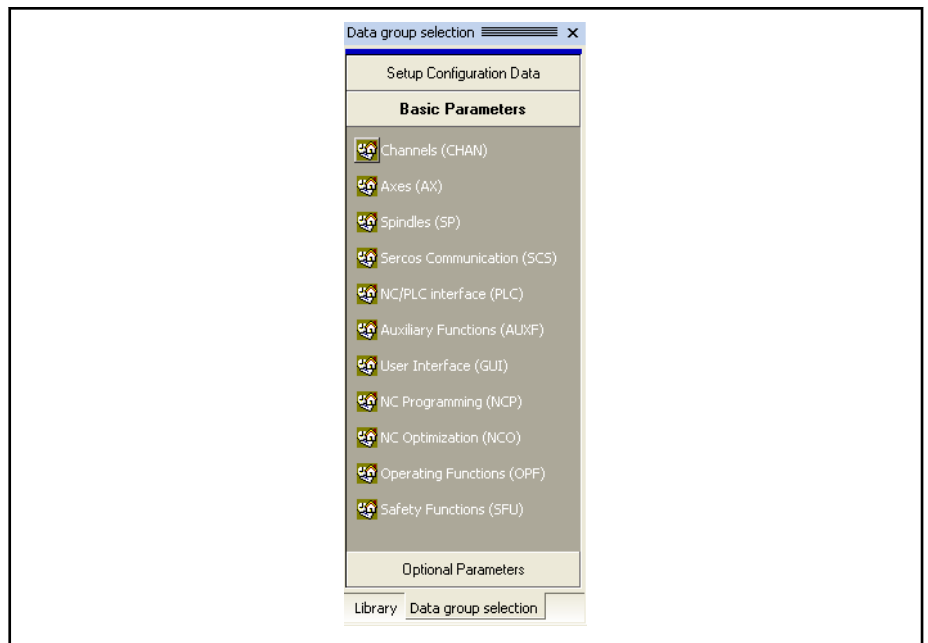


Fig. 7-181: Basic parameters



The "Configuration" tool is described in the manual "MTX Machine Parameters". Only the important steps to commission the drives are described in the following section.

Finding parameters

The configuration parameters relevant to commission the drives are described in a later chapter.

To find the listed parameters more quickly, a search function is available in the main menu under **Edit/Find**. This search function can also be started using the shortcut <Ctrl> + <F3>.

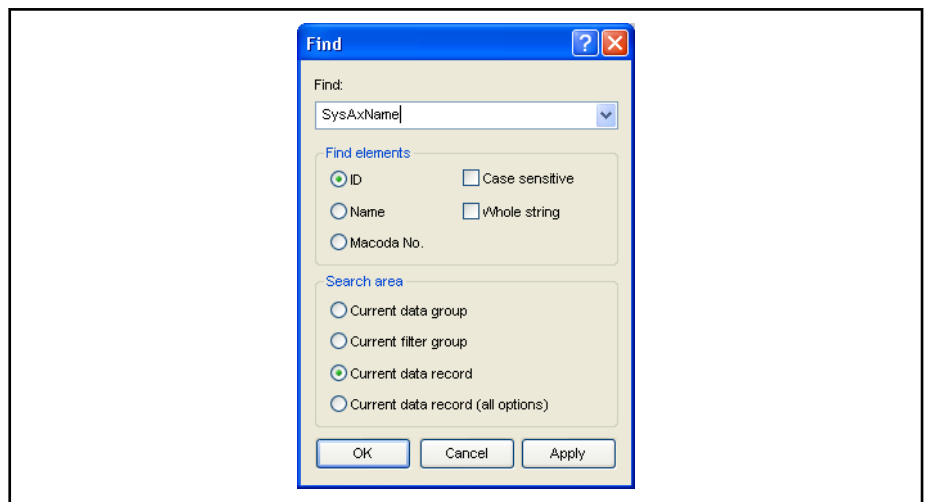


Fig. 7-182: "Find" menu

The search for "SysAxName" is exemplarily shown. The search results are displayed in Engineering Desktop under the tab **Search Results** in the lower part of the screen. Double-click on a search result to jump to the corresponding parameter.

NC Editor - SCS files

The SCS files are important when commissioning the drives. They are used for the Sercos initialization of the connected drives while the control is starting up.



The handling of the SCS files and the configuration syntax are described in the chapter [SCS handling](#), on page 381 of this documentation.

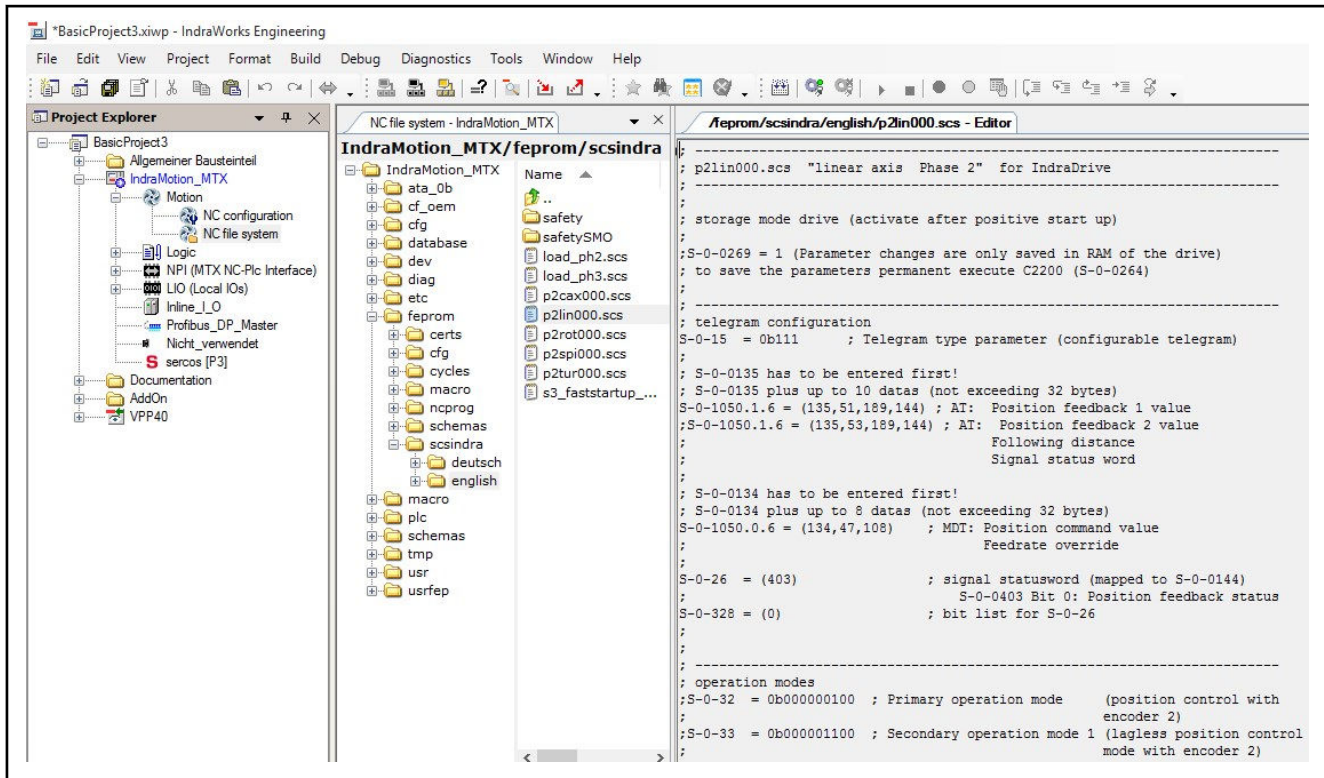


Fig. 7-183: Editor for Sercos files

The "IndraMotion MTX" control system can already provide SCS example files located in the control directory `"/root/feprom/scsindra/"` ("root" stands for the control name). Copy the examples to the directory `"/root/"` and adjust them to its application.

Double-click on the SCS files to open and edit them.

Sercos master

General information

After opening the Engineering Desktop, it cannot be communicated via the non-cyclic service channel of the Sercos ring. To be able to edit drive data, the control has to go "online" first. For this function, go to the icon in the toolbar or click in the main menu under "Project - Online". If the Sercos ring is at least in Phase 2, all the associated drives are also created in the project. However, communication is not yet active.

The communication with the drives is already activated when the drive is selected. Alternatively, it is possible to use the context menu of the Sercos node to go online simultaneously with several drives.

The following functions are provided at the Sercos node.

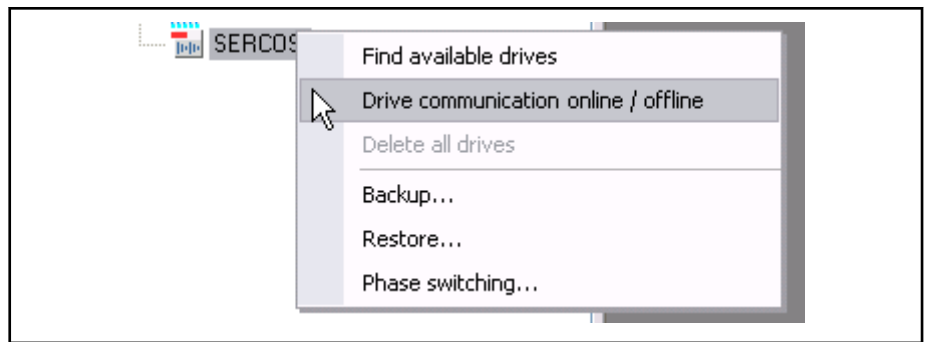


Fig. 7-184: Sercos context menu

- **Find available drives:**
All drives configured in the NC data are created. If required, a switch to Sercos phase 2 is triggered upon request. This function is already triggered when the control goes online.
- **Drive communication online/offline:**
This function establishes and cancels the communication with the drives. In a dialog, the drives can be selected for going online. This function implicitly includes searching for drives if not all configured drives are in the project.
- **Archive:**
A dialog opens in which the drives are selected for backup. Furthermore, it can be determined whether all parameters or only the changed parameters are to be backed up. The selected drives go online accordingly.
- **Restore:**
In a dialog, the drives, for which a previously created archive is to be written back again, are selected. The selected drives go online accordingly.
- **Phase switching:**
A dialog to switch the Sercos phase opens.
- **Sercos device address assignment:**
It is possible to set the Sercos address of the bus device. The devices are identified via their position in the ring
[chapter "Sercos device address assignment" on page 364](#)
- **Firmware management**
The firmware of several Sercos devices can be replaced without data loss.
[chapter "Firmware download on Sercos devices" on page 368](#)

Behavior when going online

The behavior of the drives when the control goes online can be set in the "Properties" of the Sercos node.

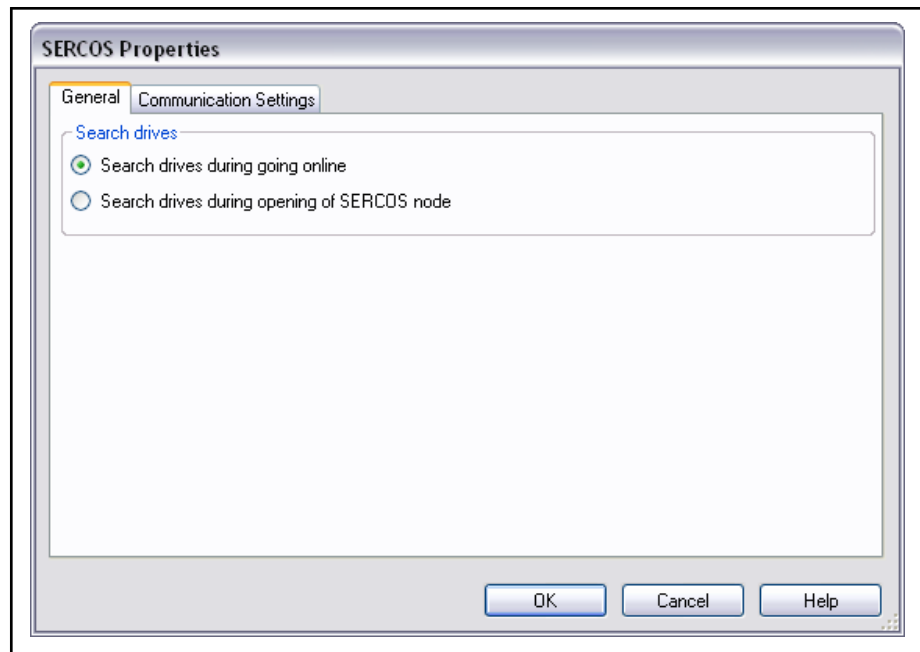


Fig. 7-185: Find Sercos settings for the drive

Before a communication can be established with the drives, the basic configuration of the drives in the project has to be compared and, if necessary, updated with the drives available at the control. This process might take several seconds depending on the number of devices at the Sercos ring. Thus, it can be selected when the adjustment is to be made.

- Search drive during going online (default)
When the control goes online, it is searched for drives. It can be immediately used for operation.
- Search drives during opening of Sercos node
When the control goes online, the Sercos node collapses. The drive configuration is not checked. To access the drive parameterization, the Sercos node is expanded. Only at this point in time, a check is made.
This setting can be very reasonable especially if there is a high number of drives in the ring and if the drive parameterization is rarely required.

Sercos device address assignment

Therefore, the control has to be online.

This function compares the configuration of the devices in the project with the devices actually found in the control.

The data from the project is displayed on the left.

The data of the real bus devices is displayed on the right. It is shown in the sequence of the devices in the ring.

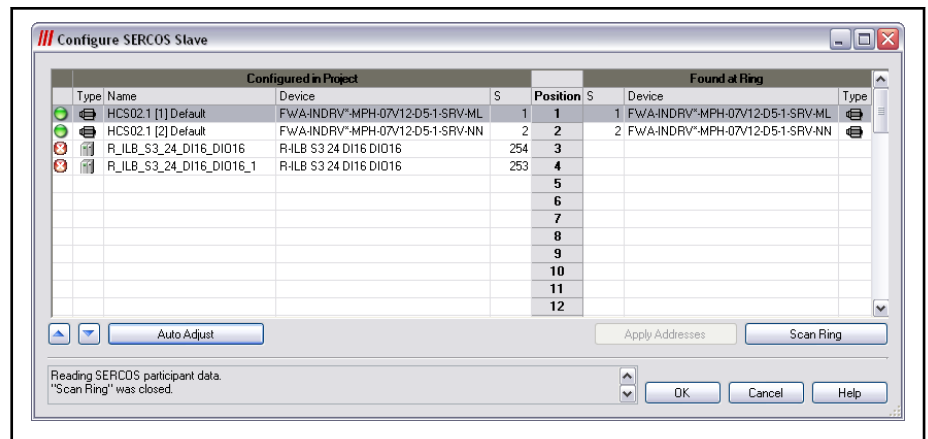


Fig. 7-186: Dialog to configure Sercos devices

Column description

- **Project - Type:**
Type of device: Drive or I/O
I/Os can be activated/deactivated by clicking on the icon. Clicking does not work for drives.
- **Project - Name:**
Name of the device in the project.
- **Project - Device:**
Name of device: Drive firmware or I/O type
- **Project - S (Sercos address):**
Configured Sercos address of the device.
This address can be changed for I/Os. The configured Sercos address of a drive can only be changed in the NC configurator. Changed addresses are displayed in "**bold**".
- **Control - Type:**
Type of device: Drive or I/O
- **Control - Device:**
Name of device: Drive firmware or I/O type
- **Control - S (Sercos address):**
Set Sercos address of device.
A new Sercos address is set for the device. Changed addresses are displayed in "**bold**".

Describing operating elements:

- **Project - Arrow up;**
Moves the selected device one up in the project.
- **Project - Arrow down;**
Moves the selected device one down in the project.
- **Project - AUTO:**
Sorts the configured devices to achieve a high number of matching cases.
- **Control - Scan ring**
The data of the devices at the ring are read in again.
- **Control - Apply addresses**

The changed addresses are written to the devices and a new Sercos startup is executed.

Status information

The status of the respective place is given as colored icon in the first column. More information on the respective status is given when keeping the mouse pointer on the icon for a moment (tooltip).

There are the following status messages:

- **Gray**
The ring has not yet been scanned. The status is unknown.
- **Green**
The configuration of the device in the ring does not match the device.
- **Yellow**
 - The device name does not match.
There is another I/O device at the ring as expected in the project.
 - Firmware does not match.
The firmware version of the drive in the ring is different than expected in the project.
 - The Sercos addresses do not match.
The device on the control matches with the configured device, but not with the address. The Sercos address can be adjusted in the project or at the device.
- **Red**
 - The Sercos address is used multiple times.
A Sercos address exists multiple times. The Sercos addresses have to be corrected until they are unique.
 - No device was found at this position.
A configured device was not found at the ring. An I/O can be deactivated in the project
 - No device is configured at this position.
An unconfigured device was found at the ring. Either the device has to be removed from the ring or added to the project.
 - The configured device type does not fit to the hardware found.
A drive is configured in the project, but an I/O is available in the ring or vice versa.

Using the Sercos UC channel

General information Drive can directly be activated with Sercos via their IP address. This allows to directly execute a firmware or PLC download for example via the Sercos interface. Furthermore, tasks such as parameterization backups are normally faster.

Activating IP channel To use IP communication, certain settings have to be made. These can be found in the properties of the "Sercos" node. The provided options differ whether being in online or offline mode.

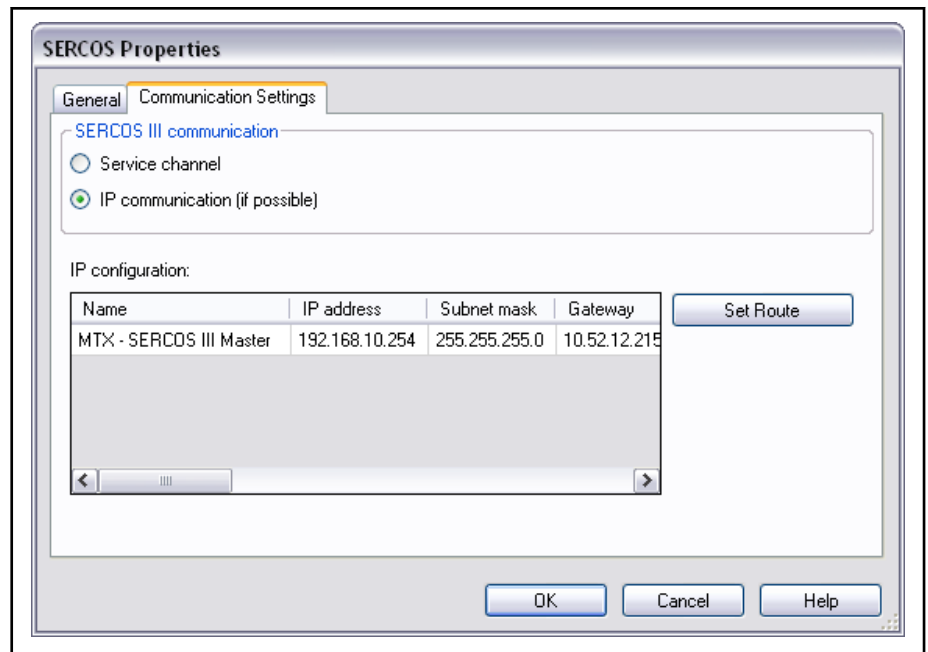


Fig. 7-187: Activating IP communication

It can be selected whether the IP channel is to be used. If no IP connection is established to the drive, the service channel is automatically used. If problems occur with this setting, it can also be specified to always use the service channel. This information is saved in the project.

To use the IP channel, the Engineering PC must be familiar with the subnet of the drives. If this is not the case, use the **Set Route** button. Select this function to enable the route to the drives.

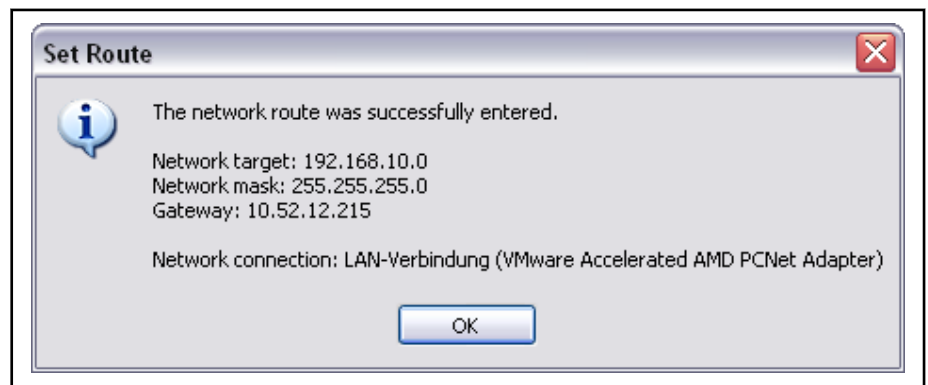


Fig. 7-188: "Set Route" was carried out

The IP channel can now be used. When going online next time with the drive, it communicates via IP channel.



- Administrator rights are required to set the route.
- A route set this way is deleted upon the next PC restart and has to be created again.



- Another option to configure the route to the drives is provided by the command "route" in the Microsoft prompt. It can be also specified via parameters that the route remains after a PC restart. For more information on this command, refer to the Windows help. The function used in the dialog corresponds to this call:

```
route add 192.168.0.0 10.52.12.215 -mask
255.255.255.0
```

- For more options, contact your network administrator.
The subnet to be routed in is determined by the address of the Sercos master. ("[IP configuration of the master](#)" on [page 368](#)). The gateway is the control.

IP configuration of the master

Open "Advanced" at the Sercos master to select different settings.

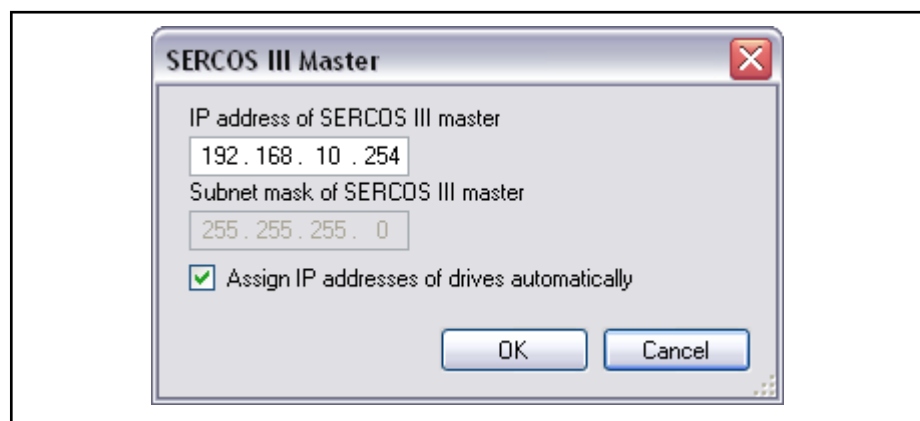


Fig. 7-189: Advanced setting of the Sercos master

- **IP address:**
IP address of the Sercos master interface (upon delivery: 192.168.143.254)
- **Subnet:**
The subnet cannot be modified and is set to the value 255.255.255.0
- **IP addresses of the drives are automatically assigned:**
If this option is activated (delivery state), a permitted IP address is assigned to each drive during Sercos startup. This IP address consists of the subnet and the Sercos address of the drive (e.g.: IP address of the master: 192.168.143.254, Sercos address of the drive: 4, IP address of the drive: 192.168.143.4).
If this option is not enabled, the address has to be specified once correctly for each drive.

Firmware download on Sercos devices

Target The firmware of multiple Sercos devices is to be exchanged without loss of data.

- Prerequisites**
- It is one of the following devices:
 - Drives supporting the firmware download via TFTP (firmware $\geq$ MPx16)
 - Drives used as PLC device and supporting the firmware download via TFTP
 - Sercos III gateway for SafeLogic compact (SLC-3-GS3S00300)

- Supplier (PowerSupply HMV0x, KMV0x)
- From the control firmware version 14V14: All Sercos devices reporting the class "SCP_TFTP = 16#0E01" in the parameter S-0-1000 "SCP type and version". The class "SCP_NRTPC=16#1801" has to be reported additionally to specify a matching IP address
- Only for control firmware versions < 14V14: The control and the computer - on which IndraWorks is installed - are located in the same subnet
- The control is switched online
- The devices are connected to the control and enabled

Procedure

Read the release notes for the respective firmware first.

In this example, the control runs in **BB RUN** and the axes are active.

It is recommended to follow this sequence:

1. Stop all active axes and ensure that no axes start during the firmware download.
2. Go to the context menu of the "Sercos" node to open the firmware management. The following window lists all available Sercos devices.

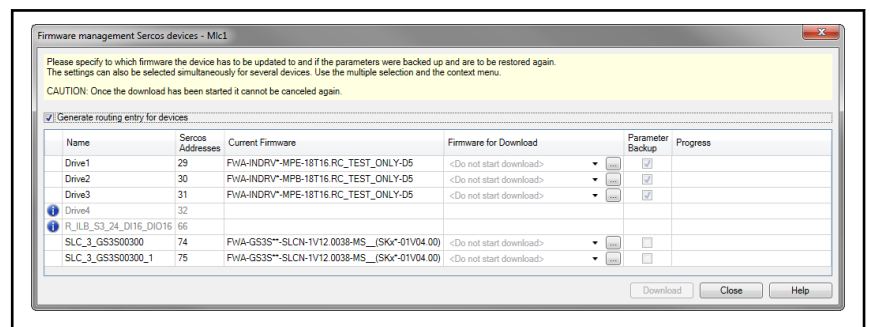


Fig. 7-190: "Firmware Management Sercos Devices" dialog

3. Only for control firmware versions < 14V14: Select whether to generate a routing entry for the devices. A routine entry is required to access the devices from the PC. If a routine entry has already been created, this options does not have to be selected.
4. Select the devices for the download. Select a firmware for the download. To select multiple devices at a time, click on these devices with the <Ctrl> key pressed and select a firmware for the download via the context menu.

Then, specify the firmware to be downloaded. Either select the firmware for each device from the drop-down list or press ... to find another firmware file. To use the same firmware for multiple devices, click on the devices with the <Ctrl> button pressed and select an already used or a new firmware via the context menu.

At the end, specify the devices for which the parameters are backed up before the download and restored after the download. The parameter backup can only be selected for devices supported for the parameter backup.

Devices, which cannot be selected for the download, are shown in grey and have an information icon with a tooltip stating why the download is not possible for this device.

5. Click on **Download** to start the process. You are then prompted whether to stop the PLC. After confirming, the download starts.

NOTICE

Once started, the process cannot be canceled.

During this process, do not switch off the control voltage of the devices or cancel their connection. This can cause irreparable damage to the devices.

6. After transferring the firmware to the devices, it is automatically tried to restart them. If this is not possible, as the device is for example not supported, restart the devices manually. A message is displayed listing the devices to be restarted. Start the devices by switching off and on the control voltage and acknowledge the message with **OK**.

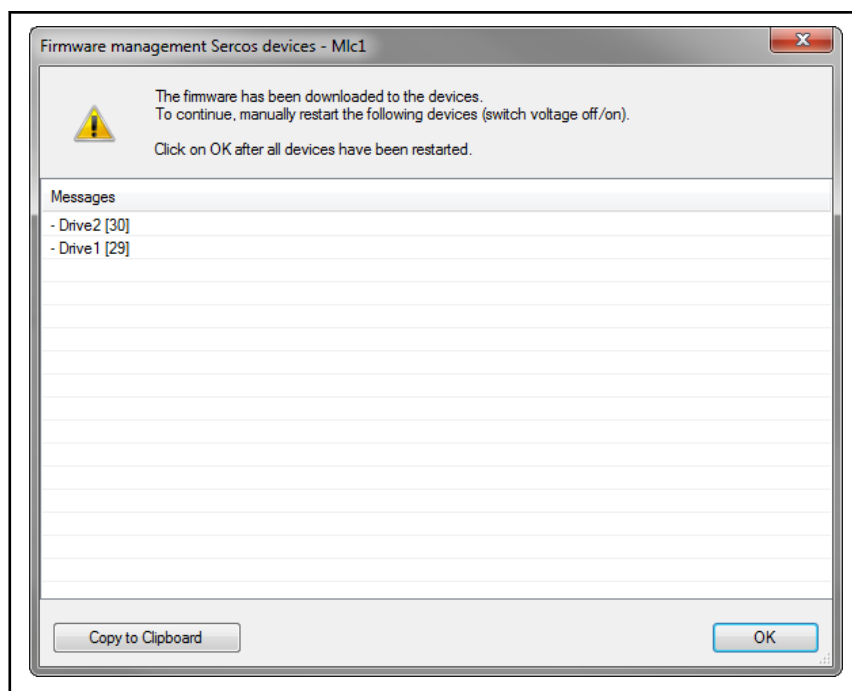


Fig. 7-191: Message to restart the Sercos devices

7. Subsequently, the function packages of the drives are restored. If might be required to restart some devices. If this restart cannot be executed automatically, a message is displayed, stating which drives have to be started again manually. Start the corresponding drives by switching off and on the control voltage and acknowledge the message with **OK**.
8. The previously archived parameters are restored. If not all parameters can be restored, a list of the parameters, which could not be restored, is displayed.



In case of another firmware, drive parameters might have been changed or even omitted.

The result of the download is displayed at the end. If errors or warnings occurred, the already archived parameters can be saved to a file. This file can be manually imported again via the parameter import (context menu of the control -> Parameters -> Import...).

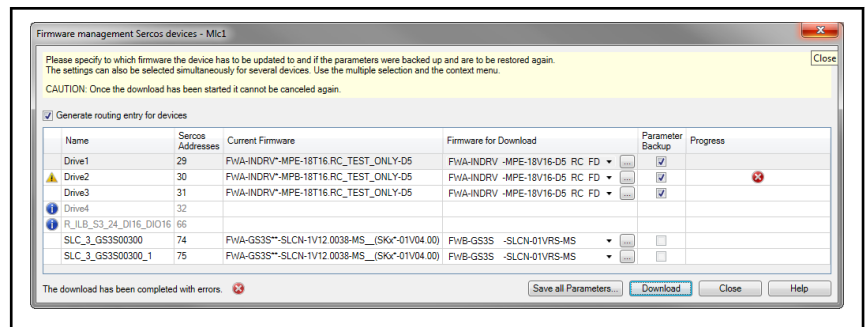


Fig. 7-192: "Firmware Management Sercos Devices" dialog – Result

Drive commissioning

Overall procedure

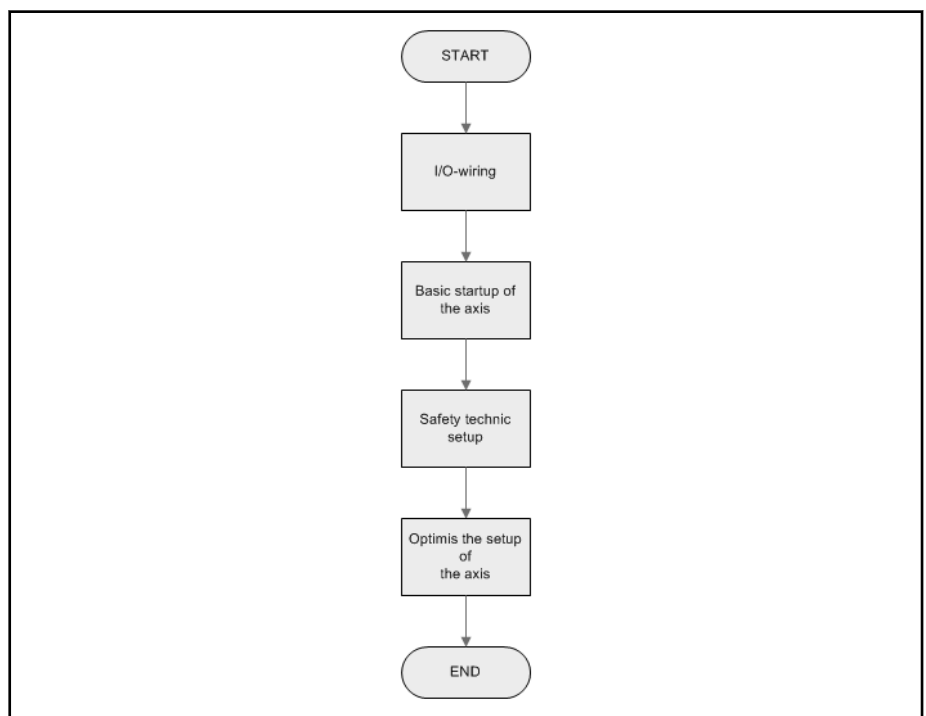


Fig. 7-193: Drive commissioning procedure

I/O wiring

General information

⚠ CAUTION

Danger of personnel injuries and machine damage due to incorrect wiring!

Wiring only according to the guidelines of Bosch Rexroth!

Handling instruction: I/O wiring

General wiring of the drives to prepare for commissioning.

Drives: Wiring the inputs and outputs, power wiring

Wire the drives according to the guidelines of Bosch Rexroth.



Follow the instructions of the IndraDrive documentation to ensure proper use of the drives.

Handling instruction: Activating the E-Stop Function

Before the axis can be operated with the "IndraMotion MTX" NC control, it must be ensured that the safety equipment is functioning properly.

System/machine: Check E-stop circuit

1. Check E-stop circuit
2. Check the safety engineering of the drives and set the parameters accordingly if required.



Observe the safety instructions on electrical drives.

		Documentation
Documentation:	Rexroth IndraDrive drive system	Control circuits for power connection

Handling instruction: General drive commissioning

Before commissioning, operate the drives of the system/machine together with the control but without the NC, if possible.

This comprises the following steps:

IW Drive: Drive commissioning

- Mechanical installation of drive amplifiers
- Electrical wiring of motor, drive amplifier and control
- Commissioning of the overall electrical system and the switch cabinet
- Basic commissioning of drives without NC
- Basic settings of axis with IW Drive



Refer to the drive documentation for information about commissioning of the drives.

The drives are optimized together with the control at a later point in time (after the control commissioning).

		Documentation
Documentation:	Rexroth IndraDrive drive system	

Drive-side parameterization

General information

To parameterize the drives, enable the communication. This is done by means of the menu point "Drive communication online/offline". There, the necessary drives are selected and switched online. Depending on the number of the drives selected, it might take a while.

During initial activation, the drives existing in the system are created in the project.

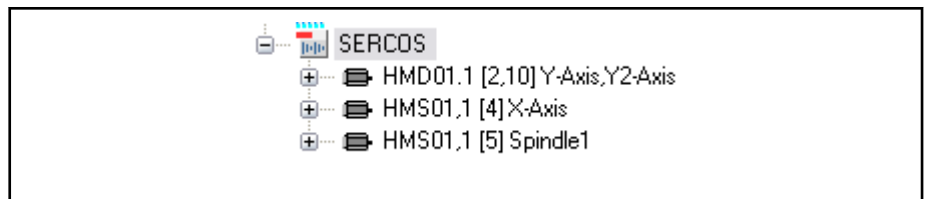


Fig. 7-194: Project tree with drives

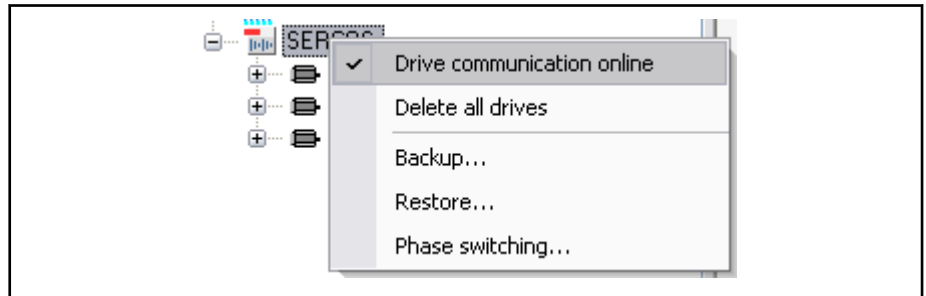


Fig. 7-195: Context menu "Drive communication online"

The current phase is highlighted in green in the upper part of the dialog. By clicking on a phase (0 to 4), switching to the relevant Sercos phase is possible. If one of the drives is in parameterization mode ("PM mode"), it is displayed in line "Sercos phase", after phase 4.

Select the phase using the buttons "Parameterize" and "Operation". An ongoing phase switching is indicated with a progress bar.

Furthermore, the functions to be executed during phase switching can be selected:

- Delete error of state class 1
- Transfer SCS files to Sercos phase 2
- Transfer SCS files to Sercos phase 3

A status message is displayed at the end of switching or if errors occur.

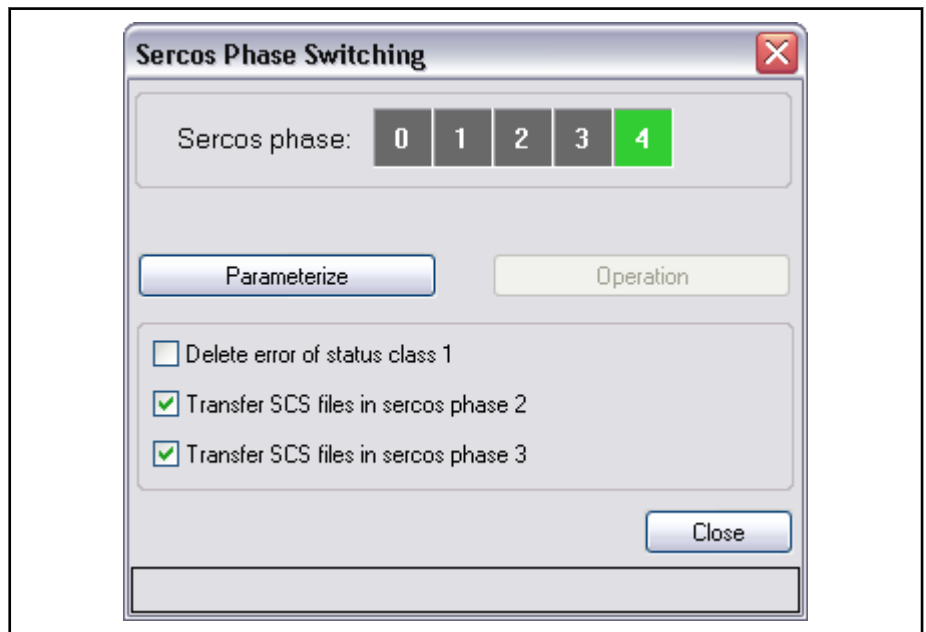


Fig. 7-196: Phase switching dialog



In case of IndraDrive drives, all relevant parameters can be transferred to phase P2. The parameter transfer in phase P3 is not applicable.

Handling instruction: Parameterizing drives in IW Drive

IW Engineering/Project tree: Extend IW data structure

1. Switch the control "Online" in order to be able to create drives

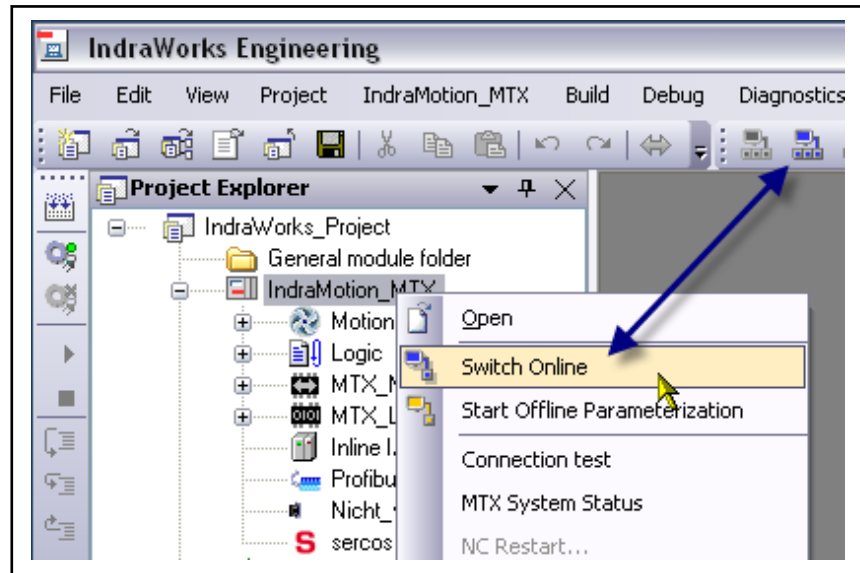


Fig. 7-197: Going online with the control

2. Creating drives automatically
3. The function "Drive communication online/offline" can be started by clicking the right mouse button on the "Sercos" node.

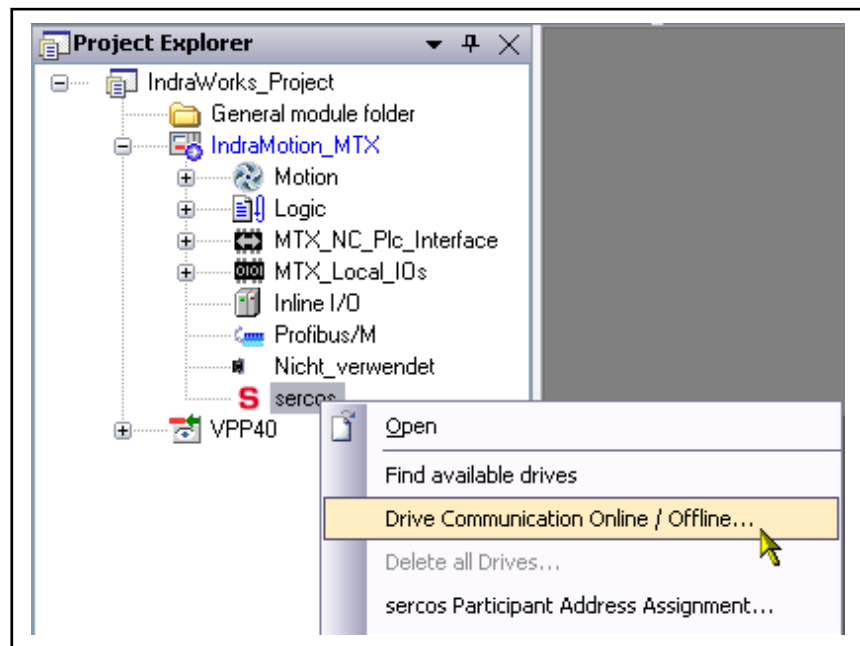


Fig. 7-198: Creating drives

To completely create the tree structure of the drives, you can first switch all the drives "Online". Only the drives that you want to edit need to be selected.

This function may take several minutes. The duration depends on the number of selected axes.

The successful creation of all the axes is then displayed as follows.

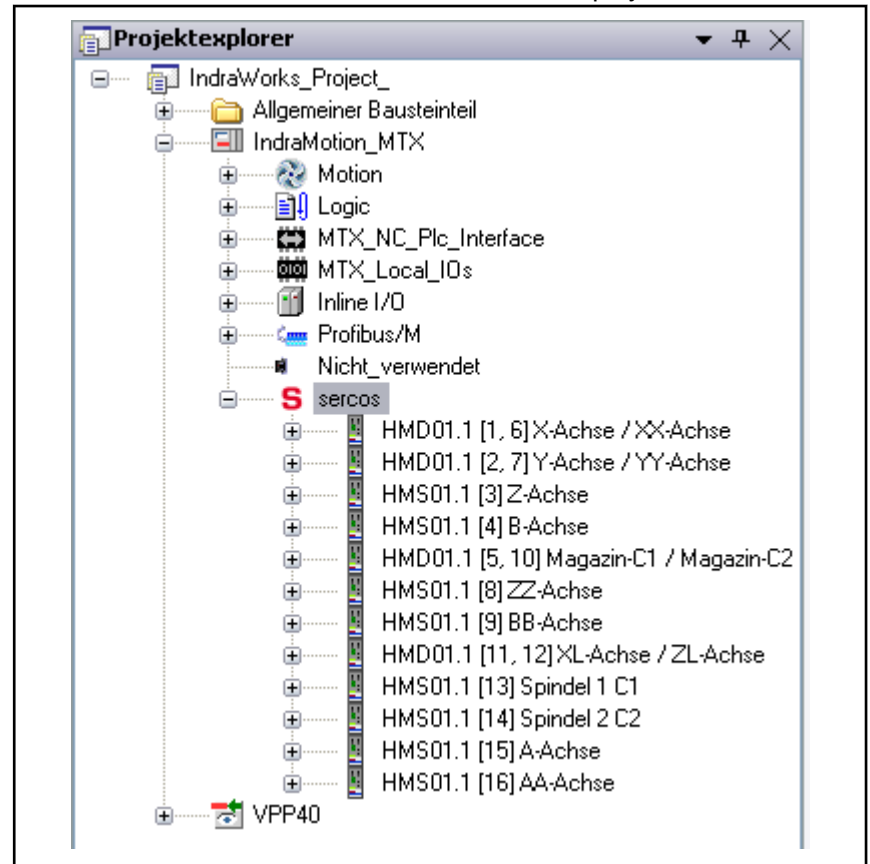


Fig. 7-199: Displaying drives



The data structure in Engineering Desktop must be supplemented so that you can access the parameters of the drive. This is required when changing or archiving the drive parameters.

All the drives that have been created in the Configurator should also be created in the IndraWorks data structure.

Documentation:	IndraWorks Commissioning	Axis commissioning

General axis commissioning

Handling instruction: Traversing the axis with control

After the axis has been configured, traverse it with the control. Before the axis is moved in Automatic mode, various basic settings should be checked.

IW-Operation/operating mode "Jogging": Manual traversing of the axis

1. Check the traveling direction and the display
2. Homing
3. Set the zero point of the axis
4. Determine and set the zero point on the machine (in the case of absolute encoders)



The override potentiometer should be set to a low value before starting the axis so that dangerous situations cannot occur.

IW-Operation/operating mode "Automatic": Move axis with test program

1. Create a test program for the axis (if desired, adapt/use the delivered test program)
2. Start the test program and test the axis
3. Circularity test



Before the drive is operated in "Automatic" mode, "MDI" mode can be used to selectively start an NC block.

IW Engineering/Sercos: Optimizing the drives

Optimize the drive with IW-Drive



A final optimization of the drives should be carried out by trained personnel.

Control commissioning

Overall procedure

In general, the steps shown in the following diagram are required to add NC axes to an existing control configuration.

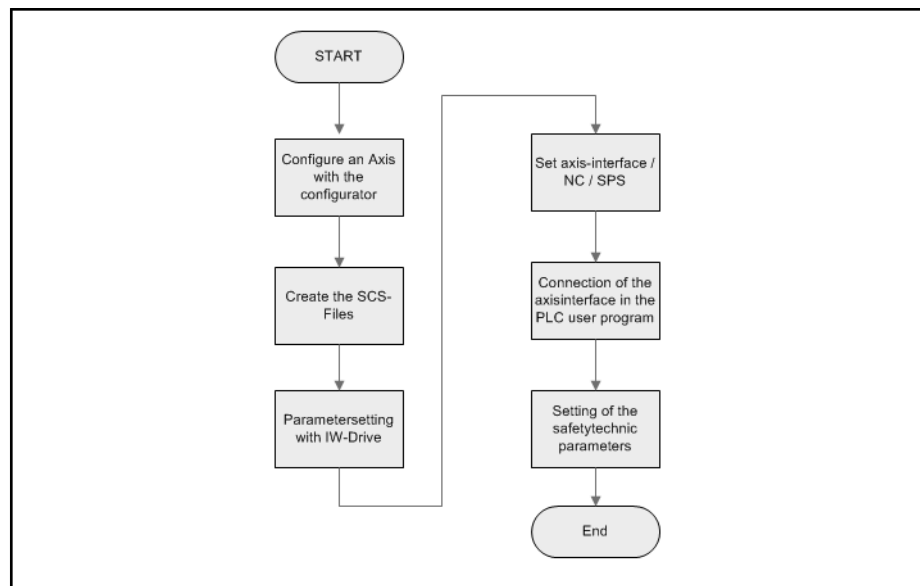


Fig. 7-200: Axis commissioning procedure

Control-side machine parameters

General information

Channels/axes must be created for a new project and a project extension. When the channel/axis structure is created, the parameters of the channels/axes are specified and default values are set. An axis or a channel can only be displayed or configured if a parameter set exists. Proceed as follows to configure channels/axes:

1. Open the editor for parameter configuration

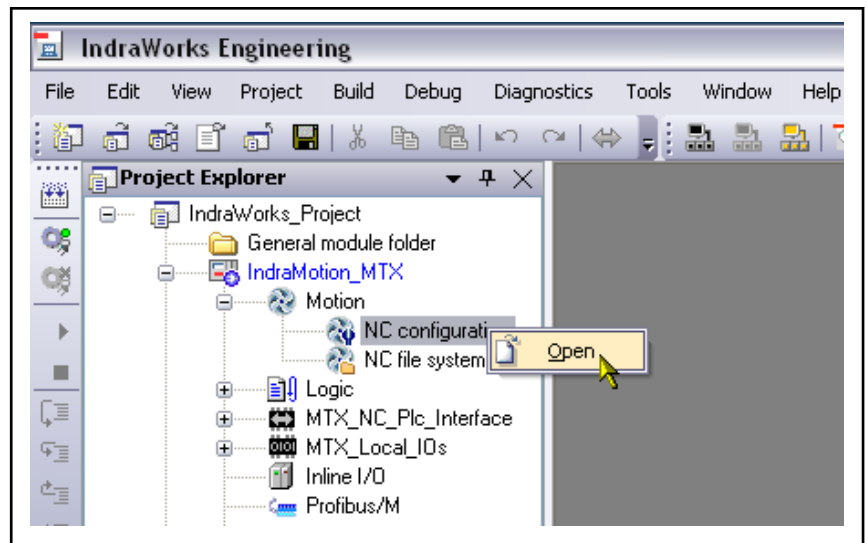


Fig. 7-201: Opening the configurator

2. Create channel/axis structure

No channels exist in a newly created IndraWorks project. Configure the channel before configuring the axis parameters.

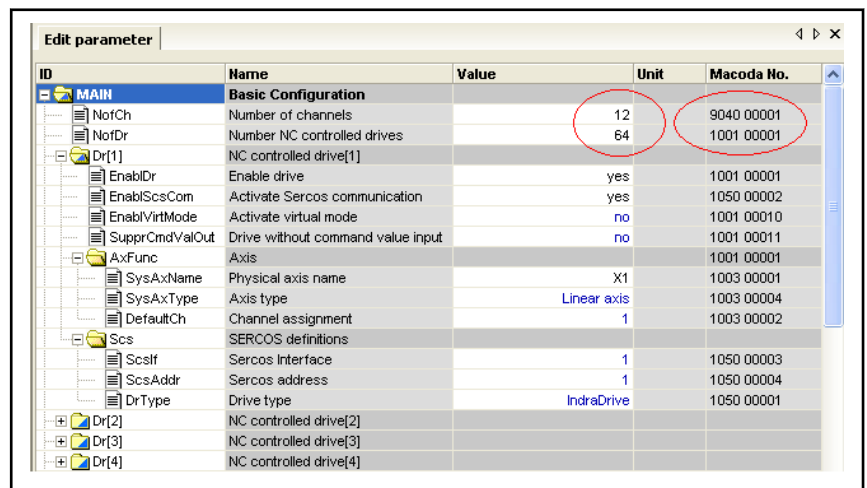


Fig. 7-202: Setting the number of channels and axes

- To set the number of channels/axes, modify the entry accordingly (see [fig. 7-202 "Setting the number of channels and axes" on page 377](#)). The channels and axes are automatically created by the system.
- Add channel parameter structure
- Press the right mouse button on the "CHAN" node and a menu opens. In this menu, a channel can be created/added under "New".

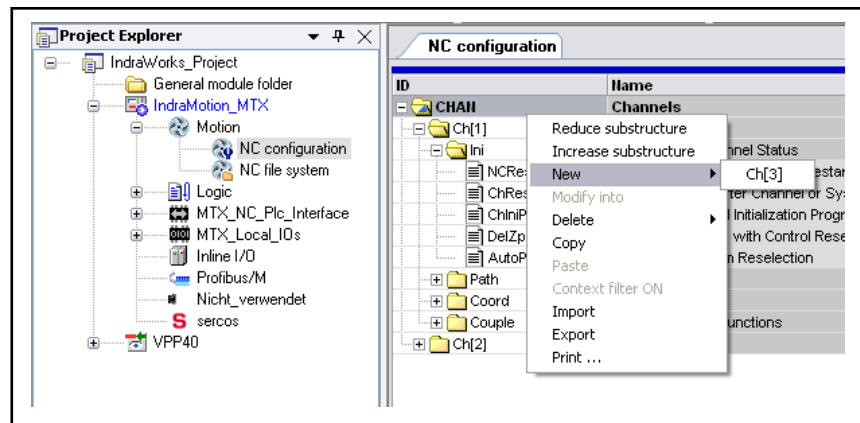


Fig. 7-203: Creating a channel structure

6. Add axis parameter structure

If channels exist, axes can be created below the "MAIN" node. This is accomplished by clicking on the "MAIN" node with the right mouse button. During the axis selection, ensure that the correct axis type is configured. The following axis types are available in the menu:

- AxFunc[i] for linear and rotary axes (i = <system drive no.>)
- SpFunc[i] for spindle (i = <system drive no.>)
- AxFunc/SpFunc[i] for spindles/C-axes (i = <system drive no.>)

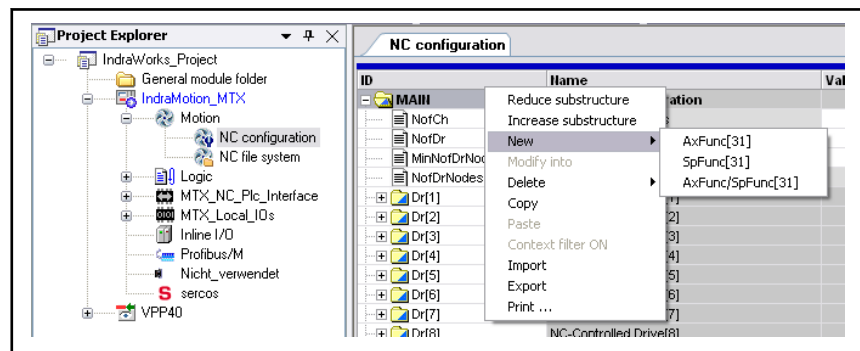


Fig. 7-204: Creating an axis parameter structure

7. Change in parameters are indicated via a change of the symbol at the "NC Configuration" node. Depending on the parameter to be changed, either a reset or a control restart is to be executed before they are applied. The tooltip at the "NC Configuration" node indicates the necessary. When completing the NC configurator, it is possible to execute the corresponding function immediately via a dialog.

Handling instruction: Creating a channel/axis

Add a drive to the created/restored project. The general operation of IndraWorks is described in the documentation on "MTX IndraWorks".

IW Engineering/configuration: Create a channel/axis

1. Create channel
2. Create an axis/spindle

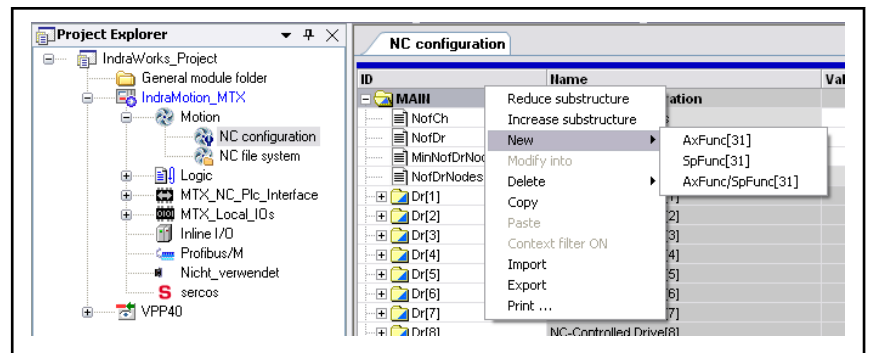


Fig. 7-205: Creating a channel/axis



An axis can only be created if a channel was created before.

		Documentation
Documentation:	IndraWorks Engineering	Working with Projects and Devices

Handling instruction: Configuring the axis parameters

The relevant axis parameters for the relevant machine data are checked or the recorded value is entered.

1. Set the NC parameters according to the machine data.
2. Check and, if necessary, modify the axis parameters and Sercos parameters.



No double addresses must be assigned to the parameter [Addr]. The addresses must be identical to the physical drive addresses.

Default settings can be retained.

3. Restart the control after the parameters have been checked.

IW-Engineering/configuration: Editing parameters

1. Open the configuration tool to edit the parameters
2. Check/edit the following parameters:
 - **SysAxType** "Axis type" (1003 00004)
 - **SysAxName** "Physical axis name" (1003 00001)
 - **TravLim(1)/TravLim(2)** "Traversing distances"
 - **MaxVel** "Maximum velocity" (1005 00002)
 - **MaxAcc** "Maximum acceleration" (1010 00001)
3. Edit the parameters for Sercos master communication
4. Edit axis/spindle parameters

Control commissioning

Edit parameter				
ID	Name	Value	Unit	Macoda No.
MAIN	Basic Configuration			
NotCh	Number of channels	Value of parameter		9040 00001
NotDr	Number NC controlled drives	64		1001 00001
Dr[1]	NC controlled drive[1]			
EnablDr	Enable drive	yes		1001 00001
EnablScsCom	Activate Sercos communication	yes		1050 00002
EnablVirtMode	Activate virtual mode	no		1001 00010
SupprCmdValOut	Drive without command value input	no		1001 00011
AxFunc	Axis			1001 00001
SysAxName	Physical axis name		X1	1003 00001
SysAxType	Axis type	Linear axis		1003 00004
DefaultCh	Channel assignment	1		1003 00002
Scs	SERCOS definitions			
ScsIf	Sercos interface	1		1050 00003
ScsAddr	Sercos address	1		1050 00004
DrType	Drive type	IndraDrive		1050 00001
Dr[2]	NC controlled drive[2]			
Dr[3]	NC controlled drive[3]			
Dr[4]	NC controlled drive[4]			

Fig. 7-206: Configurator for axis parameters (AX)

Edit parameter				
ID	Name	Value	Unit	Macoda No.
MAIN	Basic Configuration			
NotCh	Number of channels	Value of parameter		9040 00001
NotDr	Number NC controlled drives	64		1001 00001
Dr[1]	NC controlled drive[1]			
EnablDr	Enable drive	yes		1001 00001
EnablScsCom	Activate Sercos communication	yes		1050 00002
EnablVirtMode	Activate virtual mode	no		1001 00010
SupprCmdValOut	Drive without command value input	no		1001 00011
AxFunc	Axis			1001 00001
SysAxName	Physical axis name		X1	1003 00001
SysAxType	Axis type	Linear axis		1003 00004
DefaultCh	Channel assignment	1		1003 00002
Scs	SERCOS definitions			
ScsIf	Sercos interface	1		1050 00003
ScsAddr	Sercos address	1		1050 00004
DrType	Drive type	IndraDrive		1050 00001
Dr[2]	NC controlled drive[2]			
Dr[3]	NC controlled drive[3]			
Dr[4]	NC controlled drive[4]			

Fig. 7-207: Configurator for spindle parameters (SP)

 To avoid dangerous situations, check all parameters for correctness.

		Documentation
Documentation:	MTX Configuration	NC Configurator

IW Engineering/configuration: Applying parameters
Restart the NC to transfer the modified NC parameters.

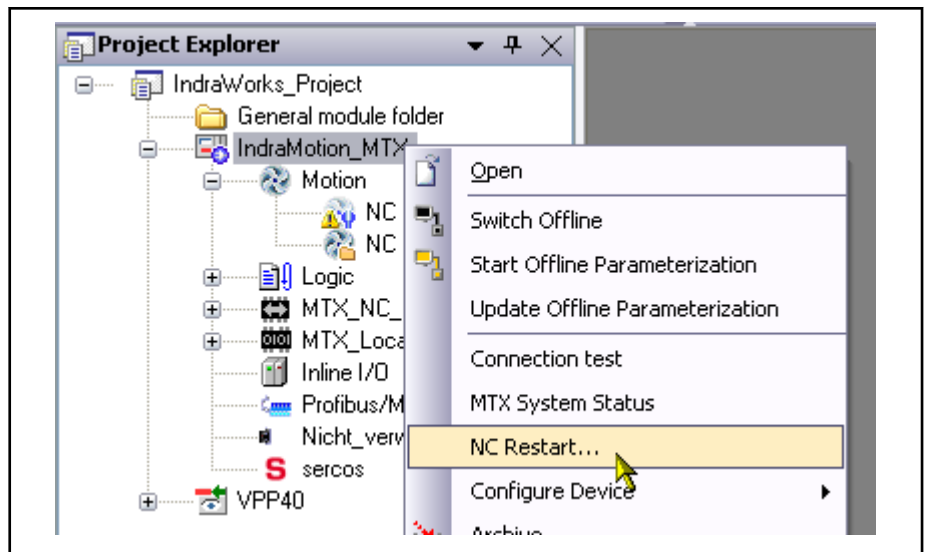


Fig. 7-208: NC restart



When using the NC emulation:
Exit EMU with a value of 14 and restart.

		Documentation
Documentation:	MTX Configuration	Editing machine parameters

SCS handling

General information

The NC provides a number of functions which significantly simplify handling with the connected Sercos drives:

- Sercos initialization for all connected Sercos drives during control start-up
- Automatic parameter download for all connected Sercos drives during control startup
- Program-controlled read-only access to the Sercos parameters of all connected drives using CPL commands SCS (..) and SCSL (..)
- Program-controlled write access to the Sercos parameters of all connected drives using "Writel" (WID)
- *Selectable debug functions can be used to log the following functions when starting up the NC:*
 - The Sercos parameters transmitted in the phases 2 and 3
 - The Sercos timing in phase 3



For detailed information on these CPL commands, refer to the "MTX Programming Manual".

Sercos initialization

The tasks of the Sercos initialization of the NC are as follows:

- Closing all Sercos rings. Thus, it is checked whether the Sercos transfer lines are OK

- Structure of the Sercos timing and the communication channels to the connected drives
- Starting up the phases of all Sercos drives up to phase 4 (cyclic operation is enabled)



The Sercos initialization can be executed either with or without automatic parameter download in the drive.

Relevant configuration parameters:

Splnd		(1001 00001)
ENAVirtMode		(1001 00010)
SysAxName	"Physical axis name"	(1003 00001)
DefaultCh	"Channel assignment"	(1003 00002)
SysAxType	"Axis type"	(1003 00004)
DrType	"Drive type"	(1050 00001)
ENAScsCom	"Enable drive in the ring"	(1050 00002)
ScsIf	"Number of the Sercos ring"	(1050 00003)
ScsAddr	"Sercos address"	(1050 00004)
KindAutoTime¹⁾	"Type of automatic transmission time calculation"	(1050 00010)
TransTime (see footer 1)	"Transmission time of the drive telegrams"	(1050 00011)
TransTimeMdt (see footer 1)	"Transmission point in time of master data telegrams"	(1050 00012)
EnableTestMode	"Enabling test mode"	(1050 00020)
ScsTimeOut	"Time monitoring for Sercos startup"	(1050 00021)
ModeScsFiles	"Download Sercos data in phase 3"	(1050 00022)
OpticTransPow (see footer 1)	"Optical transmission power stage"	(1050 00031)
ScsBaudRate (see footer 1)	"Sercos transmission rate"	(1050 00032)
ScsCycTime	"Sercos cycle time"	(9030 00001)

Configuring SCS files

Starting parameterization and initialization:

- Switch off the power supply to the drives!
- The listed parameters must be adapted to the application.

The "Automatic Calculation of Transmission Time" (1050 00010) function is used to simplify the Sercos commissioning.

If "Automatic Calculation of Transmission Time" is enabled, the following applies:

- At startup, the NC automatically calculates the Sercos transmission points in time T1 to T4 from the telegram lengths (AT, MDT) and from the drive protection times (T1min, TATMT, TMTSG, T4min)

1) This parameter is not supported anymore

Abbreviation	Designation
AT	Drive telegram
MDT	Master data telegram
TATMT	
TMTSG	

- The following do not apply anymore:

TransTime

"Transmission Point in Time of Drive Telegrams" (1050 00011)

TransTimeMdt

"Transmission Point in Time of Master Data Telegrams" (1050 00012)

- The configured Sercos addresses must be identical to the Sercos addresses set on the drives. To do this, refer to the documentation on the drives.
- Trigger control reset



When parameter downloading is active, the current drive parameters are overwritten by parameter values in the Sercos files of the NC. If the Sercos files are not adapted, unintended drive reactions or dangerous machine states can result.

Thus, disable the automatic parameter download for all drives for which there are presently no adapted Sercos files in the NC.

- After the control startup, check the Sercos phase of the drives on the drive itself. Ideally, the drives should display phase "4".

If phase "4" is not displayed, the following has to be considered:

- Drive displays phase "0":
The NC cannot close the ring (the transmitted MST does not return to the NC). Verify if all Sercos devices in the ring are active and if the Sercos connections are installed correctly.
- Drive displays phase "1":
The NC cannot find a specified drive. Check whether all Sercos addresses parameterized in the NC correspond to those of the drives.
- Drive displays phase "2":
The NC was not able to establish a communication channel or a correct timing to the drive, or the drive could not be switched to phase "3" due to a parameterization problem. Check the drive parameterization (if an automatic parameter download is activated, check the respective Sercos file for phase 2 in the NC. If applicable, activate the logging function for transmitted Sercos parameters, see [chapter "Logging transmitted Sercos parameters" on page 387](#)).
- Drive displays phase "3":
The drive could not be switched to phase 4 due to a parameterization or timing problem. Check the drive parameterization (if automatic parameter download is activated, check the respective Sercos file for phase 3 in the NC. If applicable, activate the logging function for transmitted Sercos parameters, see [chapter "Logging transmitted Sercos parameters" on page 387](#)).



If the parameters of drives are not set correctly, the power supply must remain switched off.

Creating Sercos files (SCS files)

Automatic parameter downloading is always activated (by default) in the MTX NC control. If the parameter download is active, the NC can completely parameterize connected drives at startup. Prerequisite: Correctly adjusted Sercos files (*.scs) in the NC. Sercos files are saved in ASCII format.

From version MTX 14V12, creating an scs file for the drive type "IndraDrive, HNC100-3X" (DrType, 1050 00001) is not required. The control determines all Sercos scs parameters (usually contained in the scs files) automatically, using configuration parameters and a new machine parameter group "Sercos Parameter (SCSP)" and transmits the determined Sercos parameters to the drive. Scs files can still be used. All Sercos parameter assignments in the scs files overwrite the automatically generated Sercos parameters.

The two files "load_ph2.scs" and "load_ph3.scs" are significant for the parameter download. Specify in these files which Sercos files are transmitted to which drives via the "use..." command.

The following files are used for transmission:

- File "load_ph2.scs" in phase 2
- File "load_ph3.scs" in phase 3

Syntax of the "use" command:

use <FileName> **for** (<Number>[,<Number>])

- **use**
Download command
 - If the command is in "load_ph2.scs", the content of <FileName> is sent in phase 2
 - If the command is in "load_ph3.scs", the content of <FileName> is sent in phase 3
- <FileName>
Sercos file whose content is to be transmitted to the drive <Number>
- <Number>
System drive index (if <File name> is to be transmitted to several drives, separate the individual numbers by commas)

Syntax of the "ccuse" command:

ccuse <FileName>

- **ccuse**
Command to implement a CC configuration file
 - This command may only be used in "load_ph2.scs"
 - Up to 10 CC configuration files can be implemented
- <FileName>
Sercos configuration file in Sercos list format (binary file with preceded lengths)

Example:

Excerpt from the file "load_ph2.scs"

use p2ilin00.scs for (1,2,3) ; Parameter file for linear axis

use p2ilin01.scs for (4,9) ; Parameter file for a linear axis with an external encoder



The parameter **ModeScsFiles** "Download Sercos Data in Phase 3" (1050 00022) specifies the behavior of the NC for all Sercos files entered in "load_ph3.scs". If hiding is permitted, the files are sent to the corresponding drive only when required.

All files which can be used for the automatic download in "load_ph2.scs" and "load_ph3.scs" via the "use" command can include:

- comment lines
- Lines for drive parameterization

```

NC Editor /mnt/load_ph2.scs
;
;-----
; load_ph2.scs Control of the SERCOS-Phase 2
; Axisnumber 1..64
;-----
;
; Diagnostics of the SERCOS-Phase 2
; Help for looking for initialising bugs
dbg p2dbg.scs ;Diagnostics turn on when delete ";"
; The transfer of the ID-number will be note
;-----
;
; Settings of the SH_00942 fuer axis und spindles
;
use p2common.scs for (1,2,3,4,5,6,7,8) ; common parameters (Data save on FEP)
use p2ilin00.scs for (1,2,3,4,5,7,8) ; parameter for linearaxis
use p2icax00.scs for (6) ; parameter for linearaxis 6
use p2linX2.scs for (3) ; parameter for linearaxis 3
use p2linZ2.scs for (4) ; parameter for linearaxis 4
;use p2spinWS.scs for (6) ; parameter for workpiecespindle 6
;-----

```

Fig. 7-209: Example of a Sercos file

Comment lines **always** start with a semicolon (;). The NC interprets all characters from the semicolon up to the end of the line as comment. Comments are not transferred to the drives. They are used to improve structure and clarity.

Lines for the drive parameterization are structured as follows:

<ID number> = <Value> [; <CommentText>]

The following applies:

- <ID number>:
Sercos parameter in the format S-x-xxxx.x.x or P-x-xxxx.x.x (S- and P-parameters). Refer to the drive documentation for available Sercos parameters
- <Value>:
Parameter value in the following formats:
Decimal: e.g. 500
Binary: e.g. 0b...
String: e.g. "Text"
Parameter list: e.g. (S-0-0047, S-0-0189)
- <CommentText>:
Comment; has to be separated from <Value> by at least one space and one semicolon

Example:

```
;This is a comment line (comment)
S-0-0121 = 1 ;Input revolutions (parameter with com.)
S-0-0122 = 1 (decimal value)
S-0-0032 = 0b0011 (binary value)
S-0-0142 = "Application block" (string)
S-0-0016 = (S-0-0051, S-0-0189) (ID list)
```

If **all** adapted Sercos files (*.scs) required for your application are already available, copy all relevant Sercos files into the "root directory" ("/") of the NC (for the directory structure, see the "IndraWorks" documentation). To enable or disable, proceed as described in the following sections.

If **no** Sercos files (*.scs) have been generated for your application, create new files in the "root directory" ("/") of the NC by means of the editor, or copy the example files enclosed in the delivery into the "root directory" ("/") and adapt them to your requirements (see Handling Instruction). The example files are located in the directory "//root/usrfep/".

Activating:

1. Remove the character ";" in the files "load_ph2.scs" and/or "load_ph3.scs" to the left of the relevant "use" commands.

As a result, the NC will no longer interpret these lines as comment lines but as download commands the next time it is started.

2. Save the modified file(s)
3. Create a backup copy of the file(s)

*Example:**Automatic download (in phase 2) of the files*

- "p2linall.scs" in the drives with the system drive numbers 1, 2 and 4
- "p2lin3.scs" in the drive with the system drive number 3

Excerpt from the file "load_ph2.scs"

```
;
use p2linall.scs for (1,2,4)
use p2lin3.scs for (3)
;
```

Disabling:

1. Set the character ";" in the files "load_ph2.scs" and/or "load_ph3.scs" to the left of the relevant "use" commands.

As a result, the NC will no longer interpret these lines as download commands, but as comment lines the next time that it is started.

2. Save the modified file(s)
3. Create a backup copy of the file(s)

Example:

Automatic downloading is to be deactivated for both files from the example shown above:

Excerpt from the file "load_ph2.scs"

```
;
;use p2linall.scs for (1,2,4)
;use p2lin3.scs for (3)
;
```



If backup copies of all relevant Sercos files exist in the "user FE-PROM" ("/usrfep"), it is not sufficient to merely delete or rename the relevant Sercos files in the "root directory" ("root") to deactivate automatic parameter downloading. The NC also searches for these files in the "user FE-PROM (usrfep)" if it cannot find them in the "root" directory.

Logging transmitted Sercos parameters

All Sercos parameters transmitted in phase 2 or phase 3 can be logged for diagnostic purposes. Program the "dbg" command in the "load_ph2.scs" file (for phase 2) or in the "load_ph3.scs" file (for phase 3).

Prerequisite:

The "dbg" command has to be programmed prior to the first "use" command.

Syntax of the "dbg" command:

dbg <FileName>

- dbg
"Create log" command
 - If the command is in "load_ph2.scs", the logging function is activated in phase 2
 - If the command is in "load_ph3.scs", the logging function is activated in phase 3
- <FileName>
Name of the file which is to be in the log

Activating:

1. The function can be activated by removing the character ";" to the left of the "dbg" command in the file "load_ph2.scs" or "load_ph3.scs".

As a result, the NC will no longer interpret the line as a comment line, but as a command to log the Sercos parameters the next time that it is started.

2. The file has to be saved when exiting the editor.

Disabling:

1. The function can be deactivated by setting the character ";" to the left of the "dbg" command in the file "load_ph2.scs" or "load_ph3.scs".

Thus, the NC will interpret the line as a comment line the next time that the control starts up.

2. The file has to be saved when exiting the editor.



The diagnostic function is already in the Sercos files delivered by Bosch Rexroth.

Logging Sercos timing

For diagnostic purposes, the NC can log the Sercos timing starting from phase 3. To do so, program the "opt" command in the "loadph3.scs" file.

Prerequisite:

The "opt" command has to be programmed after the last "use" command.

"opt" command syntax:

opt -m <FileName>

- opt -m
"Logging of Sercos timing" command
- <FileName>
Name of the file which is to be in the timing log



This function should only be used during commissioning!

Activating:

1. The function can be activated by removing the character ";" to the left of the "opt" command in the file "load_ph3.scs".
As a result, the NC will no longer interpret the line as a comment line, but as a command to log the Sercos timings the next time that it starts up.
2. The file has to be saved when exiting the editor.

Disabling:

1. The function can be deactivated by setting the character ";" to the left of the "opt" command in the file "load_ph3.scs".
Thus, the NC will interpret the line as a comment line the next time that the control starts up.
2. The file has to be saved when exiting the editor.

Handling instruction: Creating SCS file

IW-Engineering/file system: Adapt SCS files

1. Select root directory
Required SCS files (templates under root/feprom/scsindra/...):
 - load_ph2.scs
 - load_ph3.scs
 - p2common.scs
 - p2xxx000.scs – This file can have another name, depending on the drive.
2. Copy file "p2xxx000.scs" as many times as there are drives in the ring and name them according to the names of the drives.
Identical drive types can be addressed using one SCS file:
 - lin ;for linear axis
 - rot ;for rotary axis
 - spin ;for spindle
 - cax ;for the C-axis (spindle with rotary axis functions)
3. Double-click on the SCS files to open and adapt them

```

NC Editor /mnt/load_ph2.scs
;
;-----
; load_ph2.scs Control of the SERCOS-Phase 2
; Axisnumber 1..64
;-----
;
; Diagnostics of the SERCOS-Phase 2
; Help for looking for initialising bugs
dbg p2dbg.scs ;Diagnostics turn on when delete ";"
; The transfer of the ID-number will be note
;-----
;
; Settings of the SH_00942 fuer axis und spindles

use p2common.scs for (1,2,3,4,5,6,7,8) ; common parameters (Data save on FEP)
use p2lin00.scs for (1,2,3,4,5,7,8) ; parameter for linearaxis
use p2icax00.scs for (6) ; parameter for linearaxis 6
use p2linX2.scs for (3) ; parameter for linearaxis 3
use p2linZ2.scs for (4) ; parameter for linearaxis 4
;use p2spinWS.scs for (6) ; parameter for workpiecespindle 6
;-----

```

Fig. 7-210: Example of a Sercos file



The SCS templates (files) are stored in the root directory of the control.

The file "load_ph3.scs" primarily consists of remarks; only one log file is activated with the entry "dbg p3dpg.scs". This diagnostic file logs the data transfer to the drive. This diagnostic file can also be activated in the SCS file "load_ph2.scs".

If it is stored elsewhere in the file structure of the control, ensure that the data is not accidentally overwritten in another location.

Handling instruction: Activate an NC restart

After the SCS files have been successfully configured, an NC restart must be activated so that the control can read the current SCS files and start up the drives.

IW Engineering/configuration: Activate an NC restart

Activate an NC restart



Only after the NC control is restarted, the modified data (parameters, SCS files) is applied.

PLC commissioning

General information

Activating axes/spindles in PLC user program

The required connection of the inputs and outputs on the axis and the channel interface and the creation of the axis and channel interface are described in this handling instruction.

Handling instruction: Creating a channel/axis/spindle interface

This handling instruction describes the creation of a channel/axis/spindle interface in IndraLogic.

IW Engineering/IndraLogic: Creating a channel interface

This handling instruction describes how the user can create and connect a channel interface.

		Instruction
Instruction:	PLC data configuration	Channel interface

IW Engineering/IndraLogic: Creating an axis interface

This handling instruction describes how the user can create and connect an axis interface.

		Instruction
Instruction:	PLC data configuration	Axis interface

IW Engineering/IndraLogic: Creating a spindle interface

This handling instruction describes how the user can create and connect a spindle interface.

		Instruction
Instruction:	PLC data configuration	Spindle interface

Offline parameterization

There is the option to parameterize the drive parameters offline instead of parameterizing them online. Therefore, the control has to be switched to offline parameterization.

The offline parameterization is only relevant for drive data. Control data cannot be parameterized offline.

Prerequisites:

To use the offline parameterization, offline data has to be generated first.

That can be carried out as follows (MTX is online):

- Call the function **Update Offline Parameters** at the MTX node
- Call the function **Adjust offline data** at the drive node
- Going offline with the MTX

When going offline with the MTX, the data of all drives online at this point in time are compared with each other. This function depends on the setting in the "Options" dialog (see [fig. 7-211 "Options dialog: Offline parameterization of the MTX" on page 391](#)).

Data adjustment

When going online with the drive, all possibly existing offline data is compared to the data in the drive. Differences are displayed and can be applied in the drive or in the offline parameterization. This behavior can be disabled in the **Options** dialog (see [fig. 7-211 "Options dialog: Offline parameterization of the MTX" on page 391](#)).

Options

To set up the offline parameterization, there are two settings in the **Options**(Tools ► Options).

Checks can be switched off when going online/offline.

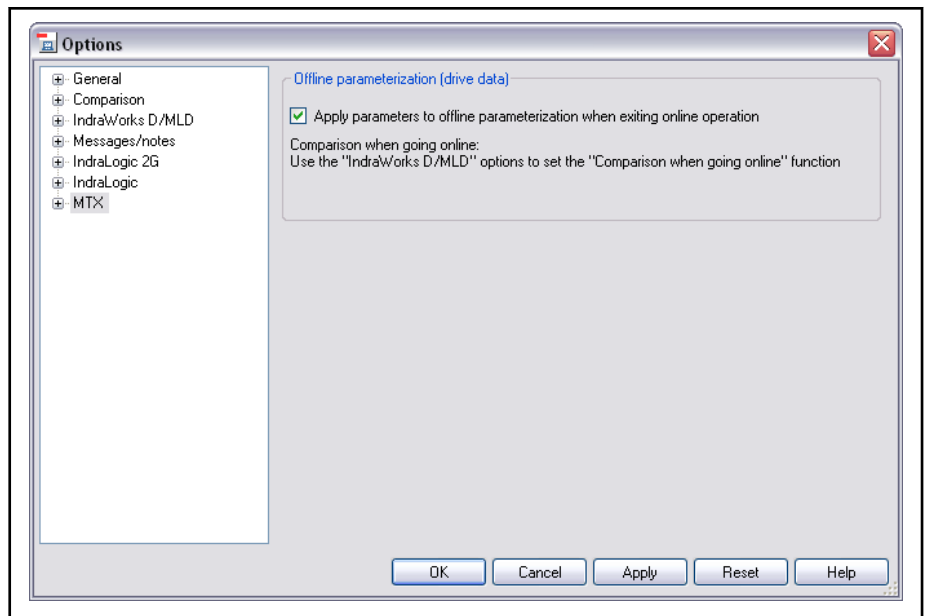


Fig. 7-211: "Options" dialog: Offline parameterization of the MTX

Using the offline parameterization

First, the control has to be switched to offline parameterization.

The individual drives can also be switched to offline parameterization via mouse click or via the function **Drive communication online/offline**. The drive data can also be archived.



Other options of the offline parameterization of a drive are described more in detail in the drive documentation.

Sercos fast startup

Use the fast startup function to avoid the transfer of Sercos parameters that already have the correct parameter value in the drive.

Thus, the Sercos startup is accelerated in case of an unmodified system configuration.

The function calculates the checksum of the entire system configuration and saves this checksum as system date. Upon the control startup, the new checksum is calculated and compared to the saved checksum.

Configure and activate Sercos fast startup

The "s3_faststartup_list.scs" file contains all Sercos parameters that are not transferred during a fast startup. All Sercos parameters in scs files contained in S-0-0192 (IDN list of operating data to be backed up) can be included in "s3_faststartup_list.scs". An example file can be found in the "/root/feprom/scsindra" control directory. By copying the "s3_faststartup_list.scs" file to the root or user feprom, the fast startup function is activated.

The following system settings are taken into consideration when generating the checksum:

- System configuration (MACODA)
- Sercos parameters (SCSP)
- Sercos scs files
- Per drive, checksum of operating data (S-0-0531) to be backed up

The current checksum is saved in the "SysSercCfgInfo.SercConfigChecksum" system date.

7.12.2 Sercos device failure

Sercos ring behavior in case of Sercos device failure

From MTX version 13V06, the Sercos ring behavior in case of Sercos device failure changed.

The Sercos device failure does not necessarily result in a complete Sercos communication cancellation. After a Sercos device failure or a failure of parts of the Sercos ring, communication with all other Sercos devices that can physically still be reached is maintained.

Possible causes of failure:

- Sercos device switch off
- Disconnecting a Sercos connection line
- Failure in Sercos device

The user can display the remaining Sercos devices in the IndraWorks Operation screen "Sercos diagnostics".

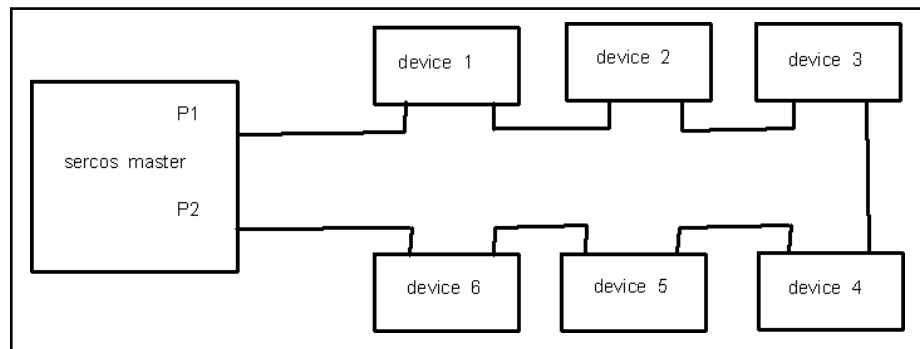


Fig. 7-212: Sercos ring in "ring topology"

In the figure above, a Sercos ring with 6 Sercos devices is shown. The Sercos ring is connected to the master ports P1 and P2 ("ring topologies"). The Sercos data is redundantly sent from both master ports and also received at both master ports.

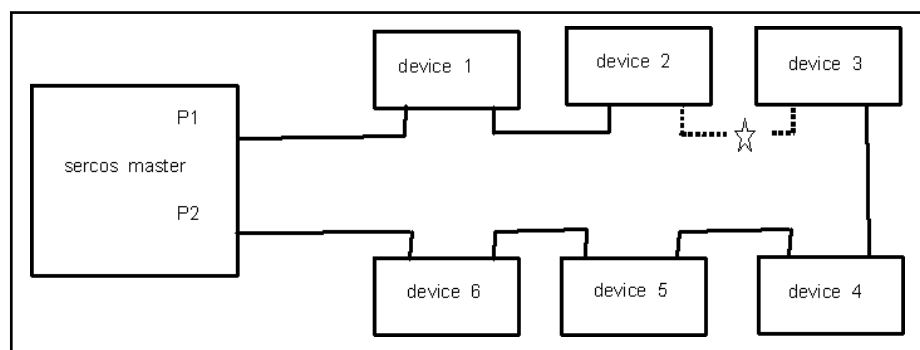


Fig. 7-213: Sercos ring with interruption

In the figure above, the Sercos connection between Sercos device 2 and Sercos device 3 is interrupted. The redundant communication ("ring topology") is changed to "line topology" is lost but the Sercos master can still communicate with all Sercos devices.

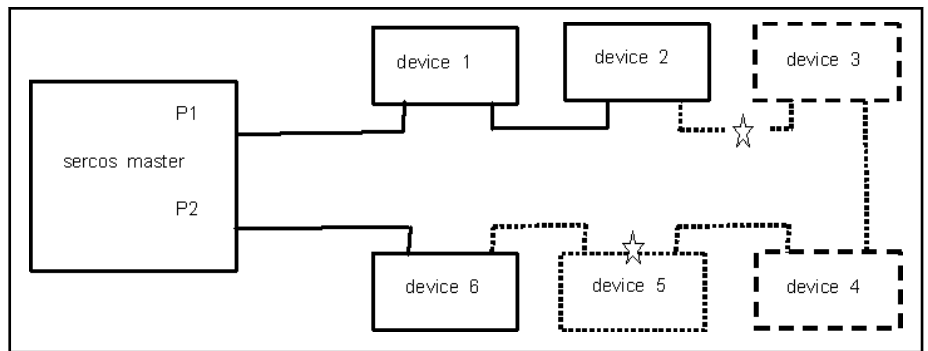


Fig. 7-214: Figure 3, Sercos device failure

In figure 3, the connection line of the Sercos devices 3, 4 and 5 fails, e.g. due to a failure of the 24V supply. Now, the Sercos devices 3, 4 and 5 cannot be contacted anymore. The Sercos ring still remains in Sercos phase CP4.

Error reaction

The Sercos device failure results in an error message and an error reaction. The error reaction in the MTX depends on the device type. The failed Sercos device executes the parameterized error reaction or the error reaction specified by the manufacturer in case the communication is interrupted.

A question mark is displayed in the control display if not all Sercos devices can be contacted.

MTX display: "P 4 RUN" --> "P 4? RUN"

Error reaction in MTX in case of Sercos drive failure

The corresponding NC channel or the running NC program is set to standstill by decelerating all NC channel axes.

Error reaction in MTX in case of Sercos I/O device failure

The input data of the Sercos I/O device received at the PLC interface are reset.

Deleting the Sercos communication error

A failed Sercos device can report back independently as soon as the cause of failure has been removed. The control resumes the communication to the reporting Sercos device after a single device startup phase after CP4 (hotplug).

The single device start-up after CP4 is only possible for Sercos devices supporting hotplug.

Sercos drives that have been included in the Sercos ring again can only be traversed after a reset and after the Sercos communication error has been deleted. Sercos I/O devices do not require deleting of errors.

Startup of the Sercos ring despite missing Sercos devices

If not all parameterized Sercos devices report back during the activation phase of the control after a waiting time of several minutes, the Sercos ring is still runs up to CP4. Handle missing Sercos devices like failed Sercos devices.

The missing Sercos devices can report back via hotplug if the Sercos configuration is saved in the control. The Sercos configuration is saved after a successfully completed Sercos CP4 startup with all parameterized Sercos devices.

7.12.3 Drive-integrated Safety Engineering

Basic method of functioning

The commissioning of the integrated safety engineering is briefly described in this chapter. An extensive explanation, with additional sample applications, can be found in the documentation **"Rexroth IndraDrive Integrated Safety Technology According to IEC 61508"** (part no. R911327664).

Description *With regard to the use of safety engineering, it is distinguished between:*

- **normal mode** and
- **Special mode**

In the special mode, the following is possible:

- **Special mode with standstill ("Safe standstill")**
- **A special mode with motion ("Safe motion")**

"Safe Stop 1" or "Safe Stop 2" can be configured for special mode with standstill.

Regardless of whether the operation is normal or special, "Safe Stop 1 (EMERGENCY STOP)" can also be used as Safe stop:

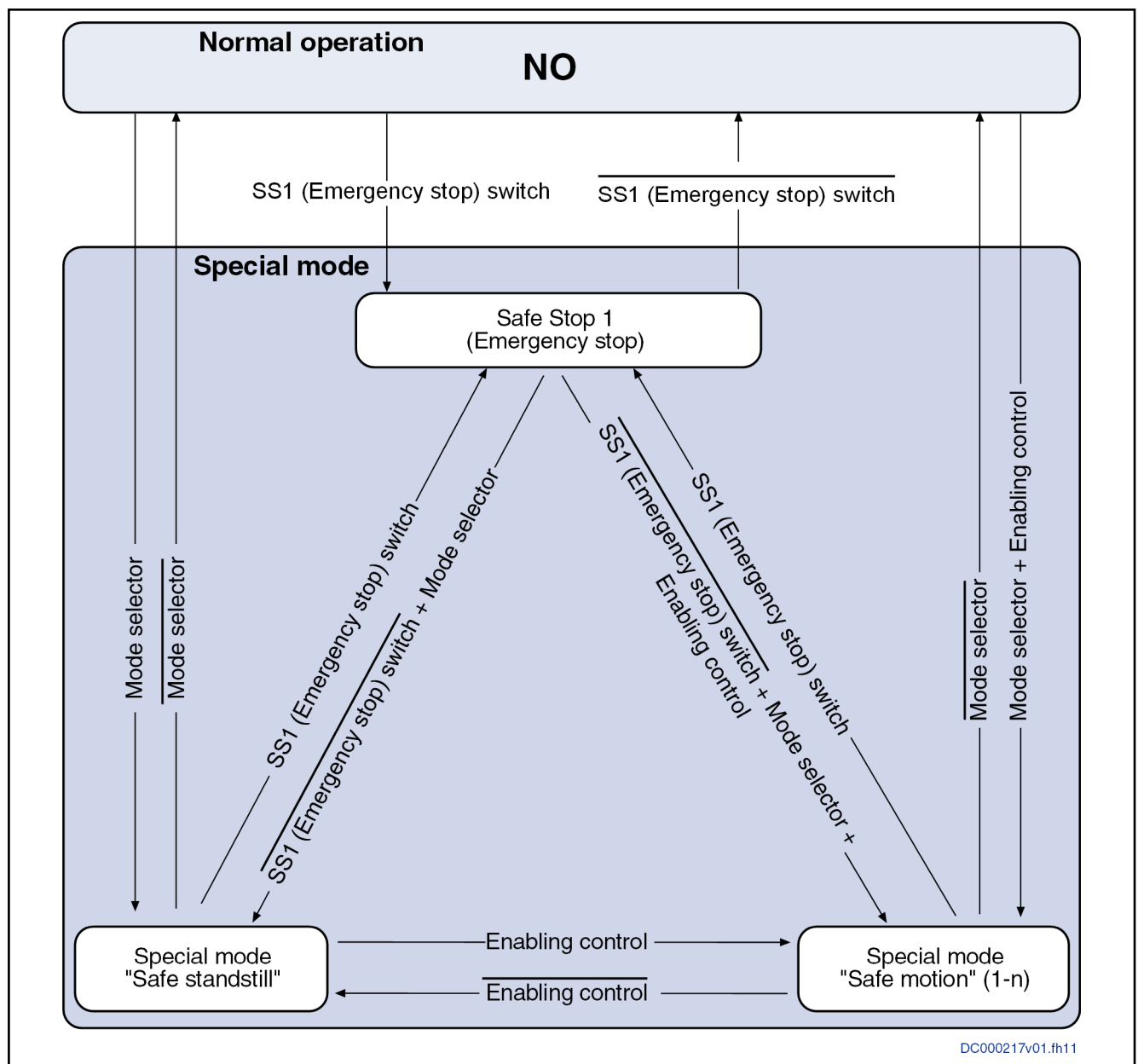


Fig. 7-215: Safety engineering operation modes

Overview of the operating modes

"Safe standstill" Safety functions

- Safe Stop 1** "Safe Stop 1" corresponds to stop category 1 according to EN 60204-1. The drive cannot generate torque/force and thus cannot generate dangerous movements. No monitorings are active in "Safe Stop 1". When the drive lock is active, "SS1" is shown on the display of the Rexroth IndraDrive control device.
- Safe Stop 2** "Safe Stop 2" corresponds to stop category 2 according to EN 60204-1. In safety function "Safe Stop 2", a dual-channel monitor prevents the drive from carrying out dangerous movements due to errors. When the drive lock is active, "SS2" is shown on the display of the Rexroth IndraDrive control device.

Safe Stop 1 (EMERGENCY STOP)

"Safe Stop 1 (EMERGENCY STOP)" corresponds to stop category 1 according to EN 60204-1.

The safety function "Safe Stop 1 (EMERGENCY STOP)" corresponds to the safety function "Safe Stop 1" but it is canceled by actuating an enabling device.

When the drive lock is active, "SS1 ES" is shown on the display of the Rexroth IndraDrive control device.

This is used, for example, in spindle drives to exchange tools manually and to handle axes for movements by hand.

"Safe motion" Safety functions**Safely-limited velocity**

In the safety function "Safely-limited velocity", a dual-channel monitor prevents the drive from exceeding the specified velocity limit values (P-0-3244, P-0-3254, P-0-3264, P-0-3274).

When the movement lock is active, "SMM" is shown on the display of the Rexroth IndraDrive control device.

A motion is enabled using an enabling device (ED). The actuation time of the enabling device is monitored.



For more safety functions, refer to documentation "Rexroth IndraDrive Integrated Safety Technology According to IEC 61508" (part no. R911327664).

In special mode "Safe Motion", the operation between Motion 1 and Motion 2 can be switched using the SI switch 1 (S1).

Example: Installation and PLC interface

Application structure

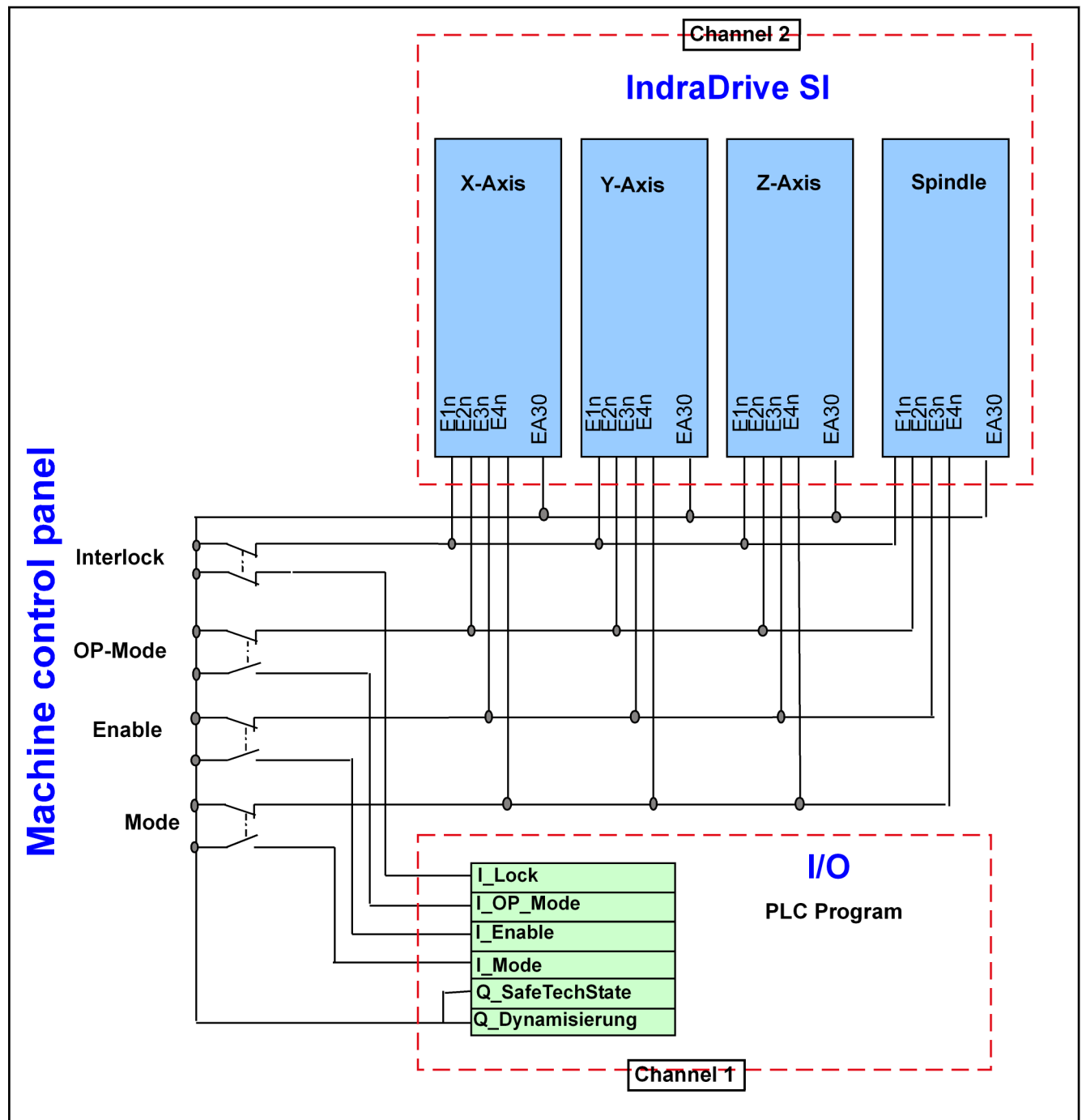


Fig. 7-216: Wiring

Example The safe 2-channel activation of the safety engineering occurs in channel 1 using a standard input of the PLC I/O level and in channel 2 directly by the safety module of the drive control devices (see previous figure).

Since the safety functions should be applied for all axes and spindles simultaneously, the input signals of all control devices (signals of safety module channel 2) are to be cross-connected.

For channel 1 (PLC I/O level), the safety signals can be passed on in the PLC user program. Therefore, a total of only one input per input signal is required for all the axes.

Due to the dynamization of the signals that is required for safety engineering, all sensors for activating the safety functions are to be supplied using dynamization signal "Q_Dynamization".

The signal that is required for this purpose is to be generated in the PLC user program and provided for all drives via IO30

When the dynamization signal is generated, observe the dynamization limit values parameterized in the IndraWorks Drive (period duration and pulse duration) under consideration of the runtimes in the PLC and peripherals. If the values are exceeded, the drives are switched off and an error is issued.



It is recommended that dynamization be executed as "isolated dynamization". As a result, the dynamization pulse can be set to 50 ms, which shortens the reaction times.

PLC program part

Assign the signals as follows in the PLC user program:

Safety-oriented machine function	Short	I/O signal
Safe Stop 1 (EMERGENCY STOP)	SS1 ES	qAX_SafDrvLock
Special mode (for open safety equipment)	BA	qAx_SafOpModeSwitch
Acknowledgement key (for movement when safety equipment is open)	CK	qAx_SafEnablCtrl
Switching to safe movement	S1	qAx_SafSwitch1
Dynamization	DYN	qAx_SafTecState

Tab. 7-54: Interface signals

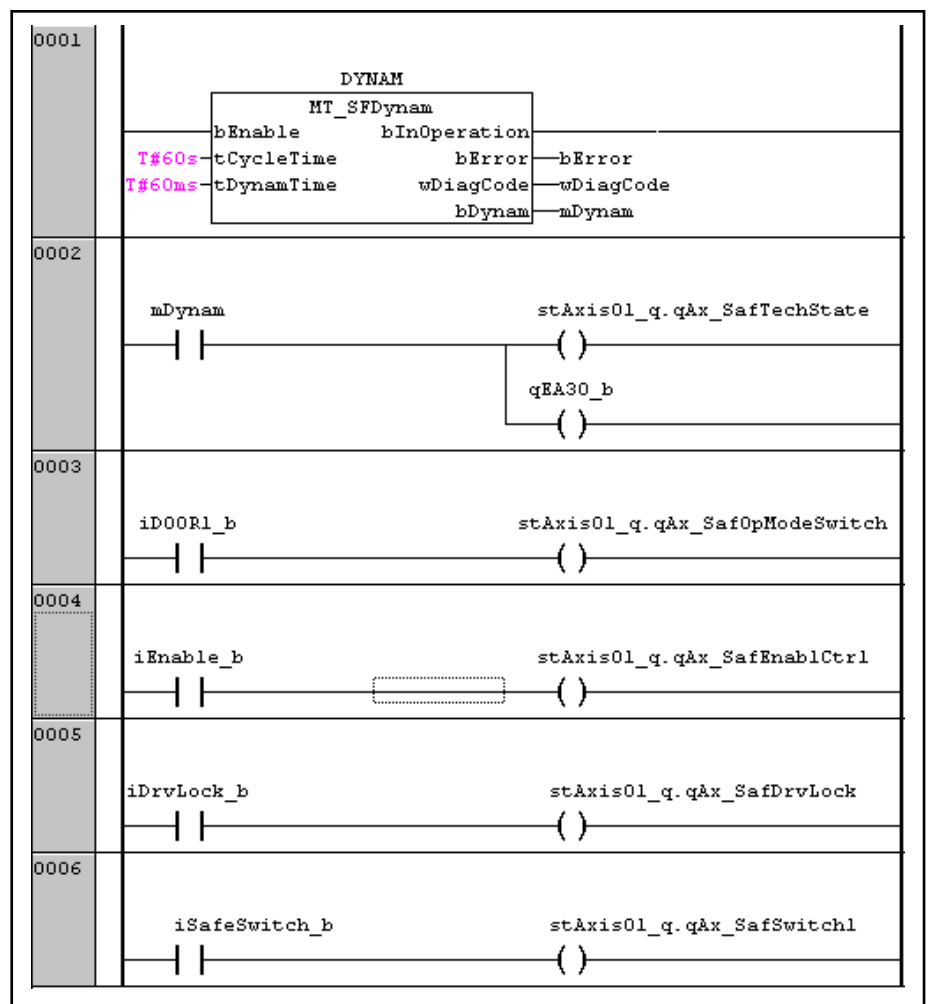


Fig. 7-217: PLC program example

SCS configuration

General information

In the SCS files for axes with safety engineering, make the divergent setting for the parameters for Sercos communication depending on the axis type:

Excerpt from an SCS file:

Program:

```

;-----
; Configuration of cyclic telegram
S-0-1050.1.6 = (135,51,144,P-0-3213,P-0-3215) ; AT
;          |   |   |   |   -SI operating state
;          |   |   |   |   -SI operating state
;          |   |   |   |   -Signal status word
;          |   |   |   |   -Actual position value 1
;          |   |   |   |   -Drive status
;
S-0-1050.0.6 = (134,47,145) ; MDT
;          |   |   |   |   -Signal control word
;          |   |   |   |   -Position command vale
;          |   |   |   |   -Master control word
;
;-----
; Signal status word
S-0-26 = (S-0-403,P-0-3213,P-0-3214)
;          |   |   |   |   -SI status output controller
;          |   |   |   |   -Status safe position
;          |   |   |   |   -Status Actual position value
; Bit numbers
S-0-328 = (0,7,0)

```

```

;          | | -SI status output controller
;          | -Status safe position
;          -Status Actual position value
;
;-----
; Signal status word
S-0-27 = (P-0-3212,P-0-3212,P-0-3212,P-0-3212,P-0-3212)
;          |         |         |         |         -Dynamization
;          |         |         |         |         -Safety switch
;          |         |         |         |         -Enabling switch
;          |         |         |         |         -Drive anhebt
;          |         |         |         |         -Operating mode
; Bit numbers
S-0-329 = (0,1,2,3,10)
;          | | | | -Dynamization
;          | | | | -Safety switch
;          | | | | -Enabling switch
;          | | | | -Drive inhibit; -Operating mode ;-----

```

Handling instruction: Commissioning of Safety Engineering

Following is a description of the step-by-step commissioning of the IndraDrive safety engineering integrated into the drive using the MTX.

IW Engineering/IndraLogic: Adjusting the PLC program

		Instruction chapter "Handling instruction: Adjusting the PLC program" on page 400
Instruction:		Adjusting the PLC program

IW Operation/program: Configuration of SCS files

		Instruction chapter "Handling instruction: Configuration of SCS files" on page 402
Instruction:		Configuration of SCS files

IW Engineering/Configuration: Setting the machine parameters

		Instruction chapter "Handling instruction: Setting the machine parameters" on page 402
Instruction:		Setting the machine parameters

IW Engineering/Sercos: Activating the Safety Engineering in the drive

		Instruction chapter "Handling instruction: Activating the Safety Engineering in the drive" on page 403
Instruction:		Activating the Safety Engineering in the drive

Handling instruction: Adjusting the PLC program

This handling instruction describes the adaptation of the PLC program to activate the safety engineering.

IW-Engineering/IndraLogic: Inserting interface signals and dynamization

1. Double-click the "IndraLogic" node in Engineering Desktop.
The PLC is opened.



We recommend that programming be carried out in a separate subroutine in programming language LD. This should be kept as brief as possible.

2. To generate the dynamization impulse, incorporate the block "FB Dynam" in the PLC program and activate it according to the times desired for dynamization.

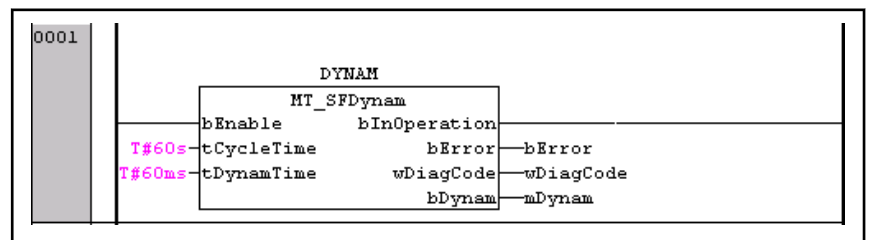


Fig. 7-218: *FB_Dynam*

3. Furthermore, the interface signals for selecting the desired safety functions are to be wired in the PLC program.

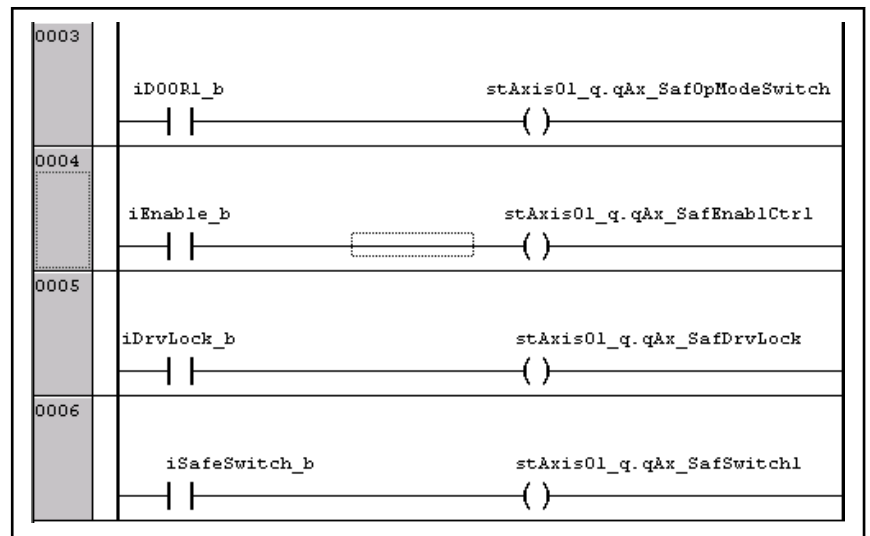


Fig. 7-219: *Interface signals*

4. The dynamization impulse must also be transferred to the PLC interface. On the outside, the dynamization output serves as the supply for the selection signals of the 2nd channel.

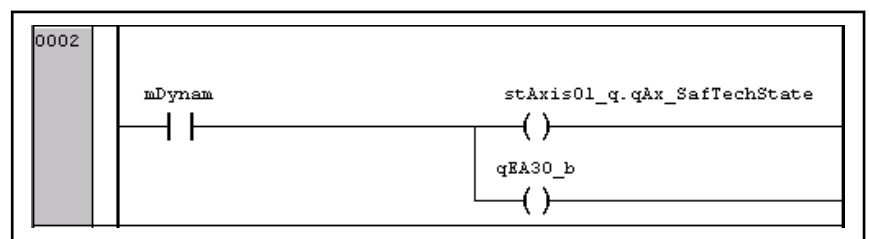


Fig. 7-220: *Dynamization interface*

Documentation:	IndraLogic programming	Safety engineering

Handling instruction: Configuration of SCS files

This handling instruction describes the changes that must be made to the SCS files to activate the safety engineering.

IW Operation Desktop/program: Adjust SCS files

Open the SCS files and enter the parameters to be transferred.



The SCS files are located in root or userfep.

Example excerpt from an SCS file:

Program:

```

;-----
; Configuration of cyclic telegram
S-0-1050.1.6 = (135,51,144,P-0-3213,P-0-3215) ; AT
;           |   |   |   |   -SI operating state
;           |   |   |   |   -SI operating state
;           |   |   -Signal status word
;           |   -Actual position value 1
;           -Drive status
;
S-0-1050.0.6 = (134,47,145) ; MDT
;           |   |   -Signal control word
;           |   -Position command vale
;           -Master control word
;
;-----
; Signal status word
S-0-26 = (S-0-403,P-0-3213,P-0-3214)
;           |   |   -SI status output controller
;           |   -Status safe position
;           -Status Actual position value
; Bit numbers
S-0-328 = (0,7,0)
;           |   |   -SI status output controller
;           |   -Status safe position
;           -Status Actual position value
;
;-----
; Signal status word
S-0-27 = (P-0-3212,P-0-3212,P-0-3212,P-0-3212,P-0-3212)
;           |   |   |   |   -Dynamization
;           |   |   |   |   -Safety switch
;           |   |   |   -Enabling switch
;           |   -Drive anhebt
;           -Operating mode
; Bit numbers
S-0-329 = (0,1,2,3,10)
;           |   |   |   |   -Dynamization
;           |   |   |   |   -Safety switch
;           |   |   |   |   -Enabling switch
;           |   -Drive inhibit; -Operating mode ;-----

```

		Instruction chapter "Handling instruction: Configuration of SCS files" on page 402
Instruction:		Create SCS files

Handling instruction: Setting the machine parameters

The handling instruction describes the configurations of the relevant machine parameters for the safety engineering.

IW operation desktop/configuration: Setting the machine parameters

1. The safety engineering function has to be activated using parameter **EnablSafe** "Safety engineering (SAFE)" in the setup (SUP).
2. In the menu **Select Data Group**, select the new item "Safety Engineering (SAFE)".
3. If necessary, set the following parameters for each drive:
 - **EnablSafeTech** set "Enable intelligent safety engineering" (1001 00002) to "yes" for activation.
 - **SupprSafeTechNc** "Hide safety engineering in the NC" (1001 00003)
 - **VelWeightFact** "Scaling factor for safe velocities" (1001 00004)
4. Common for all drives (up to MTX14V06):
NofSafeLimSpeed "Number of used and safely limited velocities" (1001 00005)

Instruction:	Instruction chapter "Handling instruction: Activating the Safety Engineering in the drive" on page 403	Editing machine parameters
Documentation:	IndraDrive Integrated Safety Engineering	Enabling safety engineering

Handling instruction: Activating the Safety Engineering in the drive

This handling instruction describes the activation of safety engineering in the drive using Rexroth IndraWorks Engineering.

IW Engineering/Sercos: Activate safety engineering in the drive

In the last step, the safety engineering is activated in the drive using IndraWorks Drive and the safety engineering wizard.

1. After switching the drives to online, open the folder "Drive-integrated safety technology" in the corresponding drive node.

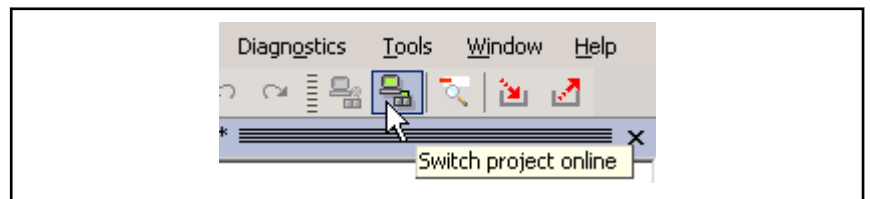


Fig. 7-221: Going online

Control commissioning

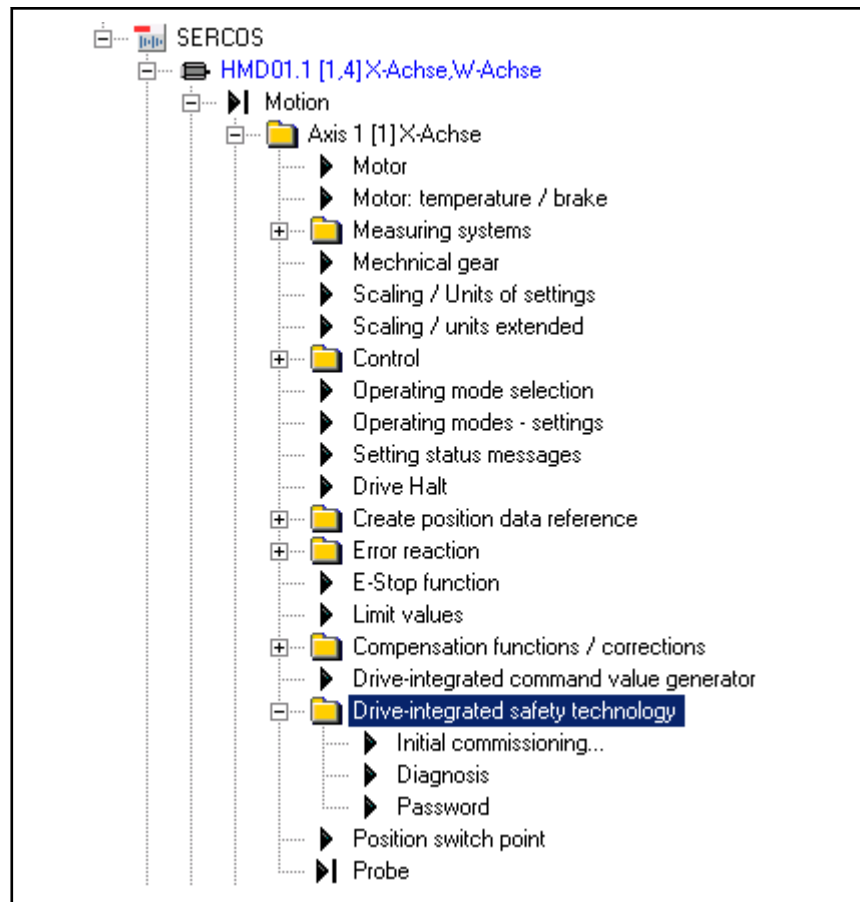


Fig. 7-222: Drive-integrated safety technology

2. Double-click "Initial commissioning" to start the safety engineering wizard of IndraWorks Drive. The wizard guides you through the configuration of the safety engineering within the drive. The number of steps depends on your specific settings.

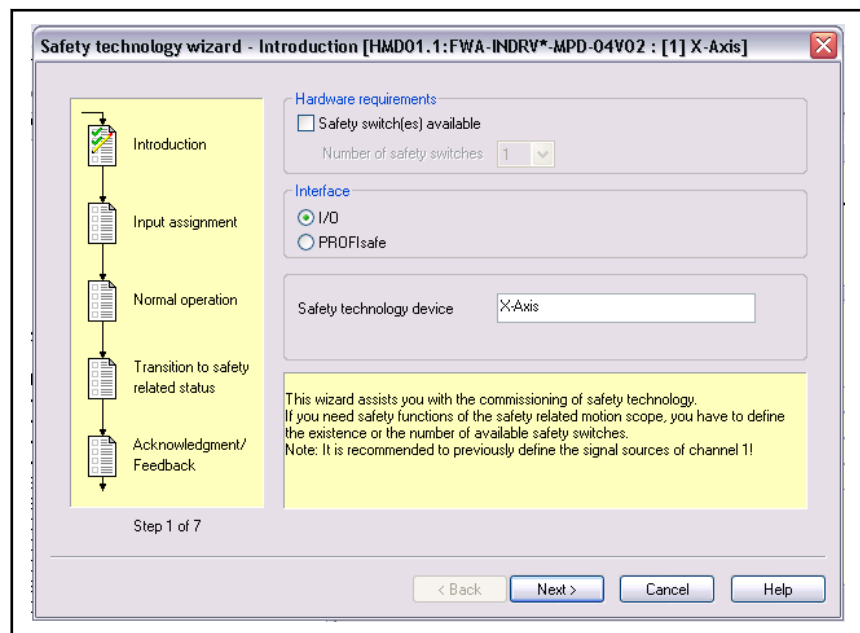


Fig. 7-223: Safety engineering wizard

Documentation:	IndraDrive Integrated Safety Engineering	Enabling safety engineering

"NC ready" bit (P-0-3212 bit 11)

Since axes/spindles have to be moved within a wide velocity/speed range, the transfer times have thus to be set sufficiently high. To reduce this transfer time in case of low velocities/speeds, bit 11 of the parameter P-0-3212 can be used.

If this bit is set, the drive immediately acknowledges the safe state.

The "NC Ready" bits are set from the PLC application. Thus, the interface signal **"qAx_SafRedTransTime"** or **"qSp_SafRedTransTime"** are stored on the axis/spindle interface. It is reported to the drive that the adaptation to the command value system is completed.



The bit has to be reset if the selected safety engineering operating state is enabled or after an application-dependent consistent time.

⚠ WARNING

If the "NC Ready" bit is set before the drive reached its selected safe operating state, the drives are set to standstill and error messages of the safety engineering are reported.

Also refer to the functional and application description "Rexroth IndraDrive Integrated Safety Technology", chapter "Transition to safety-related status".

Using the "NC ready" bits

1. To transmit the bit from the control to the drive, it has to be configured in the SCS file in the signal status word (S-0-0027/S-0-0329).

S-0-0027 = (...,P-0-3212,...)

S-0-0329 = (...,11,...)

2. To set the bit, either the interface signal **"qAx_SafRedTransTime"** or the interface signal **"qSp_SafRedTransTime"** is used in the PLC.

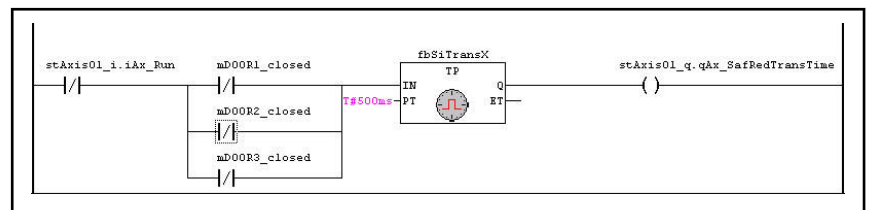


Fig. 7-224: PLC program

Schematic representation

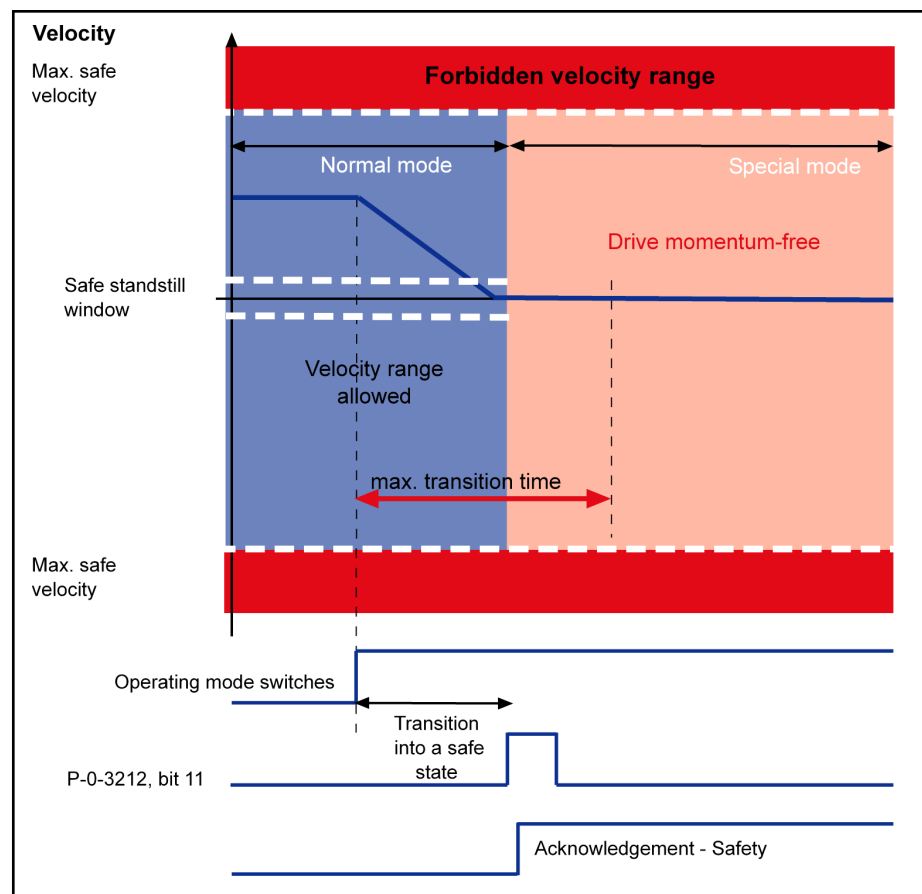


Fig. 7-225: Flowchart

⚠ DANGER**Risk of injury**

An unintended use of the bit can lead to hazardous situations.

Example:

A high-speed spindle is to be switched to ASP and the bit is set while the spindle still rotates with a dangerous velocity.

Consequence:

The drive immediately acknowledges the safe state. The safety engineering immediately identifies the error, switches off the power and the spindle spins. It is now possible that the protecting door is already unlocked even though the spindle is still dangerous.

Parameterizing the Safety Engineering in the drive

Typical report of an axis with Safety Engineering

Normal operation				OK
P-0-3234	Safety related maximum speed	41000.000	mm/min	☐
Transition to safety related status				OK
P-0-3210	Transition to safety related status	NC-controlled		☐
P-0-3220	Tolerance time transition from normal operation	0.5	s	☐
P-0-3221	Max. tolerance time for different channel states	0.3	s	☐
P-0-3225	Tolerance time transition from safety rel. oper.	0.2	s	☐
Acknowledgment/ Feedback				OK
P-0-3210	Safety related feedback ...	for control PLC		☐
Dynamization				OK
P-0-3210	Dynamization source	Slave		☐
P-0-3210	Kind of dynamization	common source		☐
P-0-3223	Time interval for dynamization of safety function selection	60.0	s	☐
P-0-3224	Duration of dynamization pulse of safety function selection	0.2	s	☐
Error reaction				OK
P-0-3210	Reaction to F7 error	Velocity command value reset		☐
Drive interlock				OK
P-0-3233	Velocity threshold for safety related halt	10.000	mm/min	☐
Safety related operational stop				OK
P-0-3230	Monitoring window for safety related operational stop	1.0000	mm	☐
Safety related motion				OK
P-0-3239	Max. activation time of enabling control for safety related motions (SBBs)	common		☐
P-0-3222	Max. activation time of enabling control	30.0	s	☐
Safety related motion 1				OK
P-0-3244	Safety related reduced speed 1	2000.000	mm/min	☐

Fig. 7-226: SI report

The safety-relevant parameters parameterized in the SI wizard are clearly presented in the safety engineering report. How reasonable values are determined is shown in the following.

For a detailed description of the individual parameters, refer to the corresponding parameter description

P-0-3234, Safety-related maximum velocity

The parameter defines a velocity limit that applies to normal mode as well as to special mode (SBB, SBH). Thus, set the maximum permitted velocity for the drive here.

P-0-3210, Transition to safety-related status

The parameter is used to configure safety functions. When using the safety engineering and the MTX, set this parameter to "NC-controlled". The control then reacts on the operation mode changes and reduces the velocity or stops the drive.

P-0-3220 Tolerance time transition from normal operation

The parameter defines the maximum time provided during the transition from normal mode to a safety function. After that time, the command value system

of the drive has to be adapted to the new safety function. This time can be computed as follows:

$$P-0-3220 > \left(\frac{P-0-3234}{MP101000001 * 1000} \right) + t_{control}$$

$$t_{control} = 2 * PLCScan + 2 * IPO + IPO * SHAPE$$

Fig. 7-227:

Typical value for this parameter: 0.5 s - 0.8 s

P-0-3221 Max. tolerance time for different channel states

The parameter defines the maximum permissible time while the input/selection signals (SI states) of the two monitoring channels can differ from each other. When overwriting the time, the error message **"F3141 Plausibility error, selection"** is generated.

This parameter value depends on the selected hardware and the signal run-times. The valid value range is between 0.1 s and 2 s.

P-0-3225 Tolerance time transition from safe mode

The parameter defines the maximum time provided during the transition from a safety function to another safety function. After that time, the command value system of the drive has to be adapted to the new safety function.

$$P-0-3225 > \left(\frac{P-0-3244}{ACCEL * 1000} \right) + t_{control}$$

$$t_{control} = 2 * PLCScan + 2 * IPO + IPO * SHAPE$$

Fig. 7-228:

Typical value for this parameter: 0.2 s - 0.5 s

ACCEL describes the axis acceleration (max. or jog) in this formula. The correct acceleration value can be selected after the following:

```
IF (MP1010 00001 < MP1010 00002) OR MP1010 00002=0 THEN
ACCEL = MP1010 00001
ELSE
ACCEL = MP1010 00002
ENDIF
```

MP1010 00001: Maximum axis acceleration (m/s²)

MP1010 00002: Jog acceleration in (m/s²)

P-0-3282 Safely monitored delay

For the NC-controlled transitions from normal mode or special mode "motion" (SBB) to the "safe standstill" (SH) or the "safe operational stop" (SBH) or "safe drive interlock" (drive inhibit) (ASP), the drive monitors whether it can reach the standstill or the velocity limit of the selected special mode within the "tolerance time transition" (P-0-3220 or P-0-3225). Therefore, check that the deceleration ramp defined in "P-0-3282" is maintained.

```
IF (MP1010 00001 < MP1010 00002) OR MP1010 00002=0 THEN
P-0-3282 = MP1010 00001
```

```
ELSE
P-0-3282 = MP1010 00002
ENDIF
```

P-0-3210, Dynamization source

Source for the dynamization signal. When using the safety engineering and the MTX, set this parameter to "Slave". The dynamization signal has to be created in the PLC.

P-0-3210 Kind of dynamization

The parameter "Kind of dynamization" is to be set to "Common source".

P-0-3223 Time interval for dynamization of safety function selection

This parameter defines the cycle time during which a mandatory dynamization takes place. The value range of this parameter is between 1 and 3600 seconds.

A typical parameter value is 60 s.

P-0-3224 Duration of dynamization pulse of safety function selection

The parameter defines the maximum period of the dynamization pulse. The value range is between 0.1 and 1 s.



An externally generated dynamization signal (typical for MTX) can also be shorter, but may not fall below the minimum pulse duration of 30 ms.

With a typical value of 0.1 s in the drive, the period of the pulse in the PLC can be set to **50 ms**.

The "Period selection of the dynamization pulse" affects the system ability to react, since the evaluation of the safety signals is stopped during the dynamization. Thus, do not choose a very long period.

P-0-3233, Velocity threshold, safe standstill

The parameter defines a velocity threshold either for the special mode "Standstill" or "Drive interlock" (drive inhibit).

Typical values:

- Linear axis: 25 - 50 mm/min
- Spindle: 5 rpm

P-0-3230 Monitoring window for safe operational stop

The parameter defines the maximum valid traversing distance related to the actual value available when SBH was activated.

Typical values:

- Linear axis: 1 mm
- Spindle: 1 degree

P-0-3222 Max. activation time of enabling control

The numerical value entered into the parameter P-0-3222 defines the maximum permissible period of the enabling device. The value range is between 0 and 3600 s.

Typical values: 30 - 60 s

P-0-3244 Safely reduced velocity 1

The parameter P-0-3244 defines the velocity threshold (bipolar) always active in the special mode "Safe motion 1" (SBB1).

Typical values:

- Linear axis: 2000 mm/s
- Spindle: 50 rpm



The same applies to the parameters:

- P-0-3254 Safely limited velocity 2
- P-0-3264 Safely limited velocity 3
- P-0-3274 Safely limited velocity 4

7.13 Safety

7.13.1 MTX control, commissioning the SafeLogic

Introduction

The SafeLogic function module provides the resources for processing safety-oriented application programs. It extends the function control to the Safety control and facilitates processing of the functional application as well as the Safety application on the same control. The Safety function module is available for IndraControl XM compact controls.

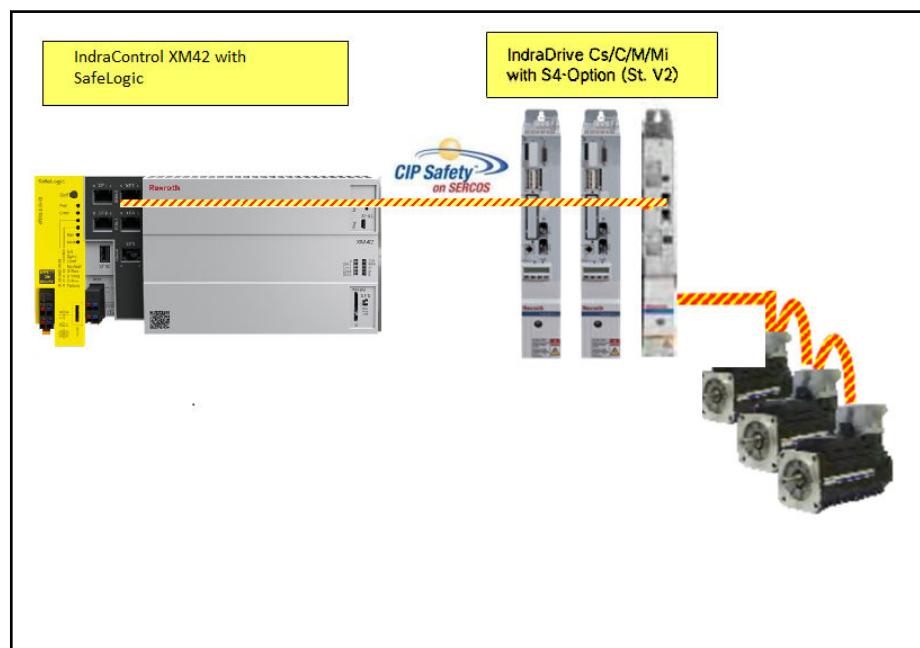


Fig. 7-229: System solution: MTX IndraControl XM with SafeLogic

Standard and Safety periphery (Safety I/O, safe drives, ...) can be operated at the same communication system. The safe data are transferred via the corresponding Safety protocols.



For detailed information on the safety technology with the MTX, refer to the documentation "MTX SafeLogic System Overview".

Mounting

Extension modules can be connected to the left side of the MTX control using the provided function module connected.



The Safety function module is positioned as last module on the left.

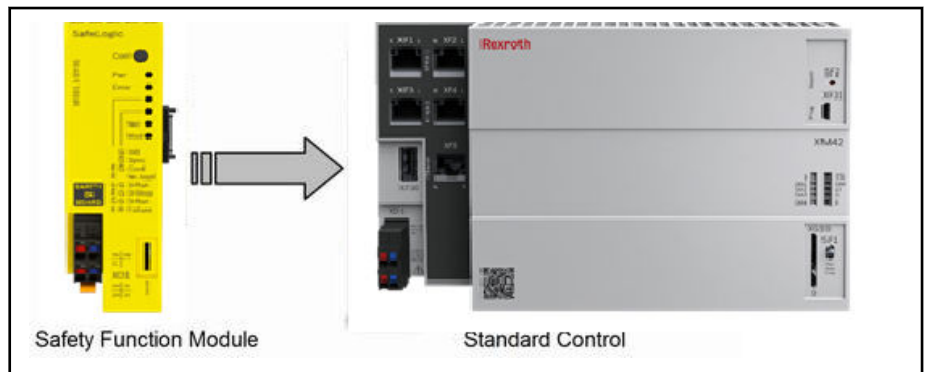


Fig. 7-230: Function module SafeLogic for IndraControl Lx5 controls

For detailed information on mounting, refer to the chapter "Mounting and installing" of the documentation "MTX SafeLogic System Overview".

For detailed information on the Safety function module, refer to the chapter "Safety CPU (CFL01.1-F1)" in the documentation "IndraControl Lxx 14VRS Function Modules".

Configuration

The SafeLogic configuration as function module is available for the following actions:

- Wizard when creating an MTX
- Menu item "Properties" in the MTX context menu

The SafeLogic can be configured as function module in the dialog to configure functions modules.

Fig. 7-231: Dialog: Configure IndraMotion MTX (function modules)



The Safety function module is positioned as last or undermost module.

For detailed information on the configuration, refer to [chapter 7.2 "MTX control, configuration"](#) on page 141.

General information

Project structure

After configuring the Safety function module, a yellow object is displayed below the control in the project tree. It is called Safety CPU. The objects required for the PLC application are automatically created below the Safety FM. The SafeLogic system is completely integrated in IndraWorks. All components can be accessed via the IndraWorks standard functionality.

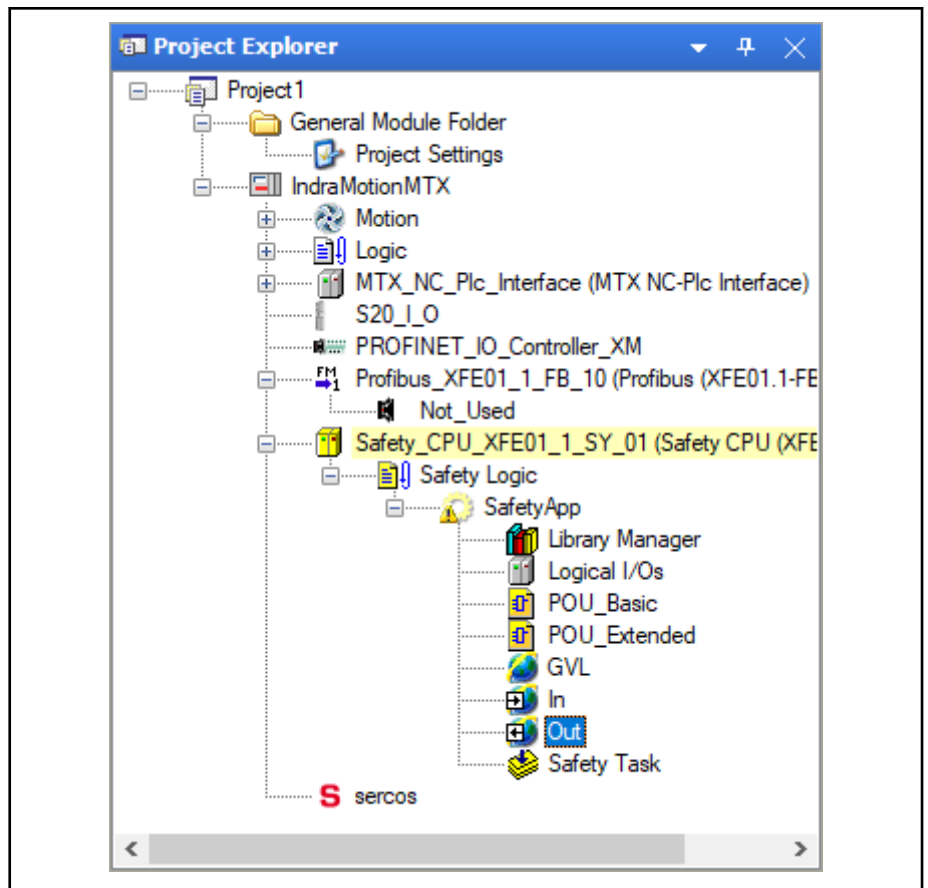


Fig. 7-232: Safety application in the Project Explorer

The IndraWorks operating concept is used to create a project and to operate the interface. IndraWorks commands not available in Safety are visible but cannot be activated.

Yellow Safety structure

Using yellow structures, the Safety objects are highlighted in the project structure and the Safety editors and optically differentiate themselves from standard objects and standard editors.

Additionally, the safe signal flows are also highlighted in yellow. This supports and facilitates the following tasks of the Safety application:

- Engineering
- Verification
- Acceptance

The following objects and nodes exist exactly once below the Safety-FM:

- Safety logic:
Logic node of the Safety control under which exactly one Safety application can be used.
- SafetyApp:
Node under which the Safety application objects can be found.
Object specifying the code execution version and the current status (pin) of the application.
In the editor of the object, the list of the current objects of the Safety application is managed.
- Library manager:
Contains the libraries of the inserted Safety control.

- **Logic I/Os:**
Nodes to which logic I/O objects can be added. These added logic I/Os are used to exchange data and I/Os with the standard control
- **Safety task:**
This object lists all programs that are loaded to the control and can be executed.

The following objects can be added below the application:

- **Basic POU (Safety):**
POU (program or function block) with programming level Basic
- **Extended POU (Safety):**
POU (program or function block) with programming level Extended
- **Global Variable List (Safety):**
Declaration of global variables only valid within the Safety application

All objects can be protected from unauthorized access via the user management.

Logic programming

A Safety application is programmed:

- in POU (program organization units)
- in GVLs (Global Variable Lists)
- in the task object

In the POU, the program code is implemented in IEC 61131-3 language FBD (function block diagram). Main features of the FBD: unambiguity, easy detection of programming errors and clear data flow.

The operating interface and handling of the SafeLogic FBD editor corresponds to the FBD editor of the standard control with regard to operating interface and handling.

Language scope

The FBD language scope is limited according to the language subset defined in PLCopen Basic and Extended. The corresponding selection for the language selection Basic Level or Extended Level is specified by the developer when recreating a POU (program or function block).



For detailed information on the different programming levels, refer to the documentation "MTX SafeLogic System Overview".

- In **Basic Level**, a Safety application can be easily implemented and subsequently verified by linking the already certified function blocks of the PLCopen library ("SafetyPLCopen").
- The **Extended Level** provides additional operators (Boolean, mathematical and others) and conditioned jumps/returns to create more comprehensive Safety applications. Following the development process, they require a more complex verification process.

Data types

The following data types are available in SafeLogic:

- *Standard data types*
 - BOOL
 - INT
 - DINT
 - WORD

- TIME
- *Safety data types*
 - SAFEBOOL
 - SAFEINT
 - SAFEDINT
 - SAFEWORD
 - SAFETIME

Operators SafeLogic provides the following operators:

Operators in basic level

- AND (2 inputs)
- AND (3 inputs)
- OR (2 inputs)
- OR (3 inputs)

Operators in Extended level

- *Boolean operators*
 - AND (2 inputs)
 - AND (3 inputs)
 - OR (2 inputs)
 - OR (3 inputs)
 - XOR
 - NOT
- *Mathematical operators*
 - ADD (2 inputs)
 - ADD (3 inputs)
 - SUB
 - MUL
 - DIV
 - EQ
 - NE
 - LT
 - LE
 - GT
 - GE
- *Other operators*
 - SEL
 - MUX

Safe data flow The safe data flow of the FBD programming is highlighted as follows in Safe-Logic:

- Constants and as constant declared variables are highlighted in yellow
- Variables with safe data type (SAFExxx) are highlighted in yellow
- Data flow of SAFE values in SAFE variables and inputs is highlighted by thick yellow lines

- Function blocks are highlighted in yellow if they have at least one SAFE output
- Operator call boxes are yellow if the output is SAFE

This is the case if:

- in operator AN: the output is SAFE if at least 1 input is SAFE
- all other operators including the conversions: the output is SAFE if all inputs are SAFE

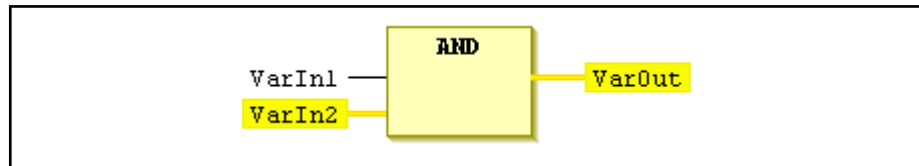


Fig. 7-233: Example of safe data flow: Operator AND with unsafe variable VarIn1, SAFE variables: VarIn2 and VarOut

I/Os

The Safety FM does not have any options to connect I/O devices. The periphery compatible with Safety is connected to the bus systems of the standard control.

The I/O signals to be processed on the Safety FM are configured in the programming system. The input signals intended for the Safety FM are filtered from the pool of the total I/Os on the standard control and are then transported to the Safety FM. The output signals are transmitted from the Safety FM to the standard control and are mixed in the pool of total I/O data by the standard control. Thus, a consistent I/O configuration has to be available on the standard control as well as on the Safety control. If the I/O configuration is not identical, it is displayed in the IndraWorks Project Explorer at the Safety FM node. It is displayed with the "Caution" icon superimposed on the control icon.



In case of I/O configuration inconsistency, no data is exchanged between the periphery and the Safety FM.

Data is exchanged between the standard control and the Safety FM via the same mechanism as with the periphery. This means, that no data is exchanged between the standard control and the Safety FM if there is an I/O configuration inconsistency.

I/O accesses

The mapping variables are edited on the "mapping page" of the logic I/O and are available for code implementation as variable of the category "Global variables".



For detailed information on the safe data exchange between the SafeLogic and the standard control, refer to the documentation "MTX SafeLogic System Overview".

7.13.2 Commissioning drives with CIP Safety on Sercos

Introduction

CIP Safety on Sercos is a protocol based on the CIP Safety mechanism to transmit safety-relevant signals via Sercos interface. The CIP Safety on Sercos protocol has been defined in cooperation with the ODVA (Open DeviceNet Vendor Association) and is certified according to IEC 61508 to SIL 3. An additional safety bus is not needed. The safe data are transmitted in real time with the standard data of the Sercos interface network. The integra-

tion of drive, periphery and safety bus as well as the standard Ethernet in a single network facilitates handling and reduces the hardware and installation costs. Integrated Safety controls and homogeneous Safety solutions can easily be implemented.

Enabling CIP Safety on Sercos on a drive

Preconditions to implement the following steps: An MTX project with created drives.



The different steps to create an MTX project with SL and to commission the drives are described in [chapter 7.13.1 "MTX control, commissioning the SafeLogic"](#) on page 410 or [chapter 7.12.1 "Axis commissioning"](#) on page 358.

To enable the CIP Safety on Sercos configuration on a drive, the drive node has to be created. This is automatically executed if the drive is switched "on-line".

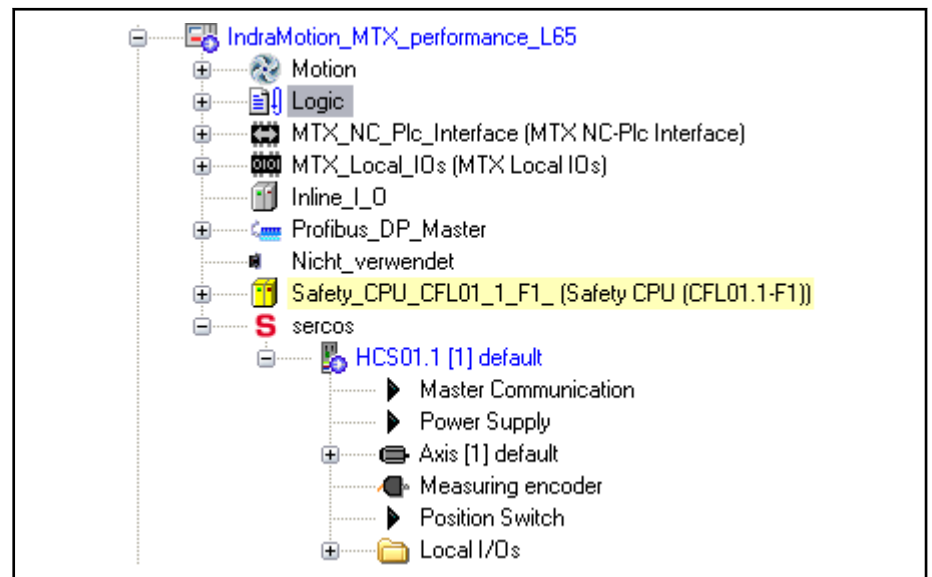


Fig. 7-234: Drive node active



To configure CIP Safety on Sercos without an available drive hardware, create the drives offline.

Drag an IndraDrive from the device library and drop it on the Sercos node. After dragging the drive to the node, a dialog opens to make settings on the Sercos address, firmware and drive type. After making all settings, the drive tree is created below the Sercos node. CIP Safety on Sercos can now be configured as described in the following.

To enable CIP Safety on Sercos on a drive, switch the MTX offline. Subsequently, CIP Safety on Sercos can be configured on the axis node of the drive.

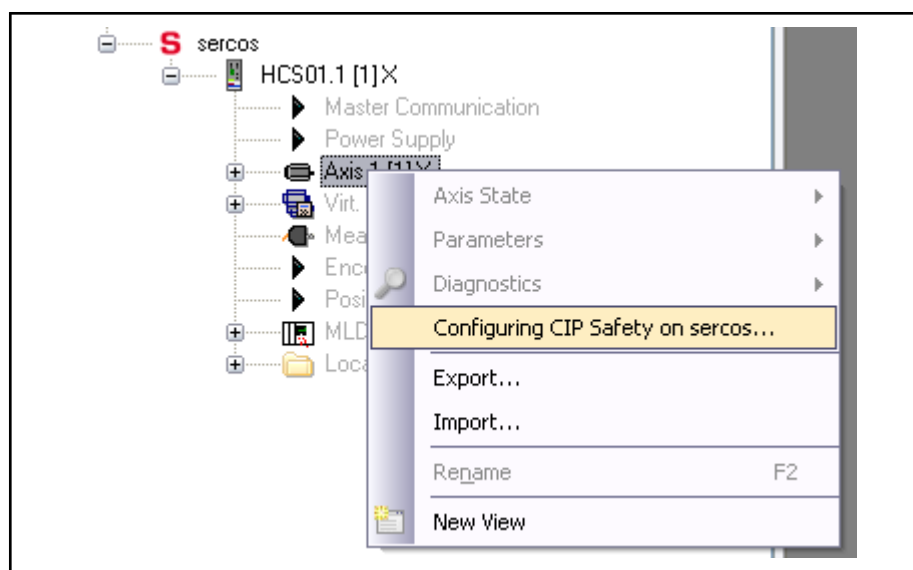


Fig. 7-235: Dialog to select the CIP Safety on Sercos configuration

In the following dialog, CIP Safety is enabled and the drive for SafeLogic is selected. One of several Safety profiles can be selected in the drop-down menu. Depending on the selected profile, different Safety functions of the drive can be used.

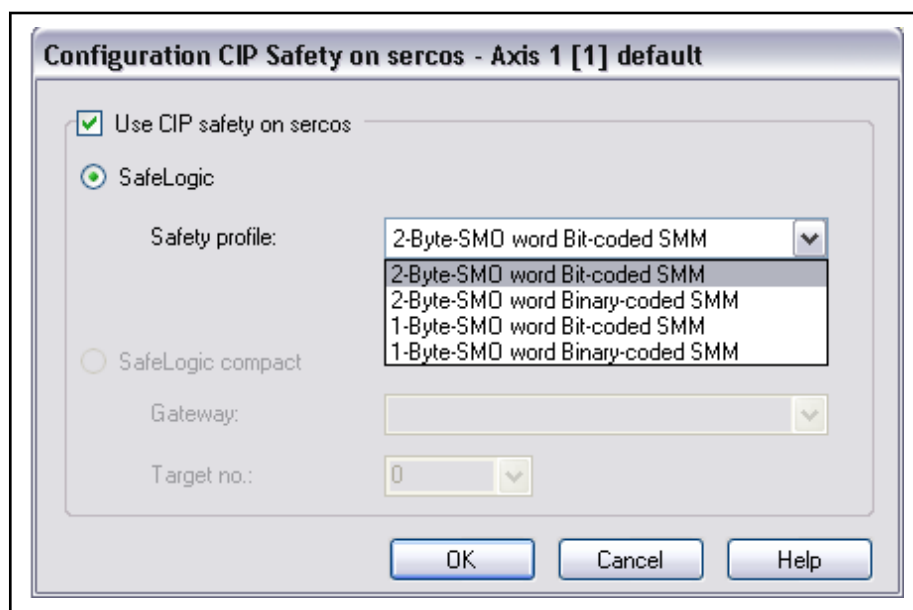


Fig. 7-236: CIP Safety on Sercos configuration dialog



The available Safety profiles are described in the documentation "Rexroth IndraDrive Integrated Safety Engineering from MPx-1x (Safe Motion)", chapter "Safe Bus Communication".

After the desired Safety profile has been selected and the configuration has been confirmed with OK, 2 elements are created in the SafetyApp under the node logic I/Os for the corresponding axis. Per axis, it is one "Control" element used to control the safety operation mode selection and one "Status" element in which the axis feedback to the corresponding state is output. In these 2 elements, the logic I/Os of the axis, different settings for CIP Safety

on Sercos and the actuation of the safety functions of the drive can be selected.

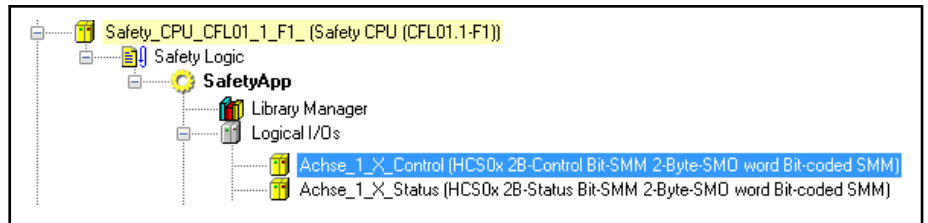


Fig. 7-237: Selecting the settings

The drive configuration in SafeLogic is described in the next chapter.

Drive configuration in SafeLogic (logic I/Os)

A window with 3 tabs opens by double-clicking on the logic I/Os of an axis. In the tab "Safe configuration", only two parameters are to be set.

Achse_1_X_Control(MTX: Safety_CPU_CFL01_1_F1_: Safety Logic: SafetyApp: Logical I/Os)				
Safe configuration I/O mapping Information				
In Work				
Name	Value	Symbolic...	Description	Unit
Vendor ID	287		Electronic key (Identity Object Attr ID 1)	
Device Type	1026		Electronic key (Identity Object Attr ID 2)	
Product Code	413		Electronic key (Identity Object Attr ID 3)	
Major Revision	13		Electronic key (Identity Object Attr ID 4)	
Minor Revision	1		Electronic key (Identity Object Attr ID 4)	
Connection name	control		Name of a connection instance	
Safety Configuration CRC (SCCRC)	0x0		Copy from target configuration	
Safety Configuration Time Stamp (SCTS)	0x0		Copy from target configuration	
Safety Network Number (SNN)	0x60000001		6 byte date/time	
Safety Device ID (SDID)	0x27		Copy from target configuration	
Format Type	2	exten_oformat	Telegram format (base format, extended format)	
Network Time Expectation (NTE)	120		worst case transmit time_m producer to consumer	ms
Expected Packet Interval (EPI)	20		Cyclic message producin_cle Time (recommended)	ms
Timeout Multiplier (TImu)	2		Number of messages tha_laring a connection error	
Ping Interval EPI Multiplier (PIEM)	19		Rate of time coordination_ssages [in EPI intervals]	
Time Coord. Msg. Min. Multiplier (TCMM)	2		Minimum transport delay_sage [in 128 µs intervals]	
Max fault number	5		Number of dropped pack_connection will be closed	
Timeout	1000		Timeout for connection s_p (forward open timeout)	ms
Device Info	SDDML-..Bit-SMM			
Creator Info	Bosch R... V1.6.0.0			

Fig. 7-238: Dialog "Safe configuration"

The parameter "Safety Network Number (SNN)" and "Safety Device ID (SDID)" are assigned once upon initial commissioning of the drive and have to be applied. Typically, all axes are in the same Safety network and thus have the same SNN. The SDID is the unique and unambiguous ID of an axis and thus differs for all axes. The parameters have to be adjusted for the logic I/O of the status and the control.

Control commissioning

Fig. 7-239: Drive commissioning dialog

After these settings have been selected, the variables with which the corresponding control or status signals are to be processed by SafeLogic have to be entered in the next tab "I/O mapping".

Functions linked to the respective control or status bit on the drive are listed in the "Description" column.

Variable	Channel	Type	Unit	Description
	Output	STRUCT		2B-Control Bit-SMM
sbMS_01_q	bit0	SAF...OOL		ModeSelectionSignal
sbES_01_q	bit1	SAF...OOL		EmergencyStopSignal
sbEC_01_q	bit2	SAF...OOL		EnablingControl
sbSMM1_01_q	bit3	SAF...OOL		SMM1Signal
sbSMM2_01_q	bit4	SAF...OOL		SMM2Signal
	bit5	SAF...OOL		SMM3Signal
	bit6	SAF...OOL		SMM4Signal
	bit7	SAF...OOL		SMM5Signal
	bit8	SAF...OOL		SMM6Signal
	bit9	SAF...OOL		SMM7Signal
	bit10	SAF...OOL		SMM8Signal
	bit11	SAF...OOL		SMM9Signal
	bit12	SAF...OOL		SMM10Signal
	bit13	SAF...OOL		SMM11Signal
	bit14	SAF...OOL		SMM12Signal
	bit15	SAF...OOL		SafeOutput_local

Fig. 7-240: Dialog "I/O mapping"

If, e.g. the variable sbES_q is connected in the SafeLogic application, the function is triggered or revoked in the drive of the EMERGENCY STOP.

Enabling the Safety Engineering in the MTX

To enable the safety engineering in the control, the dialog to configure the safety engineering has to be enabled. The dialog is enabled in the NC configuration under "Selecting optional parameters (SUP)". After the parameter "SUP/FuncGr[1]/EnablSafe" has been set to "Yes", the configuration dialog "Safety Engineering (SAFE)" under "Optional parameters" becomes visible.

In this dialog, the following parameters have to be adjusted, if required:

- Set EnablSafeTech "Enable intelligent safety engineering" (1001 00002) to "Yes" for activation
- SupprSafeTechNc "Hide safety engineering in the NC" (1001 00003)
- VelWeightFact "Scaling factor for safe velocities" (1001 00004)

Common for all drives (up to MTX14V06):

- NofSafeLimSpeed "Number of used and safely limited velocities" (1001 00005)

Adjusting the relevant SCS files

The following parameters have to be entered in the list of S-0-1050.1.6 of the SCS file:

- P-0-3231 (SMO: Operating state)
- P-0-3262 (SMO: Selected operating state)

Example:

S-0-1050.1.6 = (135,51,189,144,P-0-3231,P-0-3262) ;



By processing the control and status signals for safety functions in SafeLogic, the corresponding NC PLC interface signals of axes or spindles are not taken into consideration anymore.

These interface signals cannot be used anymore in connection with SafeLogic.

"NC Ready" bit (P-0-3261.0.1 bit 0)

Since axes/spindles have to be moved within a wide velocity/speed range, the transfer times have thus to be set sufficiently high. To reduce the transfer time in case of low velocities/speeds, bit 0 of parameter P-0-3261.0.1 can be used.

If this bit is set, the drive immediately acknowledges the safe state.

The "NC Ready" bits are set from the PLC application. Thus, the interface signal **"qAx_SafRedTransTime"** or **"qSp_SafRedTransTime"** are stored on the axis/spindle interface. It is reported to the drive that the adaptation to the command value system is completed.



Via the "NC-Ready" signal in "P-0-3261.0.1, SMO: State machine control word, functional", the control unit may signal the end of the transition process of safety technology. The SMD safety function will then immediately terminate the transition and not wait for the transition time to expire (P-0-3280.0.2 or P-0-3280.0.3 or P-0-3280.0.4).

The "NC-Ready" signal is evaluated dynamically, i.e. in an edge-triggered way. The safety technology detects a valid edge (0-1 transition), if the signal previously had been "0" for at least 10 ms.

If the transition process is terminated ahead of time via the "NC-Ready" signal, although the target velocity of the selected safety function has not yet been reached, the error "F7052 Selected target velocity exceeded" is generated and the drive is decelerated.

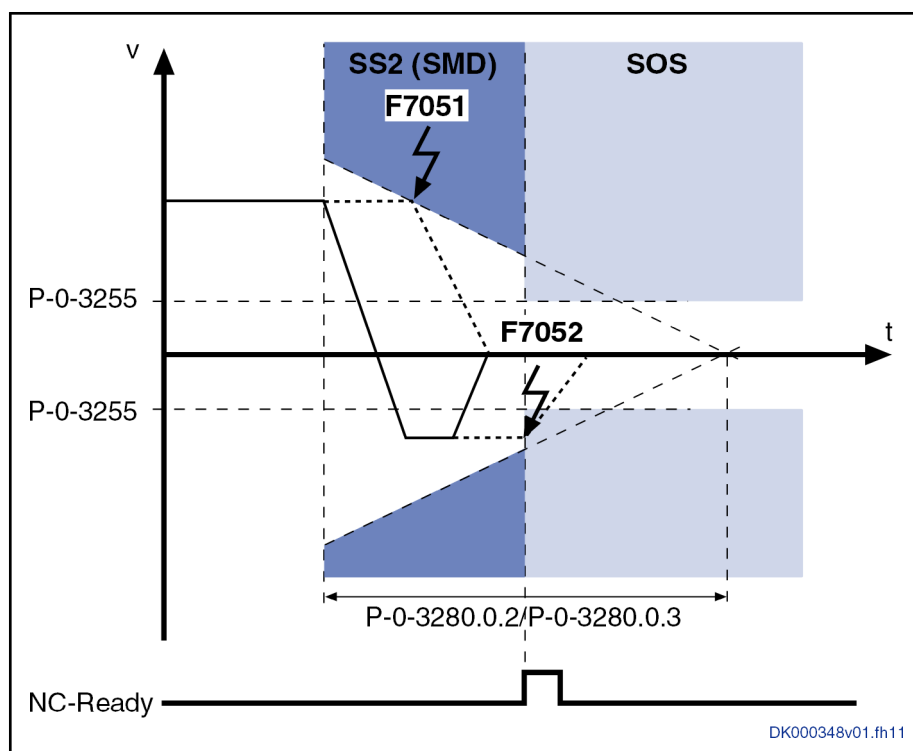


Fig. 7-241: "NC ready" bit

Also see "Rexroth IndraDrive Integrated Safety Engineering from MPx-1x (Safe Motion)", chapter "Transfer to safe state".

Using the "NC ready" bits

1. To transmit the bit from the control to the drive, it has to be configured in the SCS file in the signal status word (S-0-0027/S-0-0329).
 $S-0-0027 = (....., P-0-3261.0.1, ...)$
 $S-0-0329 = (....., 0, ...)$
2. To set the bit, either the interface signal "qAx_SafRedTransTime" or the interface signal "qSp_SafRedTransTime" is used in the PLC.

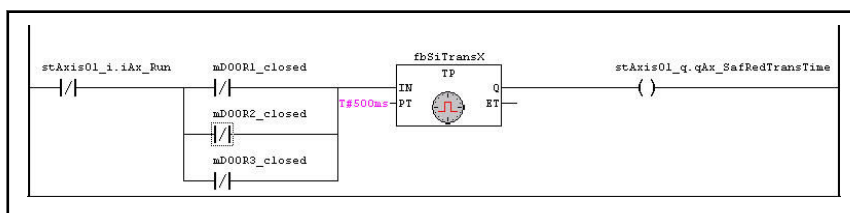


Fig. 7-242: PLC program

⚠ DANGER**Risk of injury**

An unintended use of the bit can lead to hazardous situations.

Example:

A high-speed spindle is to be switched to "Safe Stop 1 (EMERGENCY Stop)" (SS1 ES) and the bit is set while the spindle still rotates at a dangerous velocity.

Consequence:

The drive immediately acknowledges the safe state. The safety engineering immediately identifies the error, switches off the power and the spindle spins. It is now possible that the protecting door is already unlocked even though the spindle is still dangerous.

Commissioning the Safety Engineering on a drive



Commissioning the Safety Engineering is described in the documentation "Rexroth IndraDrive Integrated Safety Engineering from MPx-1x (Safe Motion)", chapter "Initial commissioning".

8 VAM simulator

8.1 Using the integrated VAM for touch devices

The touch device templates also include a touch VAM. The touch VAM is configured in the provided example project and can be used directly for the basic project. Apply the templates to your projects using the export/import function in IndraWorks Engineering.

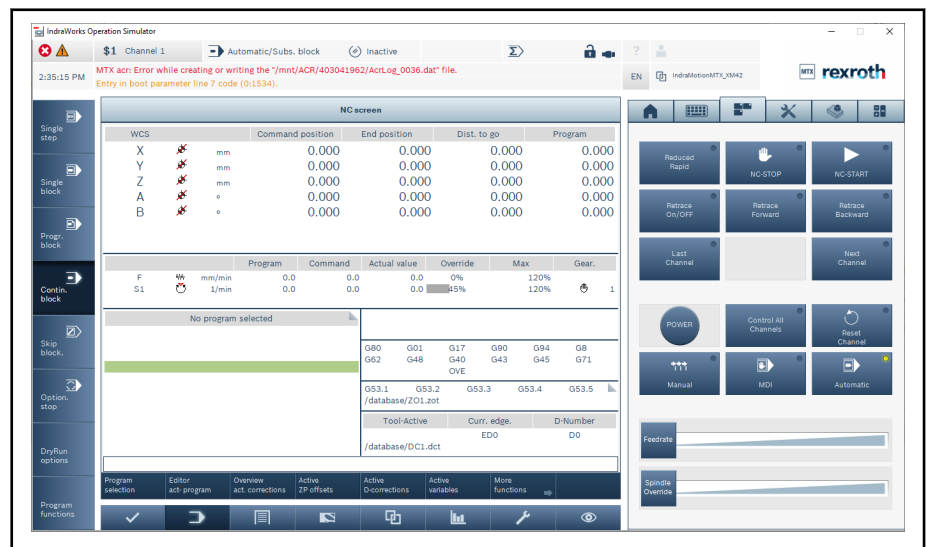


Fig. 8-1: VDP15 integrated VAM example project "Basic project"



To use the template, your program requires the respective PLC variables.

For more information, refer to the documentation "MTX 15VRS Multitouch" (DOK-MTX***-MULTI***V15-APRS-EN-P).

8.2 General description

The VAM simulator replaces the real VAM 40 and VAM 41 when working with the MTX Simulator. The appearance and functions replicate the real VAMs. In the current version, the texts are available in English and German.

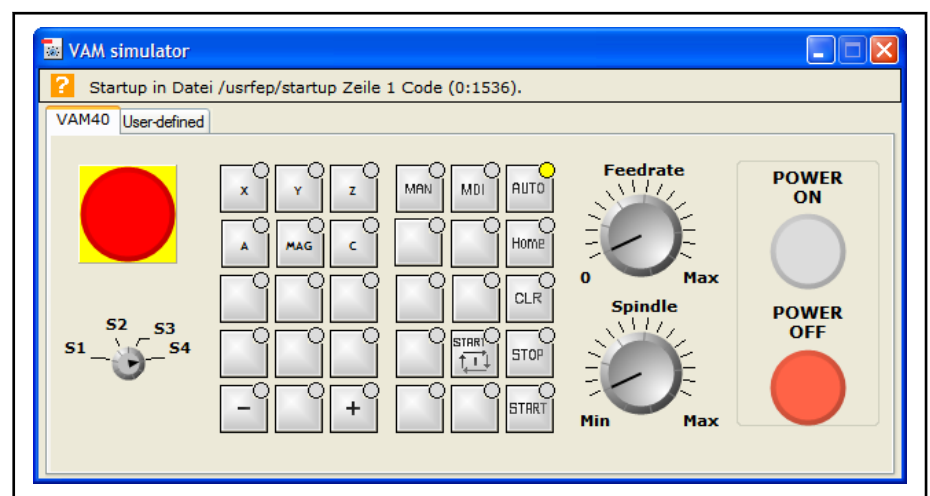


Fig. 8-2: VAM simulator VAM 40

8.3 Configurator

The VAM simulator is configured in IndraWorks Engineering.

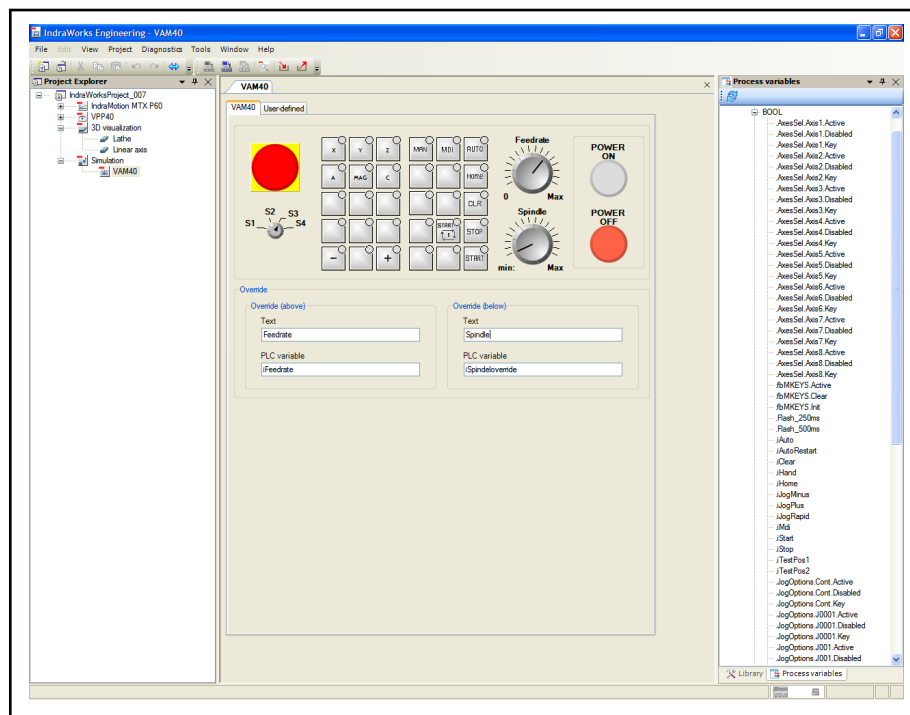


Fig. 8-3: Configuration in IndraWorks

Project Node "Simulation" with subnode "VAM40" or "VAM41"

Double-click on the project node of a virtual user panel to call the configuration screens. The top half of the screen shows the image of the virtual user panel while the configuration screen of the element selected in the image is shown in the bottom half.

Configuration screen of the E-Stop

Only the PLC variable for a later communication can be created on the E-Stop screen.

Configuration screen of the key-operated switch

The individual switch settings are labeled and the PLC variables are assigned on the configuration screen of the key-operated switch.

Override: Configuration side

The configuration screen of the override is used to label and assign the PLC variables.

Configuration page of the keypads

This is used to assign the PLC variables and to label the keypads with texts or prepared images. Each individual key can be assigned by entering a text or by dragging and dropping an image from the symbol list.



Depending on the selected type (VAM 40 or VAM 41), the following different functions are available.

Configuration screen of the quick-stop module

The configuration screen of the quick-stop module can be used to make various settings for the two keys:

- Key labeling
- Function

Since the keys of the real VAM 40 are wired via the hardware and the switches are equipped with NC and N.O. contact, these settings can also be selected here.

- Assigning PLC variables

The configuration screen for the freely configurable elements

In the configuration screen of the eight freely configurable elements, the following ones can be selected:

- LED
- Button with LED
- Button without LED
- Switch with two positions
- Switch with three positions

The center position has no function.

These elements can be labeled and assigned with PLC variables.



Since the keys of the real VAM 41 are wired via the hardware and the switches are equipped with NC and N.O. contact, these settings can also be selected here.

8.4 The application

The application is separate; it is used to control the PLC program in the MTX Emulation. This application can be started independently of IndraWorks Engineering or IndraWorks Operation. When the VAM simulator is started, the configuration is read from the currently active project.



The VAM simulator communicates directly with the Emulation. IndraWorks Operation or IndraWorks Engineering do not have to be started.

9 Upgrading the CMLx5 control to the XM42 control

9.1 Differences between the CMLx5 control to the XM42 control

There are fundamental differences between the XM42 control to the CMLx5 control:

- No Profibus on the control, only available via extension module
- No Inline I/Os
- No ready contact

9.2 Requirements for conversion

1. An executable IndraWorks suite installation, including support for XM42 controls is available (from version 15V02). The existing project can be loaded and is available as archive. In the project, a control data backup (TAR file) is available.
2. An executable IndraWorks suite installation, including support for CMLx controls is available. In the latest version 14V22 patch x , the project to be converted can be loaded and executed.

9.3 Analysis of the available hardware

The used hardware components have to be taken into consideration in the project. Part of the connected devices has to be replaced by other hardware components.

There are different solutions:

- An Inline module can be replaced by a corresponding S20 module. Optionally, additional cams can be used on an existing Profibus bus coupler, Profinet bus coupler or Sercos bus coupler.
- If "fast action" inputs are used, an S20 module is required.
- If necessary, the machine can be operated using machine control keys at the multitouch HMI device. A (mechanical) IndraControl VAM machine control panel is not necessarily required for this purpose. For more information or support, refer to the basic MTX control project.

9.4 General mode of operation

1. Start the 14V22 patch x version and load the project to be converted.
2. Convert the project to the current version 14 (control and operating devices).
3. Create a project archive.



Take into consideration that all optional components are also archived when creating the archive!

Upgrading the CMLx5 control to the XM42 control

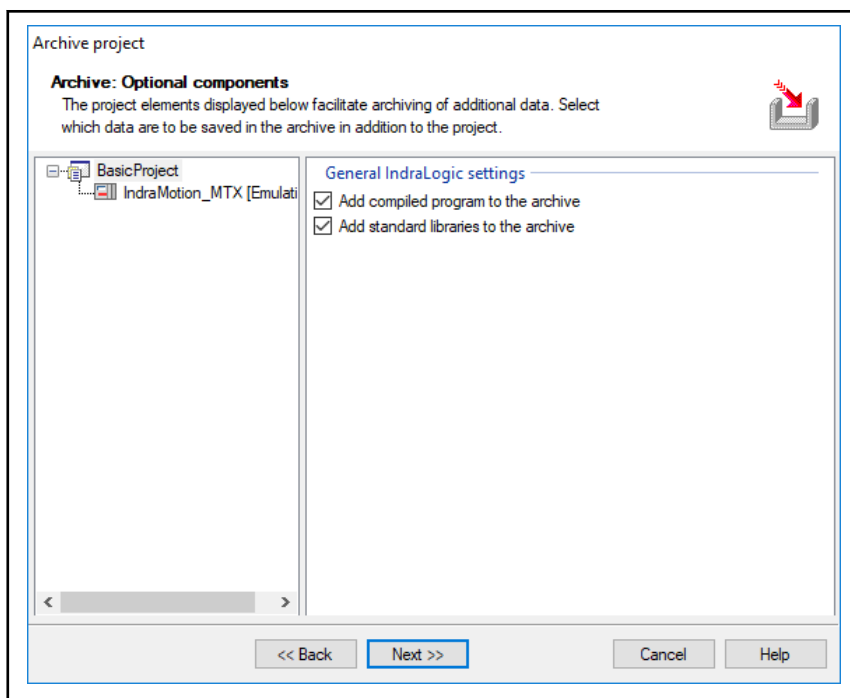


Fig. 9-1: Project archiving

4. Select the control node and export the complete node

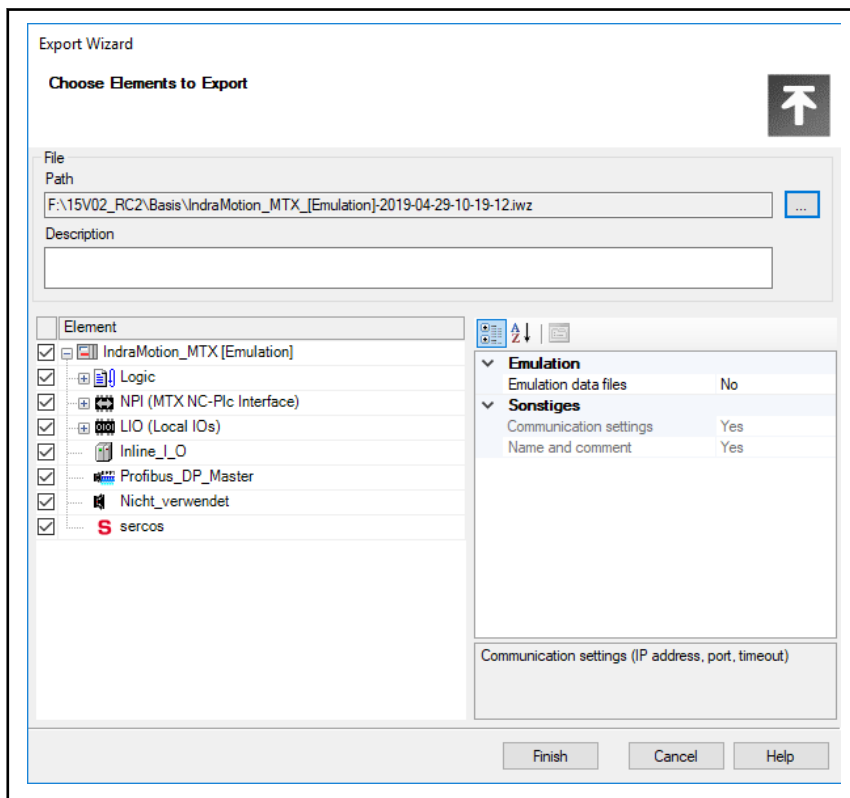


Fig. 9-2: Exporting the control node

5. Save the PLC program name or any information regarding the PLC address, module identifier, etc. The information is required in a later step.
6. Start the new IndraWorks suite 15Vxx

7. Open the project or restore the IndraWorks project archive using the function **Project ► Restore**.

The following dialog is displayed.

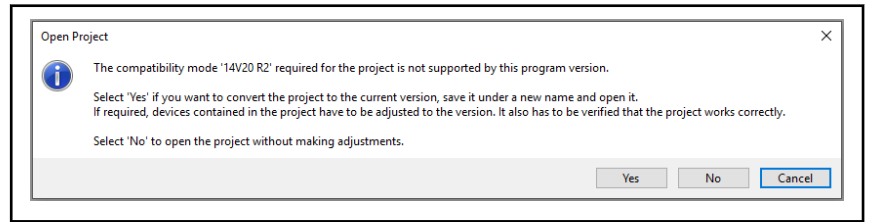


Fig. 9-3: "Open project" dialog

Confirm the dialog with "Yes" and execute the following steps:

1. Enter a new IndraWorks project name
2. Set the PLC compiler version

When confirming the dialog with "No", execute the following steps:

1. Manually set the IndraWorks compatibility mode to the latest version
 2. Manually set the PLC compiler mode
8. Add an additional XM42 control to the existing project.

There is only one XM42 control. The control type (Standard, Performance or Advanced) is specified by internal limits in the XM42 control hardware.

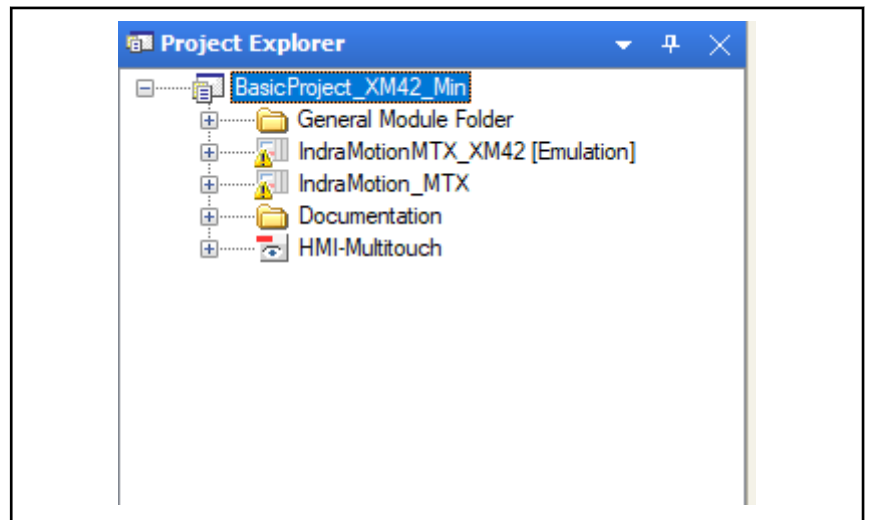


Fig. 9-4: Project Explorer after inserting the new XM42 control
Additionally, there is an XM42 control in the project.

9. Check the configuration settings of the XM42 control

Upgrading the CMLx5 control to the XM42 control

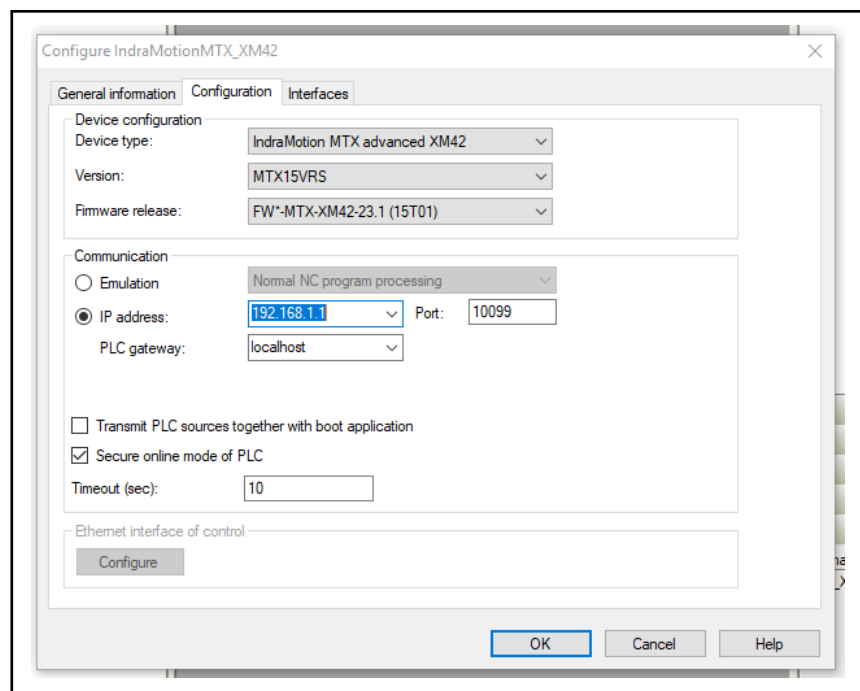


Fig. 9-5: XM42 control dialog properties

The latest setting should be entered for version and firmware release.

Enter the XM42 control address as IP address (default address 192.168.1.1).

Close the dialog.

10. Execute a connection test to the control

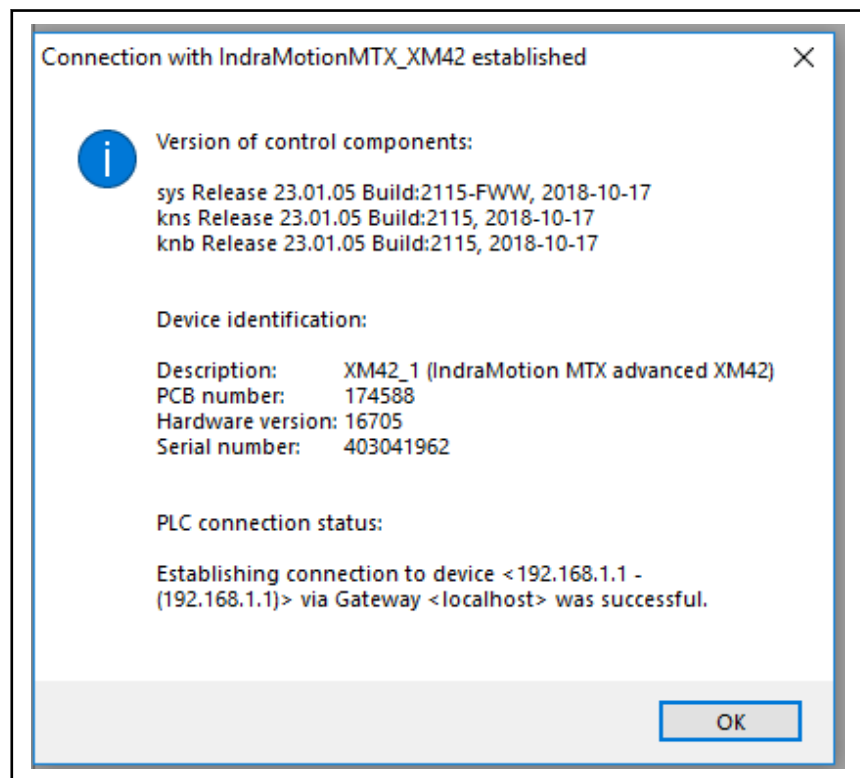


Fig. 9-6: XM42 connection test

Reasons for connection test failure:

- The IP address of the control is incorrect. Change the IP address in the settings or if you do not know the IP address, reset the IP address to default (see MTX Commissioning 15VRS)
- The IP address of your network interface in the PC is incorrect. If a CML control is connected to the interface, the IP address of the interface was 192.168.142.250. As the XM42 control has a different IP address, change the interface (e.g. to 192.168.1.200)

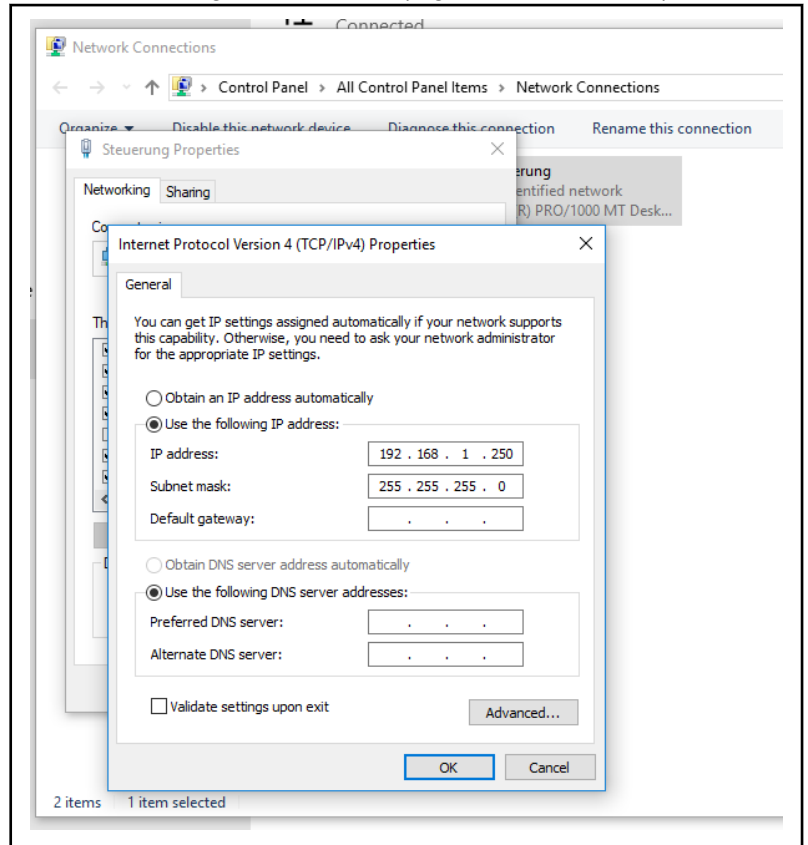


Fig. 9-7: TCP/IP4 network settings of the PC

11. The easiest way to change a CML control to a XM42 control is by using the export function and the import function in the Project Explorer. In the next steps, the use of Profibus or Inline I/Os is explained.
12. Check the PLC program name and if required, adjust it (also see step 5)
13. Select the control node of the XM42 control. Import the previously exported CML control

Upgrading the CMLx5 control to the XM42 control

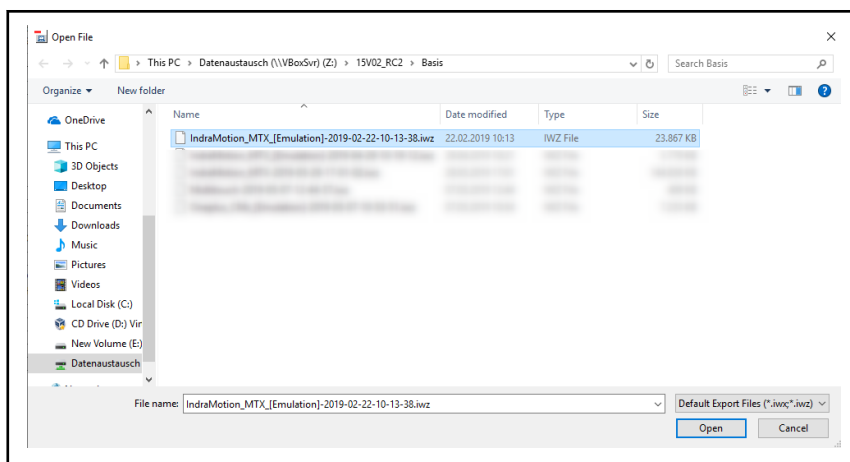


Fig. 9-8: Import of control data



Not all elements can be imported (e.g. Profibus, Inline I/O).

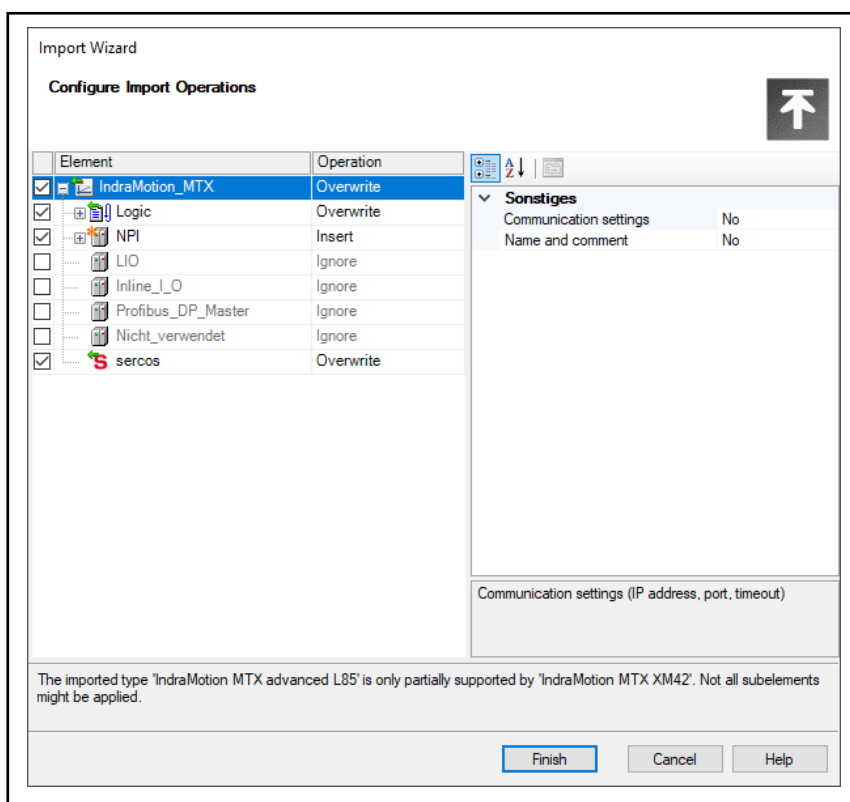


Fig. 9-9: Import wizard

14. If a Profibus module is used in the project, add a Profibus module (XFE01.1FB-10) to the project (Profibus master).

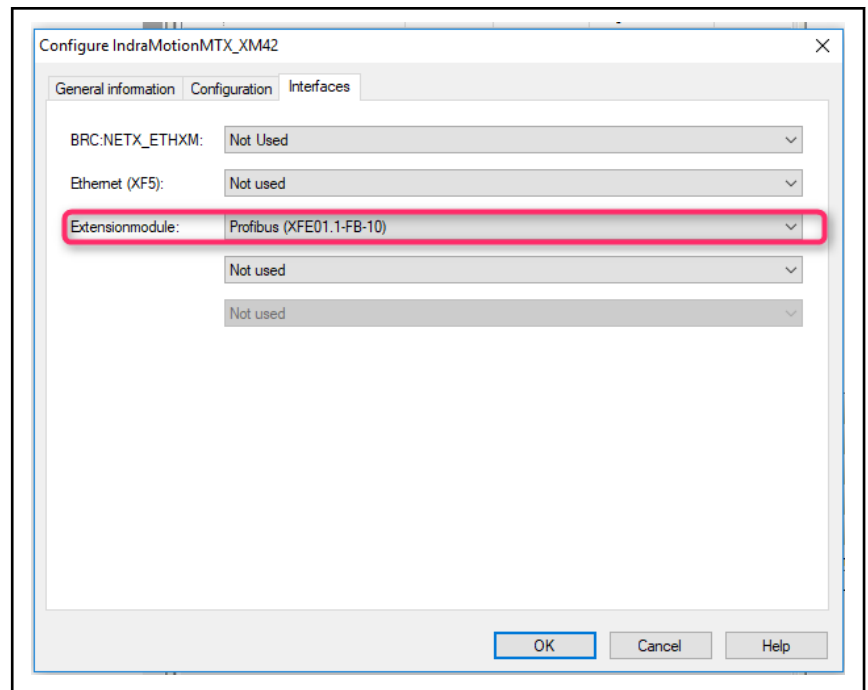


Fig. 9-10: Add an XM42 Profibus interface

Subsequently set the "Profibus DP master XM" in the Profibus module in the Project Explorer.

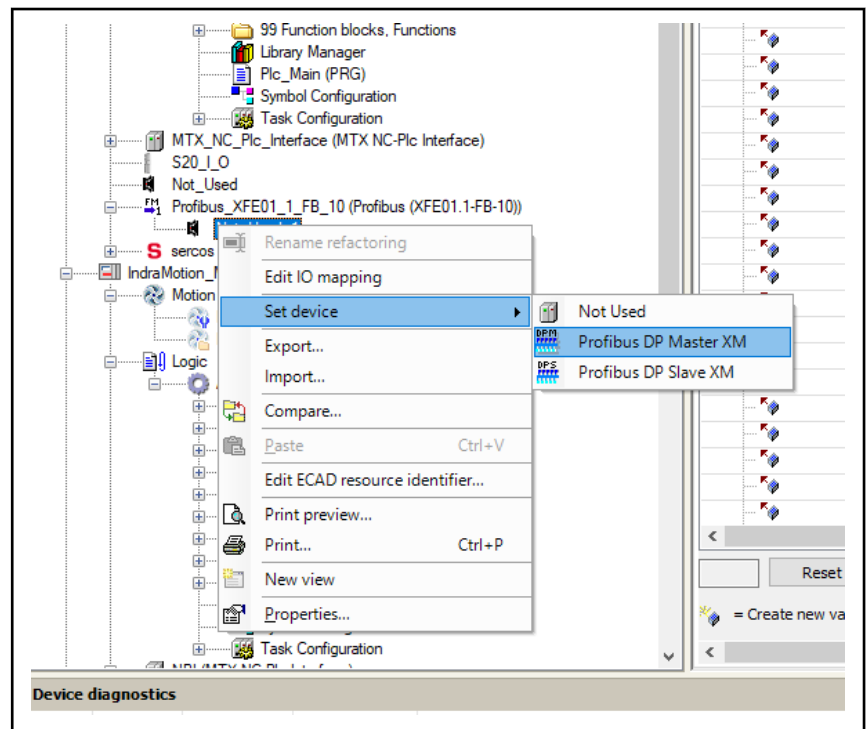


Fig. 9-11: Set the Profibus master

Subsequently, import the exported file from the CML project via **XM42 ► Profibus DP master XM** again.

The import reports an error.

Upgrading the CMLx5 control to the XM42 control

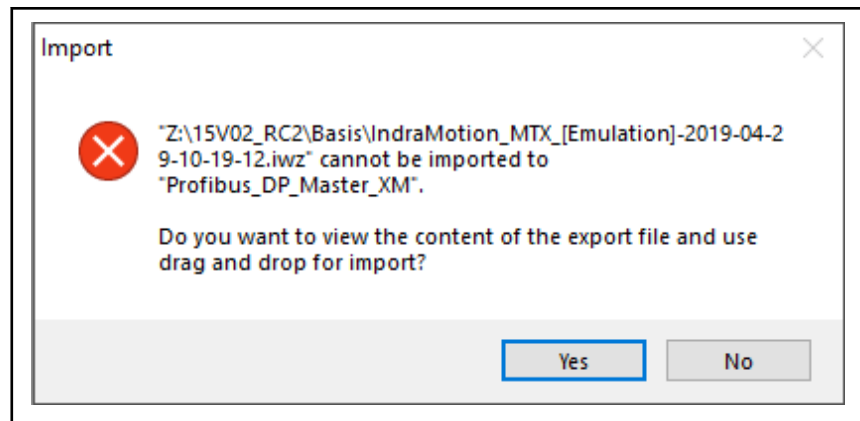


Fig. 9-12: "Import" dialog

Select the "Yes" button to start the import by using the "Drag and Drop" function.

15. Select the Profibus node in the export data view and copy the node to the Profibus_DB_Master node in the XM42 control. Subsequently, all Profibus devices should be available.

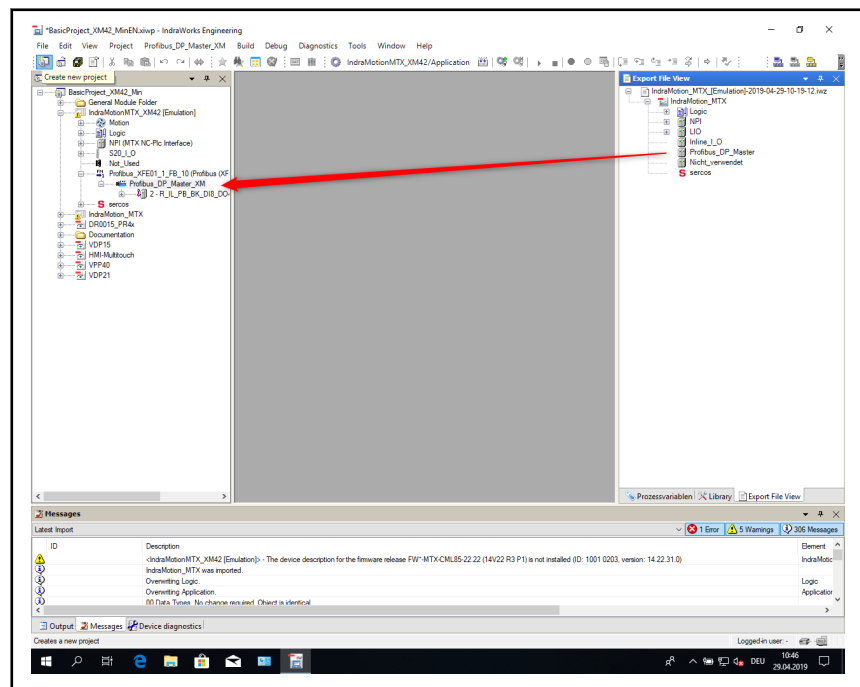


Fig. 9-13: Export data view - Moving the Profibus devices

16. If Profinet devices are available in the project, first select the Profinet node and subsequently copy the Profinet elements from the export data view
17. If local I/Os are used in the project, the only option is to replace these local I/Os by another hardware module.

Options:

- A Profibus bus coupler with the relevant modules can be used
- An S20 I/O module can be used
- A Profinet bus coupler with the relevant modules can be used

The data within the modules have to be entered.

Suggestion for Profibus local I/Os:

- R-IL PB DI8 DO4 - Profibus bus coupler
- R-IB IL 24 DI 8 - 8 digital inputs
- R-IB IL 24 DO 8 - 8 digital outputs

Suggestion for S20 local I/Os:

- S20 DIDO-8

Advantage of the Profibus bus coupler: the available Inline I/Os can be used as cams on the Profibus coupler. If no Profibus has previously been used in the project, S20 I/O modules can be used. These modules can be directly connected to the XM42 control.



If FastAction I/Os or FlyCut I/Os are used, this can only be realized with S20 I/O modules.



FastAction I/Os can only be used byte-by-byte. In the CML control, bit-by-bit use was still possible!

18. Check the I/O modules in the XM42 control

Permanently entered I/O addresses also have to be entered in the XM42 control.

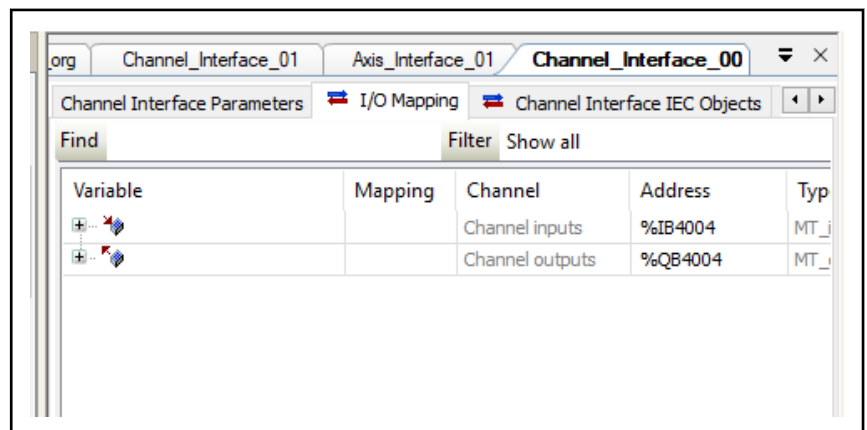


Fig. 9-14: Check the I/O addresses



Set the addresses of the individual components to the same address as in the CML project. Due to the position change in the project, address conflicts can occur!

19. Set the active PLC application

The CMLx control application is set as active control in your project. This application has been transmitted during a PLC download. The XM42 application has to be set as active application.

Select the node **XM42 ► Logic ► Application**. Right-click to open the popup menu. Select "Set active application".

Upgrading the CMLx5 control to the XM42 control

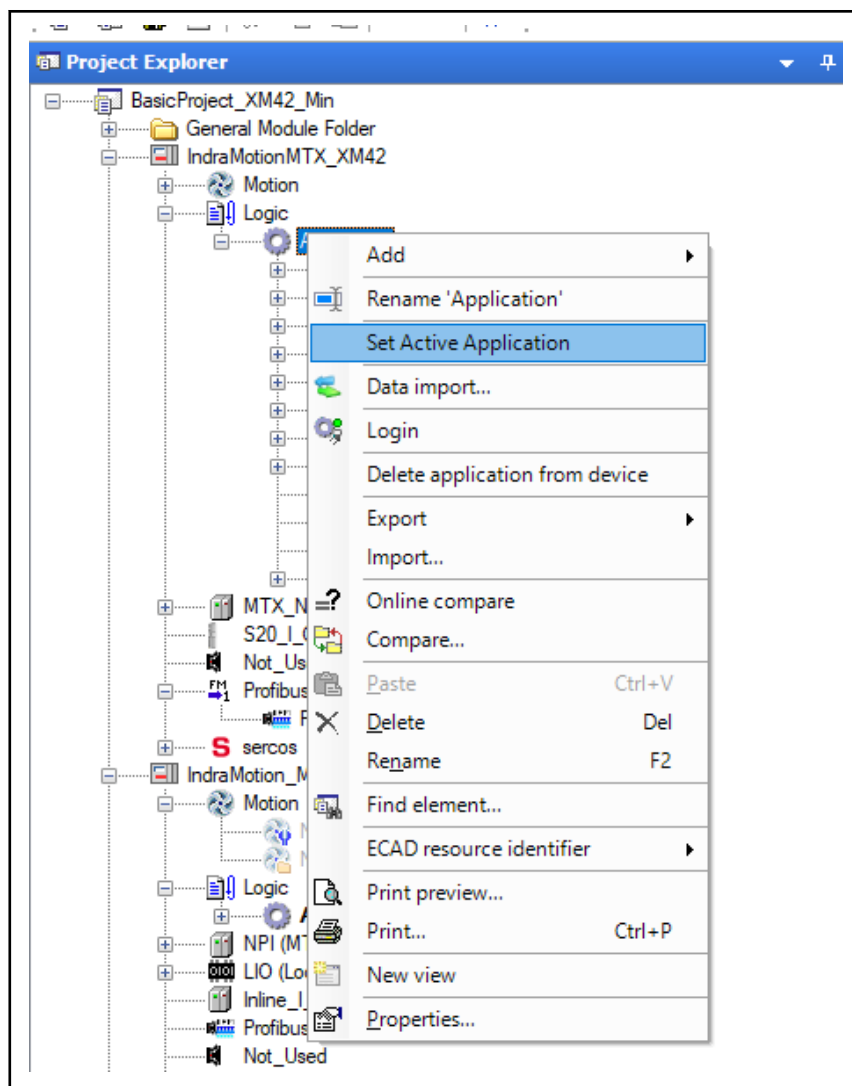


Fig. 9-15: Set the active PLC application

20. Transmit NC data

Switch the XM42 control online. Subsequently, transmit the last saved TAR file of the CML control to the XM42 control via the "Restore" function. The TAR file can be found in the node of the XM42 control. The TAR archives are stored here during the import.

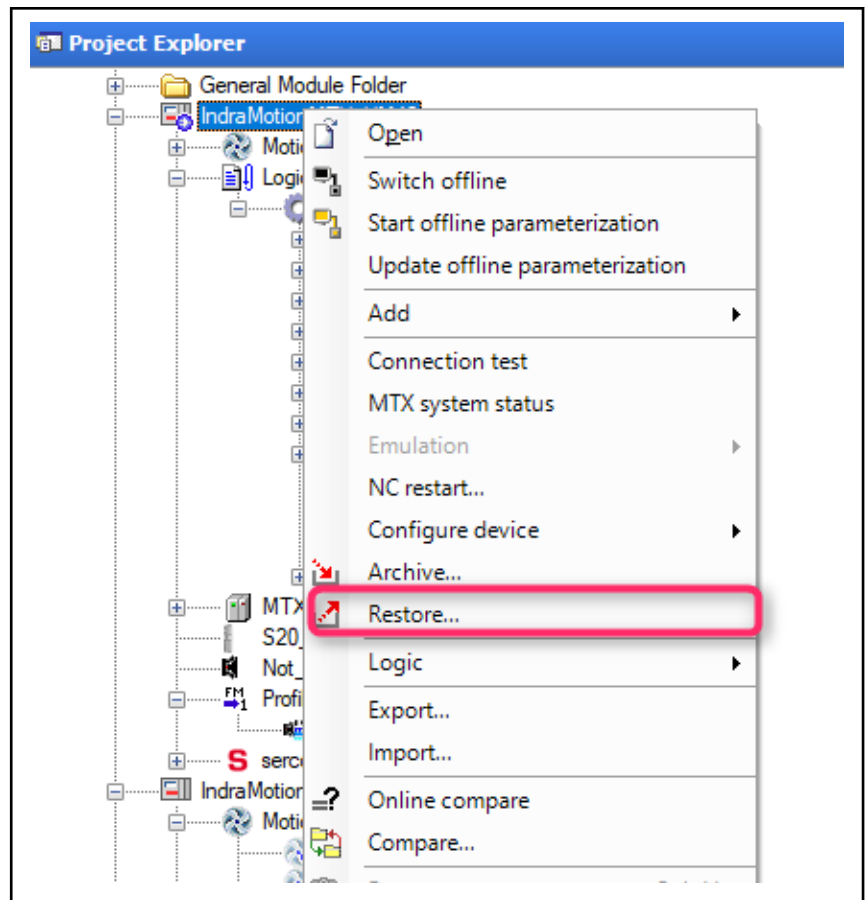


Fig. 9-16: Restoring the NC data

The NC data is now contained in the control.

21. Transmitting the PLC project

If all I/Os have been applied to the project, the PLC program can be compiled and transmitted to the XM42 control. Subsequently, start the PLC program.

Now, the PLC data is active.



Occasionally, error messages are reported during the PLC download.

Possible causes:

- Channel or axes names have not been applied correctly
- User libraries are outdated
- PLC application name has been changed
- etc.



Check the PLC library versions and if required, update them to a new version.

22. Adjusting the visualization device

Open the "Properties" dialog of the used visualization device. Update the device version to the latest available version (e.g. 15V02).

Upgrading the CMLx5 control to the XM42 control

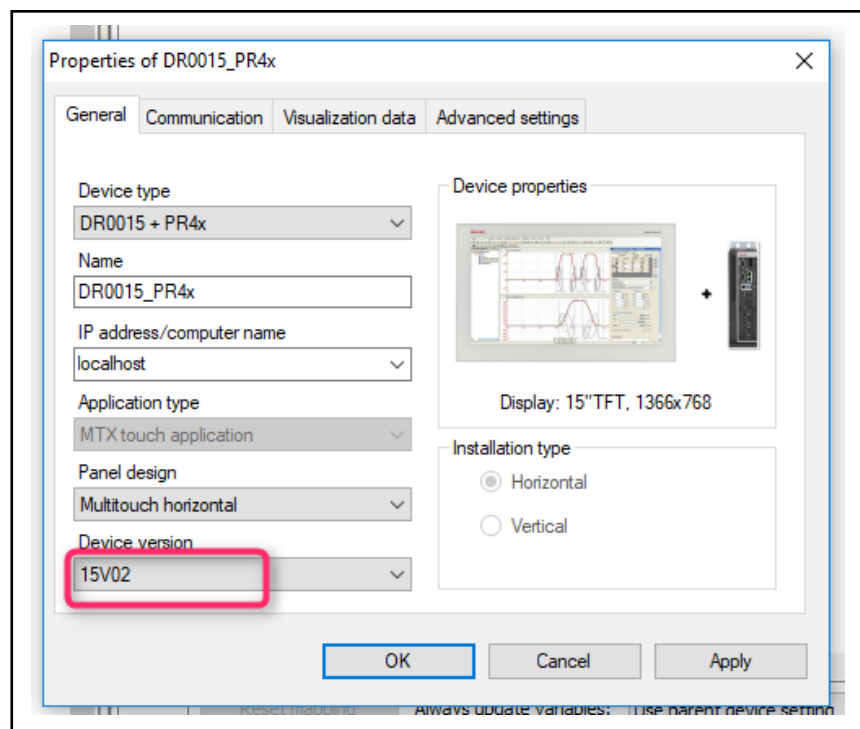


Fig. 9-17: Update the visualization device

Change to the "Communication" tab. Change the CMLx control to your XM42 control in the "Control" field. With other fields, proceed identically and select the relevant variables from your XM42 PLC program.

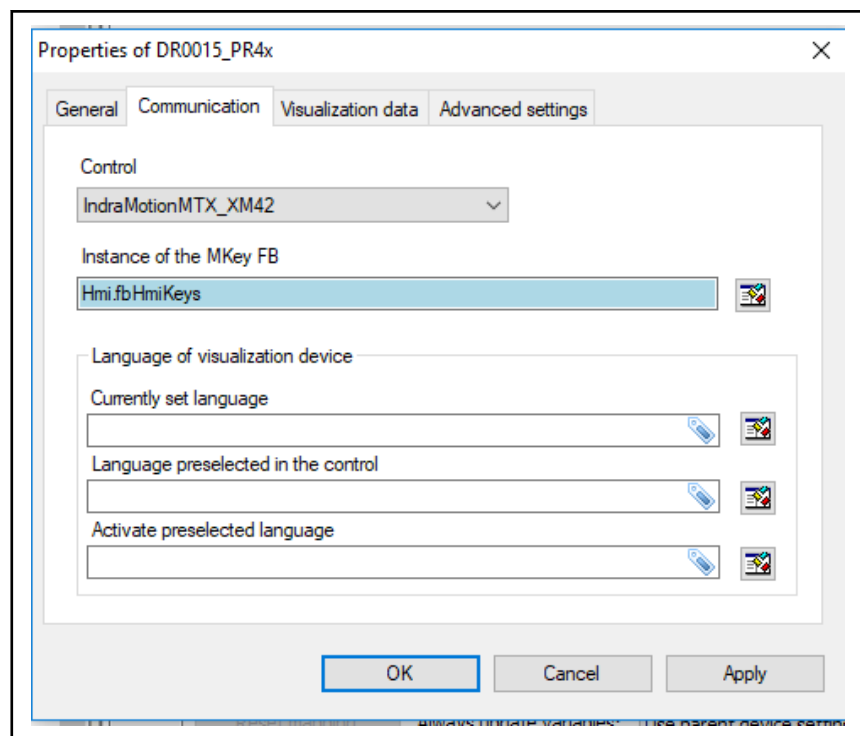


Fig. 9-18: Adjust the communication settings

Close the dialog.

Validate the visualization by using a right-mouse click on the hardware and the menu entry **Validate**.

Transmit and enable the visualization data.

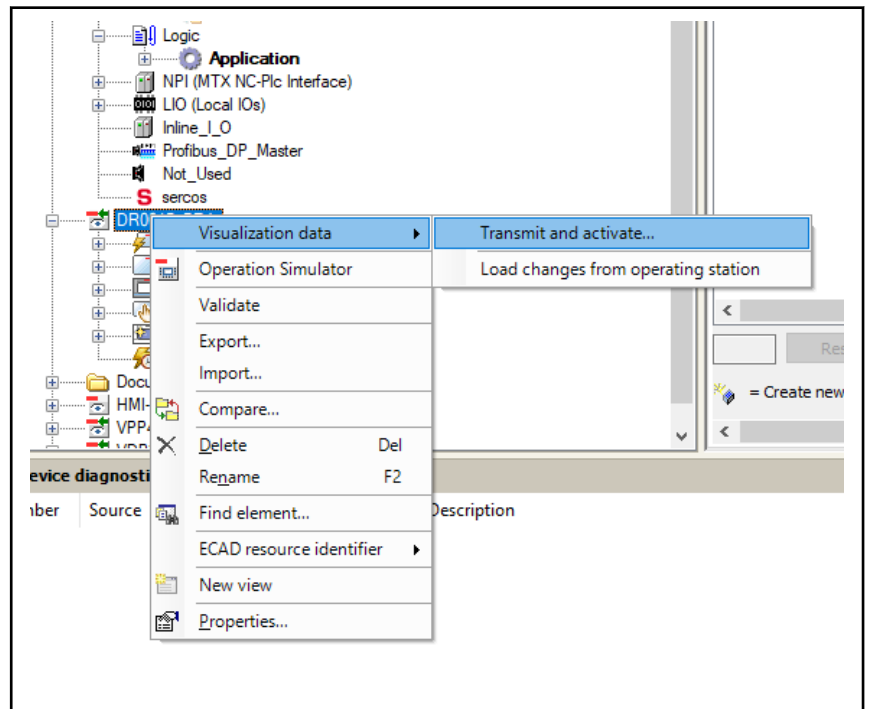


Fig. 9-19: Transmit the visualization data

23. Delete the old CML control node
24. Start the Operation desktop

9.5 Ready contact

As the controls of the XM family are not equipped with a ready contact, an available wiring has to bridge the contact.

10 Appendix

10.1 Basic data type collection

SimpleType	ann:	BaseType	Restrictions					
			Facets				Patterns	Enum.
			MinInc	MaxInc	Min. Length	Max. Length		
LifeUnit_t		Iso Latin 1 String				3		min cyc
ToolOri_t	Tool Orientation (new since 14V16)	xs:int						0 17 18 19

Tab. 10-1: Basic data types of the tool data type collection (tool_ty.xsd und basic_tool_ty.xsd)

SimpleType	ann:	BaseType	Restrictions					
			Facets				Patterns	Enum.
			MinInc	MaxInc	Min. Length	Max. Length		
isoLatin1String	Iso Latin 1 String	xs:string						
isoStr31_t	Iso Latin 1 String 0 - 31	isoLatin1String			0	31		
isoStr16_t	Iso Latin 1 String 0 - 16	isoLatin1String			0	16		
isoStr99_t	Iso Latin 1 String 0 - 99	isoLatin1String			0	99		
isoStr512_t	Iso Latin 1 String 0 - 512	isoLatin1String			0	512		
Str1_t	Type String 0 - 1	isoLatin1String			0	1		
Str2_t	Type String 0 - 2	isoLatin1String			0	2		
Str3_t	String 0 - 3	isoLatin1String			0	3		
Str6_t	String 0 - 6	isoLatin1String			0	6		
Str8_t	String 0 - 8	isoLatin1String			0	8		
Str8_NoUmlaut_t	String 0 - 8	isoLatin1String			0	8	[^äöüß]*	
Str16_t	String 0 - 16	isoLatin1String			0	16		
Str32_t	String 0 - 32	isoLatin1String			0	32		
Str80_t	String 0 - 80	isoLatin1String			0	80		
Str240_t	Type String 0 - 240	isoLatin1String			0	240		
Str240_NoUmlaut_t	Type String 0-240 without äöüß	isoLatin1String			0	240	[^äöüß]*	
Byte_t	Type Byte	xs:byte						
UnsignedByte_t	Type Unsigned Byte	xs:unsigned-Byte						

Appendix

SimpleType	ann:	BaseType	Restrictions					
			Facets				Patterns	Enum.
			MinInc	MaxInc	Min. Length	Max. Length		
Byte0_1_t	Byte 0 or 1	xs:byte	0	1				0 1
Byte0_2_t	Byte 0 - 2	xs:byte	0	2				0 1 2
Byte0_3_t	Byte 0 - 3	xs:byte	0	3				0 1 2 3
Byte0_4_t	Byte 0 - 4	xs:int	0	4				0 1 2 3
Byte0_5_t	Byte 0 - 5	xs:byte	0	5				0 1 2 3 4
Byte0_6_t	Byte 0 - 6	xs:byte	0	6				0 1 2 3 4 5 6
Byte0_7_t	Byte 0 - 7	xs:byte	0	7				0 1 2 3 4 5 6 7

SimpleType	ann:	BaseType	Restrictions					
			Facets				Patterns	Enum.
			MinInc	MaxInc	Min. Length	Max. Length		
Byte0_8_t	Byte 0 - 8	xs:byte	0	8				0 1 2 3 4 5 6 7 8
Byte0_32_t	Byte 0 - 32	xs:byte	0	32				
Byte1_2_t	Byte 1 or 2	xs:byte						1 2
Byte1_3_t	Byte 1 - 3	xs:byte	1	3				1 2 3
Byte1_4_t	Byte 1 - 4	xs:byte	1	4				1 2 3 4
Byte1_8_t	Byte 1 - 8	xs:byte	1	8				1 2 3 4 5 6 7 8

Appendix

SimpleType	ann:	BaseType	Restrictions					
			Facets				Patterns	Enum.
			MinInc	MaxInc	Min. Length	Max. Length		
Byte1_12_t	Byte 1 - 12	xs:byte						1 2 3 4 5 6 7 8 9 10 11 12
Byte-1_7_t	to be omitted	xs:byte	-1	7				
Byte0_10_t	Byte 0 - 10	xs:byte	1	10				0
Byte0_99_t	Byte 0 - 99	xs:byte	0	99				
Byte1_99_t	Byte 1 - 99	xs:byte	1	99				
Byte0_100_t	Byte 0 - 100	xs:byte	0	100				
Short_t	Type Short	xs:short						
UnsignedShort_t	Type Unsigned Short	xs:unsigned-Short						
Int_t	Type Int	xs:int						
UnsignedInt_t	Type Unsigned Int	xs:unsignedInt						
Int0_999_t	To be omitted	xs:int	0	999				
Int0_9999_t	To be omitted	xs:int	0	9999				
Int0_1T_t	Int 0 - 1000	xs:unsignedInt	0	1000				
Int0_10T_t	Int 0 - 10000	xs:int	0	10000				
Int0_100T_t	Int 0 - 100000	xs:int	0	100000				
Int0_32767_t	To be omitted	xs:int	0	32767				
Int0_65535_t	Int 0 - 65535	xs:int	0	65535				
Int0_1M_t	Type Int 0 - 1000000	xs:int	0	1000000				
Int_goet_0_t	Type Int minIncl 0	xs:int	0					
Float_t	Type Float	xs:float						
Float-10_10_t	Type Float -10 - 10	xs:float	-10	10				
Float0_100_t	Type Float 0 - 100	xs:float	0	100				

SimpleType	ann:	BaseType	Restrictions					
			Facets				Patterns	Enum.
			MinInc	MaxInc	Min. Length	Max. Length		
Float-100_100_t	Type Float -100 - 100	xs:float	-100	100				
Float-10M_10M_t	Type Float -10 Mio - +10 Mio	xs:float	-10000000	10000000				
Double_t	Type Double	xs:double						
Double0_10_t	Type Double 0 - 10	xs:double	0	10				
Double0_100_t	Type Double 0 - 100	xs:double	0	100				
Double0.01_100_t	Type Double 0.01_100	xs:double	0.01	100				
Double0_1T_t	Type Double 0 - 1000	xs:double	0	1000				
Double0_180_t	Type Double 0 - 180	xs:double	0	180				
Double0_360_t	Type Double 0 - 360	xs:double	0	360				
Double-180_180_t	Type Double -180 - 180	xs:double	-180	180				
Double0_10T_t	Type Double 0 -10000	xs:double	0	10000				
Double0_100T_t	Double 0 - 100000	xs:double	0	100000				
Double1_10M_t	Type Double 1 - 10 Mio	xs:double	1	10000000				
Double-10M_10M_t	Type Double -10 Mio - +10 Mio	xs:double	-10000000	10000000				
Double_goet_0_t	Type Double minIncl 0	xs:double	0					
Boolean_t	Type Boolean	xs:boolean						
Ovr_t	Type Override 0 - 150	xs:float	0	150				
Pos_t	Type Position (-1 Mio - +1 Mio)	xs:double	-1000000	1000000				
Dist_t	Type Position (0 - 1 Mio)	xs:double	0	1000000				
Vel_t	Type Velocity 0 - 1000 Mio [Axis Velocity]	xs:double	0	100000000				
SpSpeed_t	Spindle Speed 0 - 100000	xs:double	0	100000				
Acc_t	Acceleration 0 - 1000 [Axis Acceleration]	xs:double	0	1000				
SpAcc_t	Spindle Acceleration 0 - 100000	xs:double	0	100000				
JumpVel_t	Jump Velocity 0 - 100 000	xs:double	0	100000				
JumpAcc_t	Jump Acceleration 0 - 200	xs:float	0	200				
Torq_t	Torque 0 - 1000 [%]	xs:float	0	1000				
Sp_t	Spindle 0 - 8	xs:byte	0	8				

Appendix

SimpleType	ann:	BaseType	Restrictions					
			Facets				Patterns	Enum.
			MinInc	MaxInc	Min. Length	Max. Length		
ChSp_t	Spindle 0 - 8	xs:byte	0	32				
ChAx_t	Channel Axis (0 - 8)	xs:unsigned-Byte	0	8				
Ax_t	Lin. and rot. Axis (0 - 99)	xs:byte	0	99				
Dr_t	NC-controlled Axis Drives (0 - 99)	xs:byte	0	99				
Prec_t	Type Precision (0 .. 7)	xs:byte	0	7				
Ch_t	Type Channels (0 ..60)	xs:byte	0	60				
CS_t	Type Coordinate System	isoLatin1String				3		WCS MCS BCS
AxFun_t	Type Axis Functionality (later limit it to 0, 1, 2, 3 (Designation Main Axis X, Y, Z))	xs:int						
SpGr_t	Type Spindle Group 0 - 4	xs:byte	0	4				
DigBuff_t	Type Digitizing Buffers 20 - 2000	xs:unsignedInt	20	2000				
ResDigBuff_t	Type Reserved Dig. Buffer 5 - 100	xs:unsignedInt	5	100				
Blk_t	Block 3 - 999	xs:int	3	999				
CpuTimeBl-Prep_t	Cpu Time for Block Preparation 50 - 100	xs:byte	50	100				
OpFiles_t	Open Files 5 - 60	xs:int	5	60				
BuffNfs_t	NFS Buffer 2048 - 4194304	xs:int	2048	4194304				
CplStack_t	CPL Stack 1024 - 524288	xs:int	1024	524288				
BuffSizeExBl_t	Buffer Size External Block	xs:int	0	1048576				
MinLenCorn-Round_t	Min Block Length for Corner Rounding	xs:float	2	90				
MaxAngCorn-Round_t	Max Angle for Corner Rounding	xs:double	0	45				
NofAuxFun_t	Number of Auxiliary Functions	xs:int	0	1536				
SercBaudeRate_t	Sercos Baud Rate	xs:byte						2 4 8 16
TrTimeMdt_t	Transmission Time MDT	xs:unsignedInt	62	65535				

SimpleType	ann:	BaseType	Restrictions					
			Facets				Patterns	Enum.
			MinInc	MaxInc	Min. Length	Max. Length		
CycTime_t	Sercos Cycle Time = IPO cycle time	xs:unsignedInt	100	64000				
TrTimeMdt_t	Transmission Time MDT	xs:unsignedInt	62	65535				
CycTime_t	Sercos Cycle Time = IPO cycle time	xs:unsignedInt	100	64000				
DbHd_t	Data Base Header	isoLatin1String				4		IKQ1 IKQ2 IKQ3 IQ1 IQ2 IQ3
DECMV4_t	decimal with sign (4 bytes)	xs:double	-21474 83648	214748 3648				
DECMV2_t	decimal with sign (2 bytes)	xs:double	-32768	32768				
DECOV4_t	decimal without sign (4 bytes)	xs:double	-42949 67296	429496 7296				
DECOV2_t	decimal without sign (2 bytes)	xs:double	-65536	65536				
IDN_t	Sercos IDs	xs:string			8	8	[SP][-] [0-9][-] [0-9] (4.4)	

Tab. 10-2: Contents of the basic data type collection (basic_ty.xsd)

10.2 Tool lists configuration file

The following overview shows whether the list/editor control configuration file can be opened using the ULC configurator for all process parameters.

					Ⓐ	Ⓑ
	ListControlDefinition	ListDefinitions	»1			
		SubListDefinitions	»13			
1»	ListDefinitions	ListTitle	TitleType	»2		
		Styles	comRepDefType	»7		
		GridAttributes	»12			
2»	TitleType	TitleText			+	
		AssemblyName			+	
		ResourceName			+	

Appendix

			Ⓐ	Ⓑ
	Token		+	
	TokenOfUsertext		+	
	Alignment	AlignmentType »3		
	Font	FontType »4		
	Color	CellColour »5		
	Height		-	A
3»	AlignmentType	Align_vertical	o	B1
		Align_horizontal	o	B2
4»	FontType	FontID	o	C1
		FontSize	o	C2
		FontStyle	o	C3
5»	CellColour	Foreground		
		Background		
		ColourType »6		
		ColourType »6		
6»	ColourType	Red	-	D1
		Green	-	D2
		Blue	-	D3
		ColorName	o	D4
7»	comRepDefType	Normal		
		Style »8		
		Selected		
		Style »8		
		Highlighted		
		Style »8		
		Empty		
		Style »8		
		ColumnTitle		
		ColumnTitleType »11		
		SubColumnTitle		
		ColumnTitleType »11		
8»	Style	Fixed		
		SubStyle »9		
		Scrollable		
		SubStyle »9		
9»	SubStyle	Color		
		CellColour »5		
		Font		
		FontType »4		
		Alignment		
		AlignmentType »3		
		ImageAlignment		
		AlignmentType »3		
		OuterBorder		
		BorderType »10		
		InnerBorder		
		BorderType »10		
		WordWrap	-	E
		Display	-	F

				Ⓐ	Ⓑ
10»	BorderType	BorderSize		-	G
		BorderColour	ColourType »6		
		BorderStyle		-	H
11»	ColumnTitleType	Color	CellColour »5		
		Font	FontType »4		
		Alignment	AlignmentType »3		
		ImageAlignment	AlignmentType »3		
		InnerBorder	BorderType »10		
		WordWrap		-	E
12»	GridAttributes	BorderStyle		+	
		Stripline	BorderType »10		
		FilenameOfDefaultValues		+	
		AllowFocusOnNonEditableCells		+	
		ShowTraceMessages		+	
		ShowDebugMessages		+	
		AllowMerging		+	
		ListBackGroundColor	ColourType »6		
		KeyActionEnter		+	
		KeyActionTab		+	
		AllowFreezingWithMouse		+	
		AllowResizing		+	
		AutoSearch		+	
		ExtendLastCol		+	
		AutoResize		+	
		UseNumberDecimalSeparator		+	
		SuppressComma		+	
		CursorKeyCanCloseEditMode		+	
		TabCanCloseEditMode		+	
		DrawTextFlexgridOrg		+	
		AllowSorting		+	
		ScrollTrack		+	
		PageDownTrack		+	
		TestDOMToConfig		+	
		TestSOMToConfig		+	
		UserStringTableID		-	J
		GridHighlight		+	
		CheckRowVisibilityAtBeginning		+	

Appendix

			Ⓐ	Ⓑ
	SortEmptyRowsToEnd		-	V
	UseFixedNumberOfSubRows		-	W
	TextEditor	»47		
13»	SubListDefinitions	ScopeDefinition	ScopeDefinitionType	»14
		ColumnDefinitions	CommonColumnDefinitionsType	»15
		ColumnGeneralAttributes	»23	
		RowDefinition	»26	
		CellDefinitions	CellDefinitionsType	»29
		SublistProperties	»41	
14»	ScopeDefinitionType	PathOfMultiplier	-	K1
		ShowMultipleSublists	-	K2
		NoOfFixedColumn	-	K3
		NoOfScrollableColumn	-	K4
		NoOfFrozenColumn	-	K5
		NoOfFrozenRow	-	K6
15»	CommonColumnDefinitionsType	ColDef	ColumnDefinitionType	»16
16»	ColumnDefinitionType	PathOfMultiplier	-	L1
		ColTitle	ColumnTitleStyle	»17
		SubColumnDefinitions	SubColumnDefinitionsType	»18
17»	ColumnTitleStyle	TitleText	+	
		ProcessVariableID	+	
		AssemblyName	+	
		ResourceName	+	
		Token	+	
		TokenOfUserText	+	
18»	SubColumnDefinitionsType	SubColDef	»19	
19»	SubColDef	PathOfMultiplier	-	L2
		SubColTitle	ColumnTitleStyle	»20
		SubColWidth	+	
		CellRanges	»21	
		ColSortFlag	-	M

			Ⓐ	Ⓑ
	SortRow		-	N
20»	ColumnTitleStyle	TitleText	+	
		ProcessVariableID	+	
		AssemblyName	+	
		ResourceName	+	
		Token	+	
		TokenOfUsertext	+	
21»	CellRanges	CellRange	»22	
22»	CellRange	FirstRangeCellNo	+	
		LastRangeCellNo	+	
23»	ColumnGeneralAttributes	ColumnTitleAtributes	»24	
		SubColumnTitleAtributes	»25	
24»	ColumnTitleAtributes	ColumnTitleHeight	+	
25»	SubColumnTitleAtributes	ColumnTitleHeight	+	
26»	RowDefinition	SubRowDefinitions	»27	
27»	SubRowDefinitions	PathOfMultiplier	-	L3
		SubRowHeight	+	
		Dependency_for_Visibility	DependencyType	»28
28»	DependencyType	DepProcessVariableID	-	P1
		Operation	-	P2
		Value	-	P3
		Value2	-	P4
		FollowingDepCondition	-	P5
		Condition	DependencyType	»28
29»	CellDefinitionsType	CellDef	»30	
30»	CellDef	CellAddress	CellAddressType	»31
		ContentType	CellContentType	»32
		EditType	CellEditType	»36
		RepresentationDefinitions	»37	

Appendix

				Ⓐ	Ⓑ
31»	CellAddressType	line		+	
		Column		+	
		SubColumn		+	
32»	CellContentType	CellContentType		+	
		Comment		+	
		FixedImage		+	
		ProcessVariableID		+	
		ProcessVariableDataType		+	
		TextOrBitmap		+	
		AssemblyName		+	
		ResourceName		+	
		BitmapAssemblyName		+	
		BitmapResourceName		+	
		Token		+	
		TokenOfUsertext		+	
		ReadProcVarAllways		-	T
		BitmapTable	»33		
		BitmapTableCom		+	
		BitmapTableShowKey		-	
		TextTable	»34		
		TextTableCom		-	
		TextTableUse		-	
		TextTableShowKey		-	
33»	BitmapTable	KeyValuePair	»35		
»34	TextTable	KeyValuePair	»35		
34»	KeyValuePair	Key		o	Q1
		Value		o	Q2
36»	CellEditType	EditStatus		o	U
		EditTypeSelection		+	
37»	RepresentationDefinitions	Dependency_for_Representation	»38		
		TextFormat		+	
		NumFormat	NumFormatType »39		

				Ⓐ	Ⓑ
38»	Dependency_for_Representation	Dependency_for_Visibility	DependencyType	»28	
		Validation	DependencyType	»28	
		EditDepend	DependencyType	»28	
39»	NumFormatType	NumType			+
		DigitDef	»40		
		Pos_dec_Point_Def	»41		
		FormatString			+
		EditMask			+
40»	DigitDef	No_of_Digits			+
		DepVar			+
41»	Pos_dec_Point_Def	Position_after_dec_Point			+
		DepVar			+
42»	SublistProperties	Highlighted	»43		
		SublistActions	»45		
43»	Highlighted	TypeOfHighlight			- R
	IsHighlighted	IsHighlighted	»44		
44»	IsHighlighted	DependencyType	»28		
		HighlightColor	ColourType	»6	
45»	SublistActions	Action	»46		
46»	Action	TypeOfAction			- S
		IsCondition	DependencyType	»28	
47»	TextEditor	TextAlignment			- O1
		Font	»48		
48»	Font	FontStyle			- O2

Explanations of the process parameters that cannot, or only to a degree, be opened using the ULC configurator (indicated by - or o).

Ⓑ	Explanation	Remarks
A	Height of table title line	
B	1	Horizontal arrangement

Appendix

®	Explanation	Remarks
	2 Vertical arrangement	
C	1 Font	
	2 Font size	
	3 Font style	
D	1 Color setting	Red value
	2	Green value
	3	Blue value
	4	Color name Log. from color helper definition
E	Automatic line break	
F	Type of display	
G	Line width	
H	Line type	
J	Name of user text file	
K	1 Multiplication process variable	Multiplication of partial lists
	2 Display of partial lists	
	3 Number of fixed columns	
	4 Number of scrollable columns	Total number of main columns
	5 Number of frozen columns	
	6 Number of frozen lines	
L	1 Multiplication process variable	Multiplication of main columns
	2	Multiplication of subordinate columns
	3	Multiplication of subordinate lines
M	Column sorting permitted	
N	Line sorting permitted	
O	1 Text alignment	
	2 Font type	
P	1 Dep. on process variable	
	2 Operation / condition	

®	Explanation	Remarks
	3 Value	
	4 2nd value	(Multiplication process variable)
	5 Subsequent condition	
Q	1 Value of process variables	for bitmap table and text table
	2 Bitmap name or text token name	
R	Highlighted type	Currently not relevant
S	Action type	Currently not relevant
T	Control flag for reading the process variables	
U	This is the editability setting in the ULC configurator	Mode 3 cannot be selected (necessary for conditional editing)
V	Type of sorting	To achieve that empty locations are always displayed at the end of the list
W	Specified number of partial lists	The number of sublines of the first partial list determines the number for all partial lists

10.3 Interfaces

10.3.1 Status bits

BQ2 - Tool status bits (tool identification)

Data element	Status group	Name	Brief descr.	Bit no.	Description	Can be changed in MTGUI
BQ2	Tool status	ActiveTool	ta	1	Tool active	no
		PrimaryTool	tp	2	Primary tool	no
		UsedTool	tu	3	Used tool	no
		WarnLimit	tw	4	Warning limit reached	no
		WornOut	two	5	Tool worn	no
		PlaceCoded	TPC	6	Fixed place-coded tool	Yes
		Locked	TL	7	Tool locked	Yes
		Defective	TD	8	Tool defective	Yes
		Ignore	ti	9	Ignore tool	no
		Measured	tm	10	Probe tool	no
		ToLoad	ttl	11	Tool loaded	no
		ToUnload	ttu	12	Tool unloaded	no
		MasterTool	mt	13	Master tool	no
		ToolChange	tc	14	Tool change	no
		SisterTool	ts	15	Alternate tool	no
		HeavyTool	TH	16	Heavy tool	Yes
		NonEditable	tne	17	Tool cannot be edited	no
		Reserve 1		18	Reserve	
		Reserve 2		19	Reserve	
		Reserve 3		20	Reserve	
		Reserve 4		21	Reserve	
		Reserve 5		22	Reserve	
		Reserve 6		23	Reserve	
		Reserve 7		24	Reserve	
		UserStatus 1	TS1	25	Tool user status 1	Yes
		UserStatus 2	TS2	26	Tool user status 2	Yes
		UserStatus 3	TS3	27	Tool user status 3	Yes
		UserStatus 4	TS4	28	Tool user status 4	Yes
		UserStatus 5	TS5	29	Tool user status 5	Yes
		UserStatus 6	TS6	30	Tool user status 6	Yes
		UserStatus 7	TS7	31	Tool user status 7	Yes
		UserStatus 8	TS8	32	Tool user status 8	Yes

Tab. 10-3: BQ2 - Tool status

BQ1 - Location status bits

Data element	Status group	Name	Brief descr.	Bit no.	Description	Can be changed in MTGUI
BQ1	Place status	Free	pf	1	Free location	no
		Blocked	PB	2	Blocked location	Yes
		Ignore	pi	3	Ignore location	no
		Reserved	pr	4	Reserve location	no
		ReservedLeft	pri	5	Left half-location reserved	no
		ReservedRight	prr	6	Right half-location reserved	no
		ReservedLeftLower	prli	7	Left lower half-place reserved	no
		ReservedRightLower	prri	8	Right lower half-location reserved	no
		OccupiedLeft	pol	9	Left half-location assigned	no
		OccupiedRight	por	10	Right half-location assigned	no
		OccupiedLeftLower	poll	11	Left lower half-location assigned	no
		OccupiedRightLower	porl	12	Right lower half-location assigned	no
		PlaceTypeMonitoring	ptm	13	Location type machining	no
		DeviceTypeMonitoring	dtm	14	Device type machining	no
		FormTypeMonitoring	ftm	15	Form type machining	no
		DeviceInPlace	dip	16	Device in location	no
		Reserve 1		17	Reserve	
		Reserve 2		18	Reserve	
		Reserve 3		19	Reserve	
		Reserve 4		20	Reserve	
		Reserve 5		21	Reserve	
		Reserve 6		22	Reserve	
		Reserve 7		23	Reserve	
		Reserve 8		24	Reserve	
		PlaceUserStatus 1	PS1	25	Location user status 1	Yes
		PlaceUserStatus 2	PS2	26	Location user status 2	Yes
		PlaceUserStatus 3	PS3	27	Location user status 3	Yes
		PlaceUserStatus 4	PS4	28	Location user status 4	Yes
		PlaceUserStatus 5	PS5	29	Location user status 5	Yes
		PlaceUserStatus 6	PS6	30	Location user status 6	Yes
		PlaceUserStatus 7	PS7	31	Location user status 7	Yes
		PlaceUserStatus 8	PS8	32	Location user status 8	Yes

Tab. 10-4: BQ1 - Place status

Appendix

BQ3 - Technology status bits

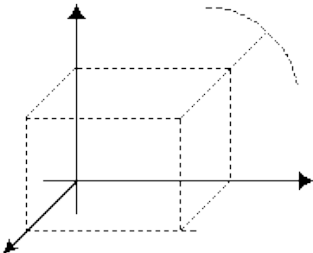
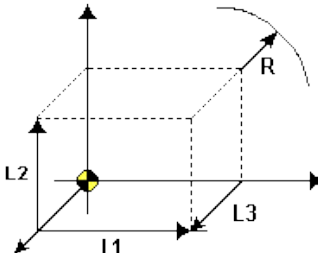
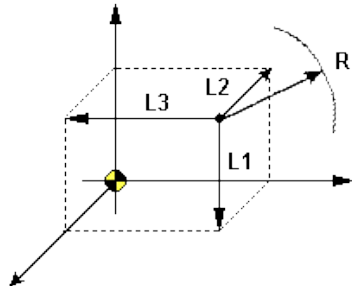
Data element	Status group	Name	Brief descr.	Bit no.	Description
BQ3	Technology status	NoOfEdges	EdNo	1	Edge number (1...16) Value range: 0 - 15
				2	
				3	
				4	
		Reserve 1	TCR1	5	Reserve
		Reserve 2	TCR2	6	Reserve
		Reserve 3	TCR3	7	Reserve
		CorrVal CH	CH	8	Angle head tool
		CorrVal L1	L1	9	L1 - Correction
		CorrVal L2	L2	10	L2 - Correction
		CorrVal L3	L3	11	L3 - Correction
		CorrVal R1	R1	12	R1 - Correction
		CorrVal R2	R2	13	R2 - Correction
		CorrVal O	O	14	Edge position
		CorrVal DIA	DIA	15	Correction value (L2) as diameter value
		ToolTech 1	T1	16	Tool technology value 1
		ToolTech 2	T2	17	Tool technology value 2
		ToolTech 3	T3	18	Tool technology value 3
		ToolTech 4	T4	19	Tool technology value 4
		ToolTech 5	T5	20	Tool technology value 5
		ToolTech 6	T6	21	Tool technology value 6
		ToolTech 7	T7	22	Tool technology value 7
		ToolTech 8	T8	23	Tool technology value 8
		EdgeTech 1	ET1	24	Edge technology value 1
		EdgeTech 2	ET2	25	Edge technology value 2
		Reserve 5		26	Reserve
		Reserve 6		27	Reserve
		Reserve 7		28	Reserve
		Reserve 8		29	Reserve
		Reserve 9		30	Reserve
		Reserve 10		31	Reserve
		Reserve 11		32	Reserve

Tab. 10-5: BQ3 - Technology status bits

10.3.2 Tool catalog

General machining

General tool

Technology		General machining	
Tool type name		General tool	
Screen			
Tool type		0000	
Edge number		can be set from 1 - max. edge number	
Tool property:			Rec/Hd/BQ3
	☐		Bit 5
	☐		Bit 6
Diameter input	☒	DIA	Bit 15
Tool correction value			Rec/UD/Ed/Geo/...
Length correction 1	☒	L1	L1
Length correction 2	☒	L2	L2
Length correction 3	☒	L3	L3
Radius correction	☒	R1	Rad
Edge position	☒	O	Ori
Tool technology date			Rec/UD/TI/Tech/...
Diameter	☒	d	Tech1
small diameter	☒	ds	Tech2
Cutting edge width	☒	tw	Tech3
Depth gauge	☒	tl	Tech4
Usable length	☒	lu	Tech5
Orientation angle	☒	tal	Tech6
Apical angle	☒	tac	Tech7
Number of teeth	☒	nt	Tech8
Edge technology date			Rec/UD/Ed/Tech/...
Edge dimension 1	☒	de	Tech1
Edge dimension 2	☒	lfe	Tech2

1)

Active if CH = TRUE (angle head tool)

2)

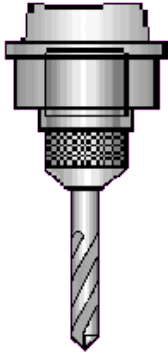
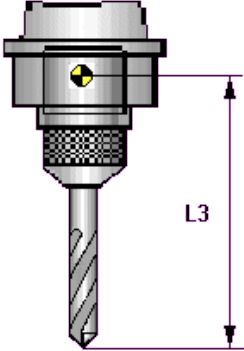
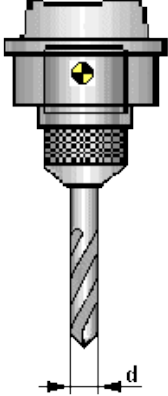
Value can be defined in the tool editor (©)

Tab. 10-6:

1000

Drilling tools

Drilling tool, general

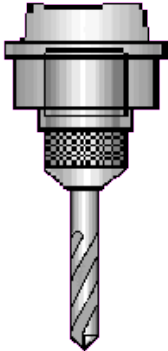
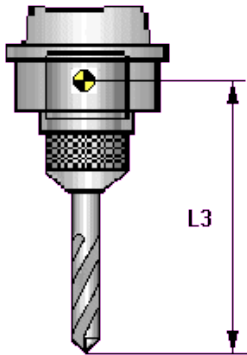
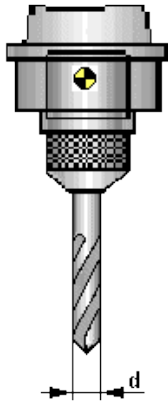
Technology		Drilling	
Tool type name		Drilling tool, general	
Screen			
Tool type		1000	
Edge number		can be set from 1 - max. edge number	
Tool property:			Rec/Hd/BQ3
	☐		Bit 5
Angle head tool	☒	CH ²⁾	Bit 6
	☐		Bit 15
Tool correction value:			Rec/UD/Ed/Geo/...
Length correction 1	☒	L1 ¹⁾	L1
Length correction 2	☒	L2 ¹⁾	L2
Length correction 3	☒	L3	L3
	☐		Rad
	☐		Ori
Tool technology date:			Rec/UD/TI/Tech/...
Diameter	☒	d	Tech1
	☐		Tech2
	☐		Tech3
	☐		Tech4
	☐		Tech5
	☐		Tech6
	☐		Tech7
	☐		Tech8
Edge technology date:			Rec/UD/Ed/Tech/...
	☐		Tech1
	☐		Tech2

1) Active if CH = TRUE (angle head tool)

2) Value can be defined in the tool editor (©)

Tab. 10-7: 1000

Twist drill

Technology		Drilling	
Tool type name		Twist drill	
Screen			
Tool type		1001	
Edge number		1	
Tool property:			Rec/Hd/BQ3
	☐		Bit 5
Angle head tool	☒	CH ²⁾	Bit 6
	☐		Bit 15
Tool correction value:			Rec/UD/Ed/Geo/...
Length correction 1	☒	L1 ¹⁾	L1
Length correction 2	☒	L2 ¹⁾	L2
Length correction 3	☒	L3	L3
	☐		Rad
	☐		Ori
Tool technology date:			Rec/UD/TI/Tech/...
Diameter	☒	d	Tech1
	☐		Tech2
Chamfer length	☒	lfc	Tech3
	☐		Tech4
Usable length	☒	lu	Tech5
	☐		Tech6
Apical angle	☒	tac	Tech7
	☐		Tech8
Edge technology date:			Rec/UD/Ed/Tech/...
	☐		Tech1
	☐		Tech2

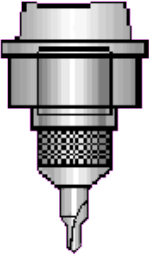
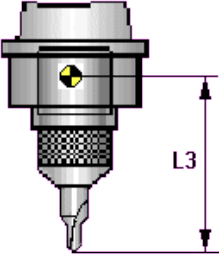
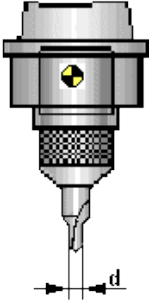
1) Active if CH = TRUE (angle head tool)

2) Value can be defined in the tool editor (⊕)

Tab. 10-8: 1001

Appendix

Center drill

Technology		Drilling	
Tool type name		Center drill	
Screen			
Tool type		1002	
Edge number		1	
Tool property:			Rec/Hd/BQ3
	☐		Bit 5
Angle head tool	☒	CH ²⁾	Bit 6
	☐		Bit 15
Tool correction value:			Rec/UD/Ed/Geo/...
Length correction 1	☒	L1 ¹⁾	L1
Length correction 2	☒	L2 ¹⁾	L2
Length correction 3	☒	L3	L3
	☐		Rad
	☐		Ori
Tool technology date:			Rec/UD/TI/Tech/...
Diameter	☒	d	Tech1
	☐		Tech2
	☐		Tech3
	☐		Tech4
	☐		Tech5
	☐		Tech6
	☐		Tech7
	☐		Tech8
Edge technology date:			Rec/UD/Ed/Tech/...
	☐		Tech1
	☐		Tech2

1) Active if CH = TRUE (angle head tool)

2) Value can be defined in the tool editor (©)

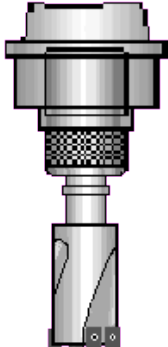
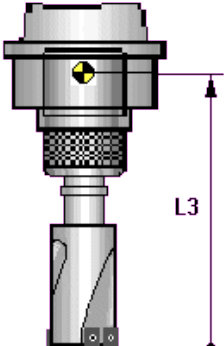
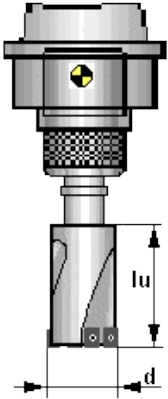
Tab. 10-9: 1002

1)	Active if CH = TRUE (angle head tool)
2)	Value can be defined in the tool editor (e)

Tab. 10-10: 1003

Appendix

Reversible tip drill

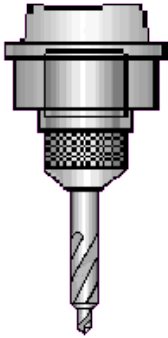
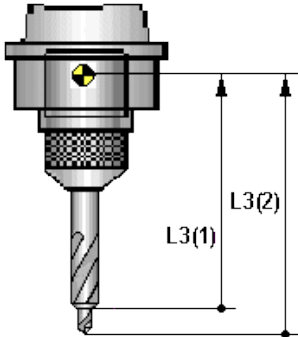
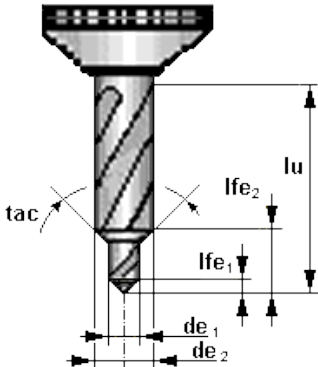
Technology		Drilling	
Tool type name		Reversible tip drill	
Screen			
Tool type		1004	
Edge number		1	
Tool property:			Rec/Hd/BQ3
	☐		Bit 5
Angle head tool	☒	CH ²⁾	Bit 6
	☐		Bit 15
Tool correction value:			Rec/UD/Ed/Geo/...
Length correction 1	☒	L1 ¹⁾	L1
Length correction 2	☒	L2 ¹⁾	L2
Length correction 3	☒	L3	L3
	☐		Rad
	☐		Ori
Tool technology date:			Rec/UD/TI/Tech/...
Diameter	☒	d	Tech1
	☐		Tech2
	☐		Tech3
	☐		Tech4
Usable length	☒	lu	Tech5
	☐		Tech6
	☐		Tech7
	☐		Tech8
Edge technology date:			Rec/UD/Ed/Tech/...
	☐		Tech1
	☐		Tech2

1) Active if CH = TRUE (angle head tool)

2) Value can be defined in the tool editor (©)

Tab. 10-11: 1004

Step drill 2 steps

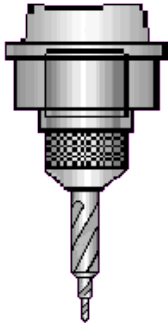
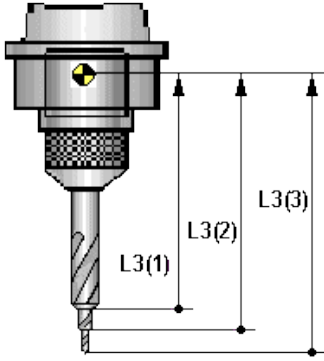
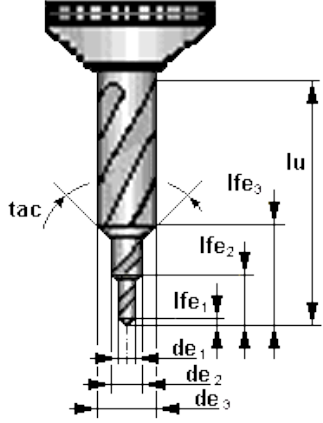
Technology		Drilling	
Tool type name		Step drill 2 steps	
Screen			
Tool type		1005	
Edge number		2	
Tool property:			Rec/Hd/BQ3
	☐		Bit 5
Angle head tool	☒	CH ²⁾	Bit 6
	☐		Bit 15
Tool correction value:			Rec/UD/Ed/Geo/...
Length correction 1	☒	L1 ¹⁾	L1
Length correction 2	☒	L2 ¹⁾	L2
Length correction 3	☒	L3	L3
	☐		Rad
	☐		Ori
Tool technology date:			Rec/UD/TI/Tech/...
	☐		Tech1
	☐		Tech2
	☐		Tech3
	☐		Tech4
Usable length	☒	lu	Tech5
	☐		Tech6
Countersink angle	☒	tac	Tech7
	☐		Tech8
Edge technology date:			Rec/UD/Ed/Tech/...
Diameter	☒	de	Tech1
Cut length	☒	lfe	Tech2

1) Active if CH = TRUE (angle head tool)

2) Value can be defined in the tool editor (⊕)

Tab. 10-12: 1005

Step drill 3 steps

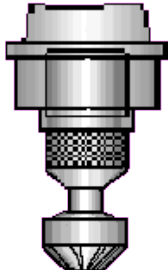
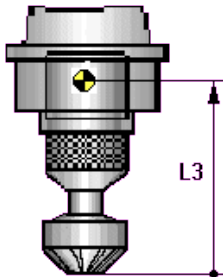
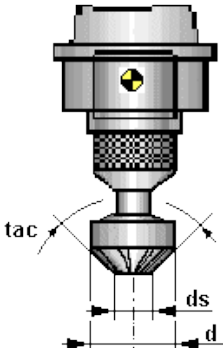
Technology		Drilling	
Tool type name		Step drill 3 steps	
Screen			
Tool type		1006	
Edge number		3	
Tool property:			Rec/Hd/BQ3
	☐		Bit 5
Angle head tool	☒	CH ²⁾	Bit 6
	☐		Bit 15
Tool correction value:			Rec/UD/Ed/Geo/...
Length correction 1	☒	L1 ¹⁾	L1
Length correction 2	☒	L2 ¹⁾	L2
Length correction 3	☒	L3	L3
	☐		Rad
	☐		Ori
Tool technology date:			Rec/UD/TI/Tech/...
	☐		Tech1
	☐		Tech2
	☐		Tech3
	☐		Tech4
Usable length	☒	lu	Tech5
	☐		Tech6
Countersink angle	☒	tac	Tech7
	☐		Tech8
Edge technology date:			Rec/UD/Ed/Tech/...
Diameter	☒	de	Tech1
Cut length	☒	lfe	Tech2

1) Active if CH = TRUE (angle head tool)

2) Value can be defined in the tool editor (©)

Tab. 10-13: 1006

Countersink

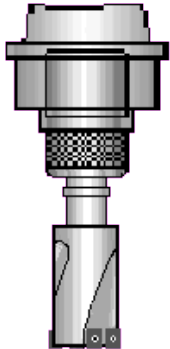
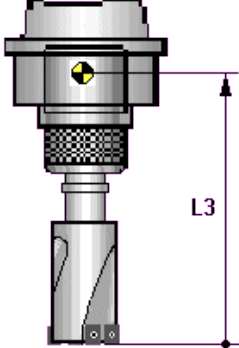
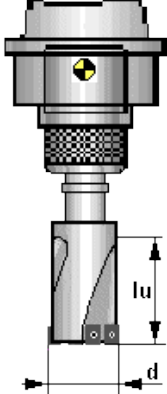
Technology		Drilling	
Tool type name		Countersink	
Screen			
Tool type		1007	
Edge number		1	
Tool property:			Rec/Hd/BQ3
	☐		Bit 5
Angle head tool	☒	CH ²⁾	Bit 6
	☐		Bit 15
Tool correction value:			Rec/UD/Ed/Geo/...
Length correction 1	☒	L1 ¹⁾	L1
Length correction 2	☒	L2 ¹⁾	L2
Length correction 3	☒	L3	L3
	☐		Rad
	☐		Ori
Tool technology date:			Rec/UD/TI/Tech/...
large diameter	☒	d	Tech1
small diameter	☒	ds	Tech2
	☐		Tech3
	☐		Tech4
	☐		Tech5
	☐		Tech6
Countersink angle	☒	tac	Tech7
	☐		Tech8
Edge technology date:			Rec/UD/Ed/Tech/...
	☐		Tech1
	☐		Tech2

1) Active if CH = TRUE (angle head tool)

2) Value can be defined in the tool editor (⊕)

Tab. 10-14: 1007

Plane countersink

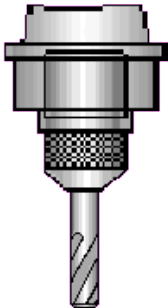
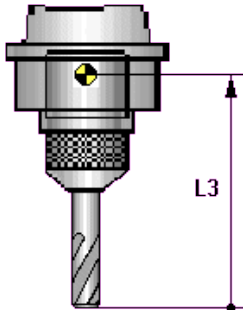
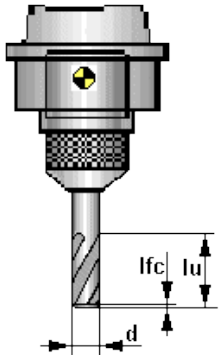
Technology		Drilling	
Tool type name		Plane countersink	
Screen			
Tool type		1008	
Edge number		1	
Tool property:			Rec/Hd/BQ3
	☐		Bit 5
Angle head tool	☒	CH ²⁾	Bit 6
	☐		Bit 15
Tool correction value:			Rec/UD/Ed/Geo/...
Length correction 1	☒	L1 ¹⁾	L1
Length correction 2	☒	L2 ¹⁾	L2
Length correction 3	☒	L3	L3
	☐		Rad
	☐		Ori
Tool technology date:			Rec/UD/TI/Tech/...
Diameter	☒	d	Tech1
	☐		Tech2
	☐		Tech3
	☐		Tech4
Usable length	☒	lu	Tech5
	☐		Tech6
	☐		Tech7
	☐		Tech8
Edge technology date:			Rec/UD/Ed/Tech/...
	☐		Tech1
	☐		Tech2

1) Active if CH = TRUE (angle head tool)

2) Value can be defined in the tool editor (©)

Tab. 10-15: 1008

Spiral countersink

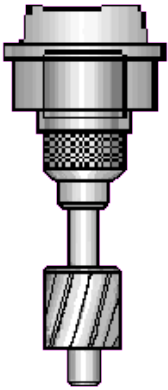
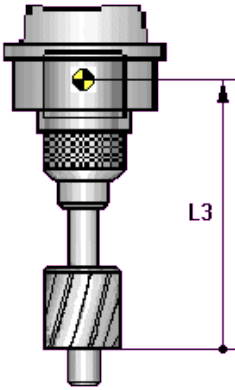
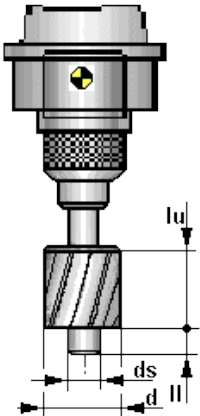
Technology		Drilling	
Tool type name		Spiral countersink	
Screen			
Tool type		1009	
Edge number		1	
Tool property:			Rec/Hd/BQ3
	☐		Bit 5
Angle head tool	☒	CH ²⁾	Bit 6
	☐		Bit 15
Tool correction value:			Rec/UD/Ed/Geo/...
Length correction 1	☒	L1 ¹⁾	L1
Length correction 2	☒	L2 ¹⁾	L2
Length correction 3	☒	L3	L3
	☐		Rad
	☐		Ori
Tool technology date:			Rec/UD/TI/Tech/...
Diameter	☒	d	Tech1
	☐		Tech2
Chamfer length	☒	lfc	Tech3
	☐		Tech4
Usable length	☒	lu	Tech5
	☐		Tech6
	☐		Tech7
	☐		Tech8
Edge technology date:			Rec/UD/Ed/Tech/...
	☐		Tech1
	☐		Tech2

1) Active if CH = TRUE (angle head tool)

2) Value can be defined in the tool editor (Ⓢ)

Tab. 10-16: 1009

Counterbore

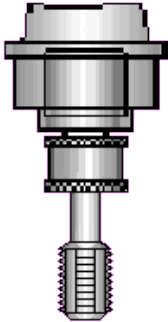
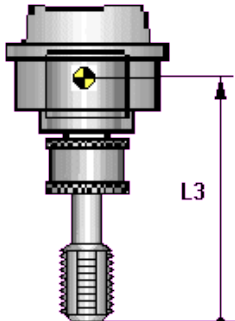
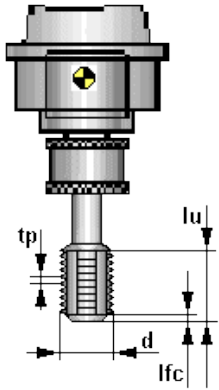
Technology		Drilling	
Tool type name		Counterbore	
Screen			
Tool type		1010	
Edge number		1	
Tool property:			Rec/Hd/BQ3
	☐		Bit 5
Angle head tool	☒	CH ²⁾	Bit 6
	☐		Bit 15
Tool correction value:			Rec/UD/Ed/Geo/...
Length correction 1	☒	L1 ¹⁾	L1
Length correction 2	☒	L2 ¹⁾	L2
Length correction 3	☒	L3	L3
	☐		Rad
	☐		Ori
Tool technology date:			Rec/UD/TI/Tech/...
large diameter	☒	d	Tech1
small diameter	☒	ds	Tech2
Spigot length	☒	ll	Tech3
	☐		Tech4
Usable length	☒	lu	Tech5
	☐		Tech6
	☐		Tech7
	☐		Tech8
Edge technology date:			Rec/UD/Ed/Tech/...
	☐		Tech1
	☐		Tech2

1) Active if CH = TRUE (angle head tool)

2) Value can be defined in the tool editor (©)

Tab. 10-17: 1010

Tap

Technology		Drilling	
Tool type name		Tap	
Screen			
Tool type		1011	
Edge number		1	
Tool property:			Rec/Hd/BQ3
	☐		Bit 5
Angle head tool	☒	CH ²⁾	Bit 6
	☐		Bit 15
Tool correction value:			Rec/UD/Ed/Geo/...
Length correction 1	☒	L1 ¹⁾	L1
Length correction 2	☒	L2 ¹⁾	L2
Length correction 3	☒	L3	L3
	☐		Rad
	☐		Ori
Tool technology date:			Rec/UD/TI/Tech/...
Diameter	☒	d	Tech1
	☐		Tech2
Chamfer length	☒	lfc	Tech3
	☐		Tech4
Usable length	☒	lu	Tech5
Thread pitch	☒	tp	Tech6
	☐		Tech7
	☐		Tech8
Edge technology date:			Rec/UD/Ed/Tech/...
	☐		Tech1
	☐		Tech2

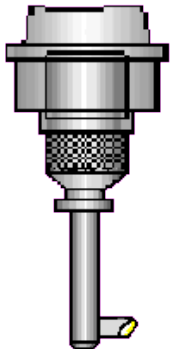
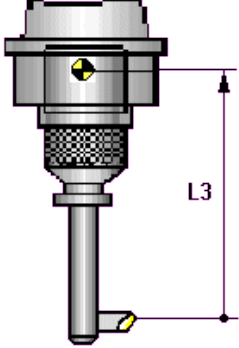
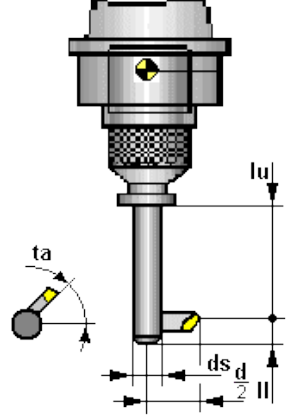
1) Active if CH = TRUE (angle head tool)

2) Value can be defined in the tool editor (Ⓜ)

Tab. 10-18: 1011

Appendix

Boring bar

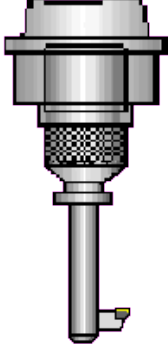
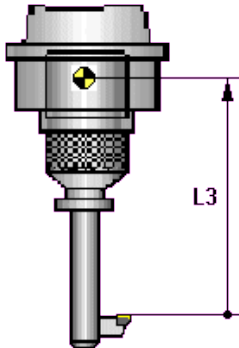
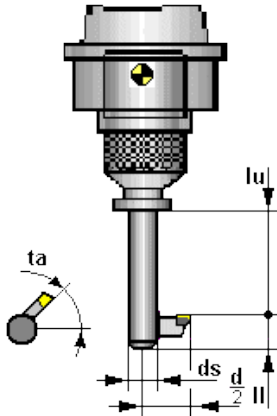
Technology		Drilling	
Tool type name		Boring bar	
Screen			
Tool type		1012	
Edge number		1	
Tool property:			Rec/Hd/BQ3
	○		Bit 5
Angle head tool	ⓔ	CH ²⁾	Bit 6
	○		Bit 15
Tool correction value:			Rec/UD/Ed/Geo/...
Length correction 1	● ¹⁾	L1 ¹⁾	L1
Length correction 2	● ¹⁾	L2 ¹⁾	L2
Length correction 3	●	L3	L3
	○		Rad
	○		Ori
Tool technology date:			Rec/UD/TI/Tech/...
Diameter	●	d	Tech1
Shaft diameter	●	ds	Tech2
Loss length	●	ll	Tech3
	○		Tech4
Usable length	●	lu	Tech5
Setting angle	●	ta	Tech6
	○		Tech7
	○		Tech8
Edge technology date:			Rec/UD/Ed/Tech/...
	○		Tech1
	○		Tech2

1) Active if CH = TRUE (angle head tool)

2) Value can be defined in the tool editor (ⓔ)

Tab. 10-19: 1012

Backward counterbore

Technology		Drilling	
Tool type name		Backward counterbore	
Screen			
Tool type		1013	
Edge number		1	
Tool property:			Rec/Hd/BQ3
	○		Bit 5
Angle head tool	⊕	CH ²⁾	Bit 6
	○		Bit 15
Tool correction value:			Rec/UD/Ed/Geo/...
Length correction 1	● ¹⁾	L1 ¹⁾	L1
Length correction 2	● ¹⁾	L2 ¹⁾	L2
Length correction 3	●	L3	L3
	○		Rad
	○		Ori
Tool technology date:			Rec/UD/TI/Tech/...
Diameter	●	d	Tech1
Shaft diameter	●	ds	Tech2
Loss length	●	l	Tech3
	○		Tech4
Usable length	●	lu	Tech5
Setting angle	●	ta	Tech6
	○		Tech7
	○		Tech8
Edge technology date:			Rec/UD/Ed/Tech/...
	○		Tech1
	○		Tech2

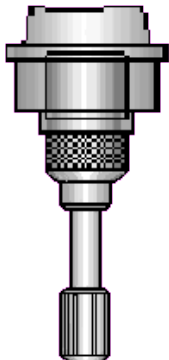
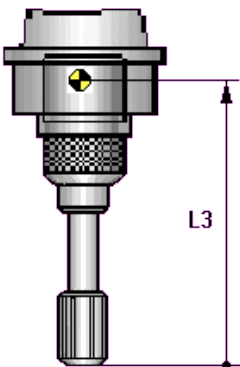
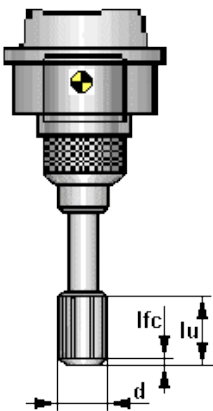
1) Active if CH = TRUE (angle head tool)

2) Value can be defined in the tool editor (⊕)

Tab. 10-20: 1013

Appendix

Reamer

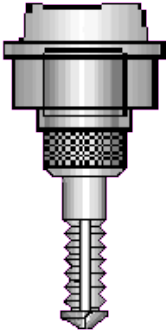
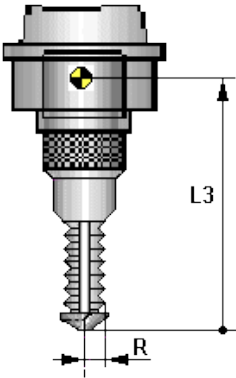
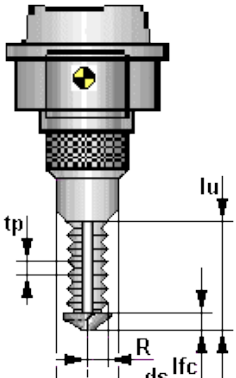
Technology		Drilling	
Tool type name		Reamer	
Screen			
Tool type		1014	
Edge number		1	
Tool property:			Rec/Hd/BQ3
	☐		Bit 5
Angle head tool	☒	CH ²⁾	Bit 6
	☐		Bit 15
Tool correction value:			Rec/UD/Ed/Geo/...
Length correction 1	☒	L1 ¹⁾	L1
Length correction 2	☒	L2 ¹⁾	L2
Length correction 3	☒	L3	L3
	☐		Rad
	☐		Ori
Tool technology date:			Rec/UD/TI/Tech/...
Diameter	☒	d	Tech1
	☐		Tech2
Chamfer length	☒	lfc	Tech3
	☐		Tech4
Usable length	☒	lu	Tech5
	☐		Tech6
	☐		Tech7
	☐		Tech8
Edge technology date:			Rec/UD/Ed/Tech/...
	☐		Tech1
	☐		Tech2

1) Active if CH = TRUE (angle head tool)

2) Value can be defined in the tool editor (©)

Tab. 10-21: 1014

Drill and thread milling cutter

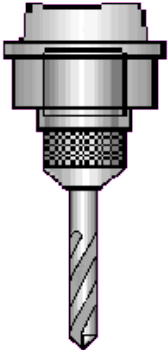
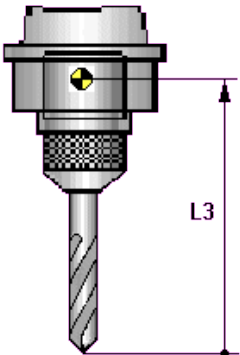
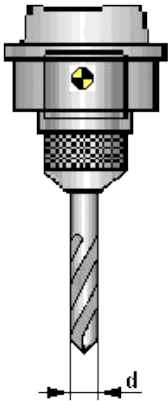
Technology		Drilling	
Tool type name		Drill and thread milling cutter	
Screen			
Tool type		1015	
Edge number		1	
Tool property:			Rec/Hd/BQ3
	○		Bit 5
Angle head tool	⊕	CH ²⁾	Bit 6
	○		Bit 15
Tool correction value:			Rec/UD/Ed/Geo/...
Length correction 1	● ¹⁾	L1 ¹⁾	L1
Length correction 2	● ¹⁾	L2 ¹⁾	L2
Length correction 3	●	L3	L3
	○		Rad
	○		Ori
Tool technology date:			Rec/UD/TI/Tech/...
Nominal diameter of thread	●	d	Tech1
Countersinking diameter	●	ds	Tech2
Length of tool tip	●	lfc	Tech3
	○		Tech4
Usable length	●	lu	Tech5
Thread pitch	●	tp	Tech6
	○		Tech7
	○		Tech8
Edge technology date:			Rec/UD/Ed/Tech/...
	○		Tech1
	○		Tech2

1) Active if CH = TRUE (angle head tool)

2) Value can be defined in the tool editor (⊕)

Tab. 10-22: 1015

Special drilling tool

Technology		Drilling	
Tool type name		Special drilling tool	
Screen			
Tool type		1016	
Edge number		can be set from 1 - max. edge number	
Tool property:			Rec/Hd/BQ3
	☐		Bit 5
Angle head tool	☒	CH ²⁾	Bit 6
	☐		Bit 15
Tool correction value:			Rec/UD/Ed/Geo/...
Length correction 1	☒	L1 ¹⁾	L1
Length correction 2	☒	L2 ¹⁾	L2
Length correction 3	☒	L3	L3
	☐		Rad
	☐		Ori
Tool technology date:			Rec/UD/TI/Tech/...
Diameter	☒	d	Tech1
	☐		Tech2
	☐		Tech3
	☐		Tech4
	☐		Tech5
	☐		Tech6
	☐		Tech7
	☐		Tech8
Edge technology date:			Rec/UD/Ed/Tech/...
	☐		Tech1
	☐		Tech2

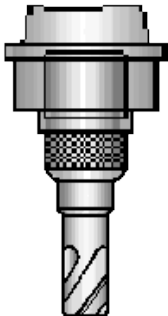
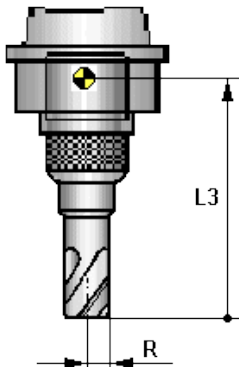
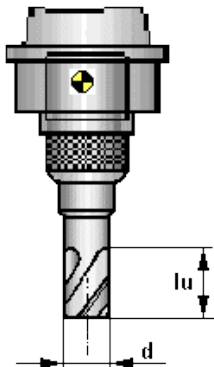
1) Active if CH = TRUE (angle head tool)

2) Value can be defined in the tool editor (©)

Tab. 10-23: 1016

Milling tools

Milling tool, general

Technology		Milling	
Tool type name		Milling tool, general	
Screen			
Tool type		2000	
Edge number		can be set from 1 - max. edge number	
Tool property:			Rec/Hd/BQ3
	☐		Bit 5
Angle head tool	☒	CH ²⁾	Bit 6
	☐		Bit 15
Tool correction value:			Rec/UD/Ed/Geo/...
Length correction 1	☒	L1 ¹⁾	L1
Length correction 2	☒	L2 ¹⁾	L2
Length correction 3	☐	L3	L3
Radius correction	☐	R1	Rad
Edge position	☒	O ¹⁾	Ori
Tool technology date:			Rec/UD/TI/Tech/...
Diameter	☒	d	Tech1
	☐		Tech2
	☐		Tech3
	☐		Tech4
Usable length	☒	lu	Tech5
	☐		Tech6
	☐		Tech7
Number of teeth	☒	nt	Tech8
Edge technology date:			Rec/UD/Ed/Tech/...
	☐		Tech1
	☐		Tech2

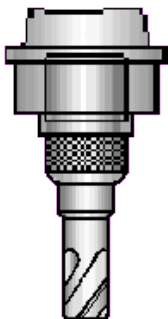
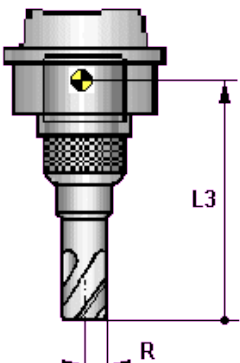
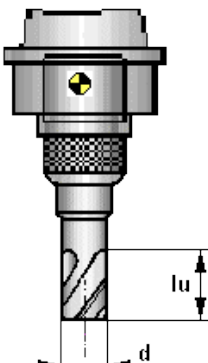
1) Active if CH = TRUE (angle head tool)

2) Value can be defined in the tool editor (⊕)

Tab. 10-24: 2000

Appendix

End mill

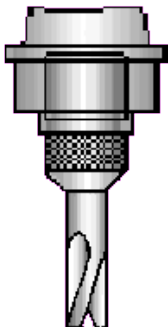
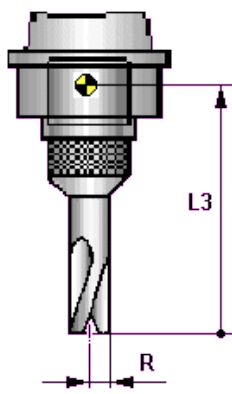
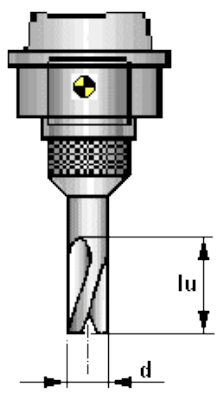
Technology		Milling	
Tool type name		End mill	
Screen			
Tool type		2001	
Edge number		1	
Tool property:			Rec/Hd/BQ3
	☐		Bit 5
Angle head tool	☒	CH ²⁾	Bit 6
	☐		Bit 15
Tool correction value:			Rec/UD/Ed/Geo/...
Length correction 1	☒	L1 ¹⁾	L1
Length correction 2	☒	L2 ¹⁾	L2
Length correction 3	☐	L3	L3
Radius correction	☐	R1	Rad
Edge position	☒	O ¹⁾	Ori
Tool technology date:			Rec/UD/TI/Tech/...
Diameter	☒	d	Tech1
	☐		Tech2
	☐		Tech3
	☐		Tech4
Usable length	☒	lu	Tech5
	☐		Tech6
	☐		Tech7
Number of teeth	☒	nt	Tech8
Edge technology date:			Rec/UD/Ed/Tech/...
	☐		Tech1
	☐		Tech2

1) Active if CH = TRUE (angle head tool)

2) Value can be defined in the tool editor (©)

Tab. 10-25: 2001

Groove milling cutter

Technology		Milling	
Tool type name		Groove milling cutter	
Screen			
Tool type		2002	
Edge number		1	
Tool property:			Rec/Hd/BQ3
	○		Bit 5
Angle head tool	⊕	CH ²⁾	Bit 6
	○		Bit 15
Tool correction value:			Rec/UD/Ed/Geo/...
Length correction 1	● ¹⁾	L1 ¹⁾	L1
Length correction 2	● ¹⁾	L2 ¹⁾	L2
Length correction 3	●	L3	L3
Radius correction	●	R1	Rad
Edge position	● ¹⁾	O ¹⁾	Ori
Tool technology date:			Rec/UD/TI/Tech/...
Diameter	●	d	Tech1
	○		Tech2
	○		Tech3
	○		Tech4
Usable length	●	lu	Tech5
	○		Tech6
	○		Tech7
Number of teeth	●	nt	Tech8
Edge technology date:			Rec/UD/Ed/Tech/...
	○		Tech1
	○		Tech2

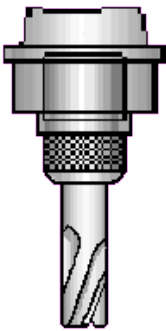
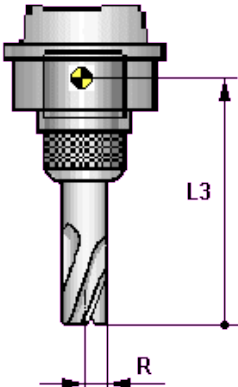
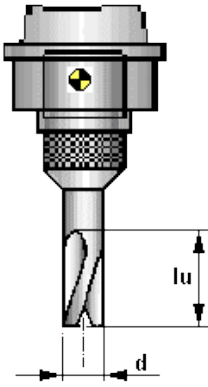
1) Active if CH = TRUE (angle head tool)

2) Value can be defined in the tool editor (⊕)

Tab. 10-26: 2002

Appendix

Long-hole milling cutter

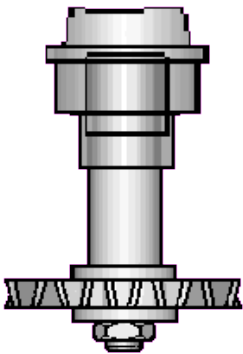
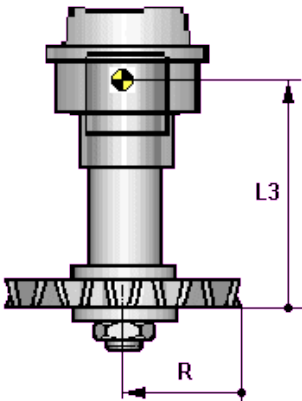
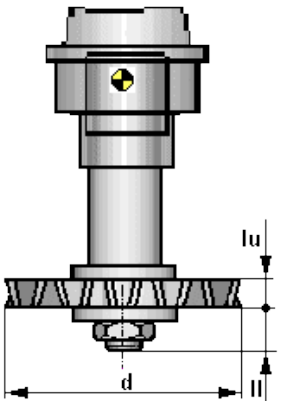
Technology		Milling	
Tool type name		Long-hole milling cutter	
Screen			
Tool type		2003	
Edge number		1	
Tool property:			Rec/Hd/BQ3
	☐		Bit 5
Angle head tool	☒	CH ²⁾	Bit 6
	☐		Bit 15
Tool correction value:			Rec/UD/Ed/Geo/...
Length correction 1	☒	L1 ¹⁾	L1
Length correction 2	☒	L2 ¹⁾	L2
Length correction 3	☐	L3	L3
Radius correction	☐	R1	Rad
Edge position	☒	O ¹⁾	Ori
Tool technology date:			Rec/UD/TI/Tech/...
Diameter	☐	d	Tech1
	☐		Tech2
	☐		Tech3
	☐		Tech4
Usable length	☐	lu	Tech5
	☐		Tech6
	☐		Tech7
Number of teeth	☐	nt	Tech8
Edge technology date:			Rec/UD/Ed/Tech/...
	☐		Tech1
	☐		Tech2

1) Active if CH = TRUE (angle head tool)

2) Value can be defined in the tool editor (©)

Tab. 10-27: 2003

Disk milling cutter

Technology		Milling	
Tool type name		Disk milling cutter	
Screen			
Tool type		2004	
Edge number		1	
Tool property:			Rec/Hd/BQ3
	○		Bit 5
Angle head tool	⊕	CH ²⁾	Bit 6
	○		Bit 15
Tool correction value:			Rec/UD/Ed/Geo/...
Length correction 1	● ¹⁾	L1 ¹⁾	L1
Length correction 2	● ¹⁾	L2 ¹⁾	L2
Length correction 3	●	L3	L3
Radius correction	●	R1	Rad
Edge position	● ¹⁾	O ¹⁾	Ori
Tool technology date:			Rec/UD/TI/Tech/...
Diameter	●	d	Tech1
	○		Tech2
Loss length	●	ll	Tech3
	○		Tech4
Usable length	●	lu	Tech5
	○		Tech6
	○		Tech7
Number of teeth	●	nt	Tech8
Edge technology date:			Rec/UD/Ed/Tech/...
	○		Tech1
	○		Tech2

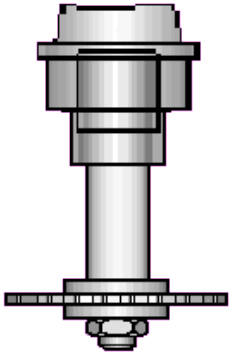
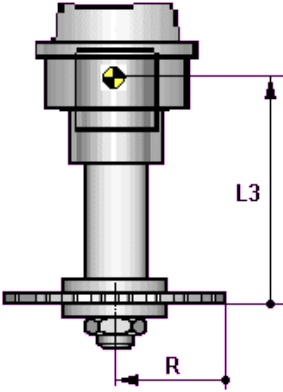
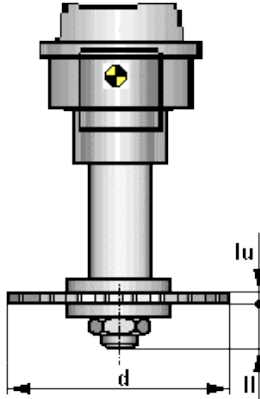
1) Active if CH = TRUE (angle head tool)

2) Value can be defined in the tool editor (⊕)

Tab. 10-28: 2004

Appendix

Saw blade

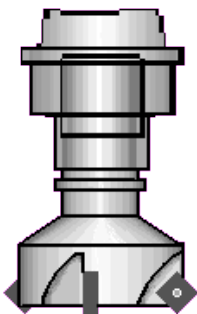
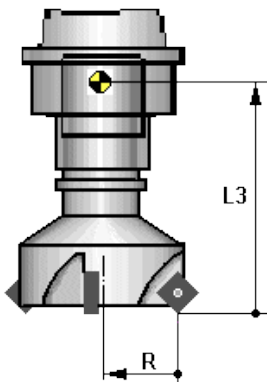
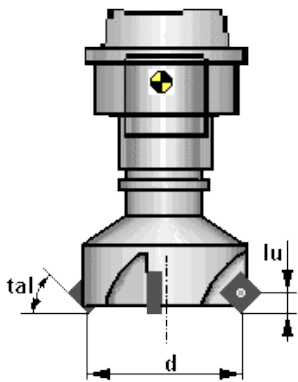
Technology		Milling	
Tool type name		Saw blade	
Screen			
Tool type		2005	
Edge number		1	
Tool property:			Rec/Hd/BQ3
	○		Bit 5
Angle head tool	ⓔ	CH ²⁾	Bit 6
	○		Bit 15
Tool correction value:			Rec/UD/Ed/Geo/...
Length correction 1	● ¹⁾	L1 ¹⁾	L1
Length correction 2	● ¹⁾	L2 ¹⁾	L2
Length correction 3	●	L3	L3
Radius correction	●	R1	Rad
Edge position	● ¹⁾	O ¹⁾	Ori
Tool technology date:			Rec/UD/TI/Tech/...
Diameter	●	d	Tech1
	○		Tech2
Loss length	●	ll	Tech3
	○		Tech4
Usable length	●	lu	Tech5
	○		Tech6
	○		Tech7
Number of teeth	●	nt	Tech8
Edge technology date:			Rec/UD/Ed/Tech/...
	○		Tech1
	○		Tech2

1) Active if CH = TRUE (angle head tool)

2) Value can be defined in the tool editor (ⓔ)

Tab. 10-29: 2005

Face-milling cutter

Technology		Milling	
Tool type name		Face-milling cutter	
Screen			
Tool type		2006	
Edge number		1	
Tool property:			Rec/Hd/BQ3
	○		Bit 5
Angle head tool	⊕	CH ²⁾	Bit 6
	○		Bit 15
Tool correction value:			Rec/UD/Ed/Geo/...
Length correction 1	● ¹⁾	L1 ¹⁾	L1
Length correction 2	● ¹⁾	L2 ¹⁾	L2
Length correction 3	●	L3	L3
Radius correction	●	R1	Rad
Edge position	● ¹⁾	O ¹⁾	Ori
Tool technology date:			Rec/UD/TI/Tech/...
Diameter	●	d	Tech1
	○		Tech2
	○		Tech3
	○		Tech4
Usable length	●	lu	Tech5
Edge angle	●	tal	Tech6
	○		Tech7
Number of teeth	●	nt	Tech8
Edge technology date:			Rec/UD/Ed/Tech/...
	○		Tech1
	○		Tech2

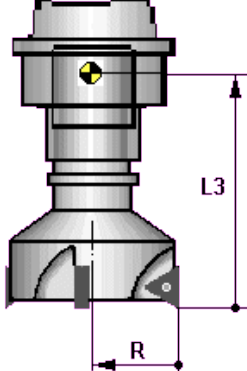
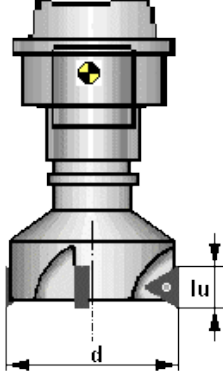
1) Active if CH = TRUE (angle head tool)

2) Value can be defined in the tool editor (⊕)

Tab. 10-30: 2006

Appendix

Corner milling cutter

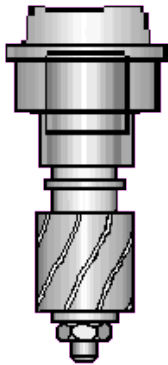
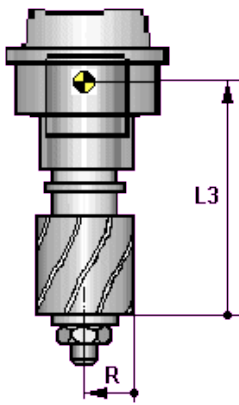
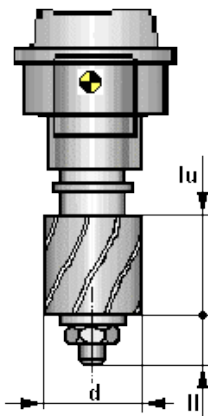
Technology		Milling	
Tool type name		Corner milling cutter	
Screen			
Tool type		2007	
Edge number		1	
Tool property:			Rec/Hd/BQ3
	○		Bit 5
Angle head tool	⊕	CH ²⁾	Bit 6
	○		Bit 15
Tool correction value:			Rec/UD/Ed/Geo/...
Length correction 1	● ¹⁾	L1 ¹⁾	L1
Length correction 2	● ¹⁾	L2 ¹⁾	L2
Length correction 3	●	L3	L3
Radius correction	●	R1	Rad
Edge position	● ¹⁾	O ¹⁾	Ori
Tool technology date:			Rec/UD/TI/Tech/...
Diameter	●	d	Tech1
	○		Tech2
	○		Tech3
	○		Tech4
Usable length	●	lu	Tech5
	○		Tech6
	○		Tech7
Number of teeth	●	nt	Tech8
Edge technology date:			Rec/UD/Ed/Tech/...
	○		Tech1
	○		Tech2

1) Active if CH = TRUE (angle head tool)

2) Value can be defined in the tool editor (⊕)

Tab. 10-31: 2007

Plain milling cutter

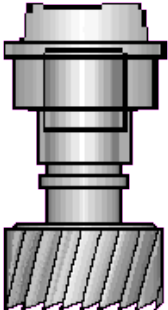
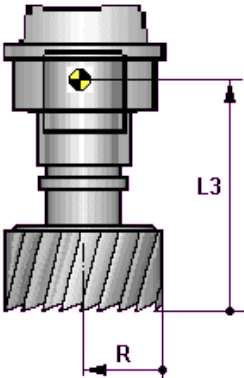
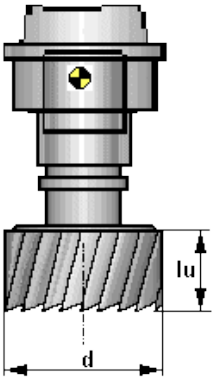
Technology		Milling	
Tool type name		Plain milling cutter	
Screen			
Tool type		2008	
Edge number		1	
Tool property:			Rec/Hd/BQ3
	○		Bit 5
Angle head tool	⊕	CH ²⁾	Bit 6
	○		Bit 15
Tool correction value:			Rec/UD/Ed/Geo/...
Length correction 1	● ¹⁾	L1 ¹⁾	L1
Length correction 2	● ¹⁾	L2 ¹⁾	L2
Length correction 3	●	L3	L3
Radius correction	●	R1	Rad
Edge position	● ¹⁾	O ¹⁾	Ori
Tool technology date:			Rec/UD/TI/Tech/...
Diameter	●	d	Tech1
	○		Tech2
Loss length	●	ll	Tech3
	○		Tech4
Usable length	●	lu	Tech5
	○		Tech6
	○		Tech7
Number of teeth	●	nt	Tech8
Edge technology date:			Rec/UD/Ed/Tech/...
	○		Tech1
	○		Tech2

1) Active if CH = TRUE (angle head tool)

2) Value can be defined in the tool editor (⊕)

Tab. 10-32: 2008

Shell end mill

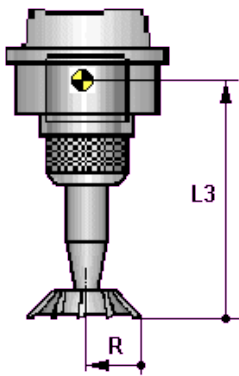
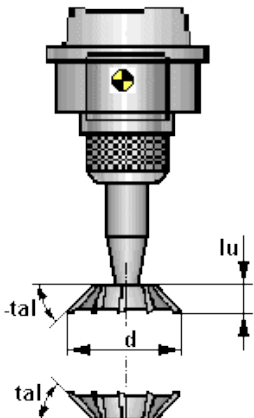
Technology		Milling	
Tool type name		Shell end mill	
Screen			
Tool type		2009	
Edge number		1	
Tool property:			Rec/Hd/BQ3
	○		Bit 5
Angle head tool	ⓔ	CH ²⁾	Bit 6
	○		Bit 15
Tool correction value:			Rec/UD/Ed/Geo/...
Length correction 1	● ¹⁾	L1 ¹⁾	L1
Length correction 2	● ¹⁾	L2 ¹⁾	L2
Length correction 3	●	L3	L3
Radius correction	●	R1	Rad
Edge position	● ¹⁾	O ¹⁾	Ori
Tool technology date:			Rec/UD/TI/Tech/...
Diameter	●	d	Tech1
	○		Tech2
	○		Tech3
	○		Tech4
Usable length	●	lu	Tech5
	○		Tech6
	○		Tech7
Number of teeth	●	nt	Tech8
Edge technology date:			Rec/UD/Ed/Tech/...
	○		Tech1
	○		Tech2

1) Active if CH = TRUE (angle head tool)

2) Value can be defined in the tool editor (ⓔ)

Tab. 10-33: 2009

Angular milling cutter

Technology		Milling	
Tool type name		Angular milling cutter	
Screen			
Tool type		2010	
Edge number		1	
Tool property:			Rec/Hd/BQ3
	○		Bit 5
Angle head tool	⊕	CH ²⁾	Bit 6
	○		Bit 15
Tool correction value:			Rec/UD/Ed/Geo/...
Length correction 1	● ¹⁾	L1 ¹⁾	L1
Length correction 2	● ¹⁾	L2 ¹⁾	L2
Length correction 3	●	L3	L3
Radius correction	●	R1	Rad
Edge position	● ¹⁾	O ¹⁾	Ori
Tool technology date:			Rec/UD/TI/Tech/...
Outer diameter	●	d	Tech1
	○		Tech2
	○		Tech3
	○		Tech4
Usable length	●	lu	Tech5
Angle (kappa)	●	tal	Tech6
	○		Tech7
Number of teeth	●	nt	Tech8
Edge technology date:			Rec/UD/Ed/Tech/...
	○		Tech1
	○		Tech2

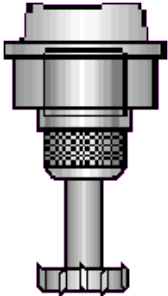
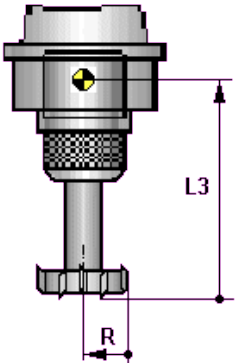
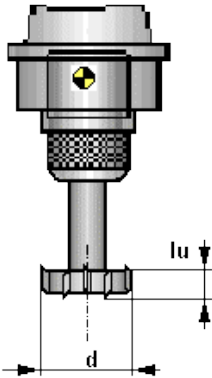
1) Active if CH = TRUE (angle head tool)

2) Value can be defined in the tool editor (⊕)

Tab. 10-34: 2010

Appendix

T-groove cutter

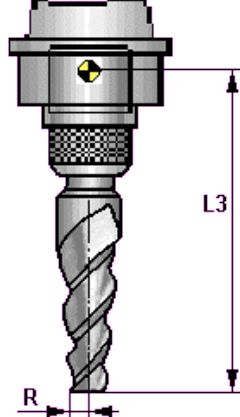
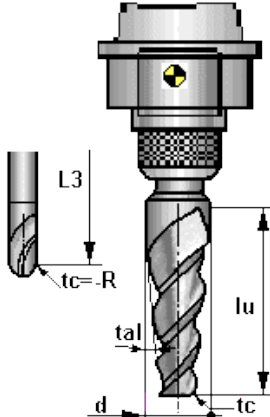
Technology		Milling	
Tool type name		T-groove cutter	
Screen			
Tool type		2011	
Edge number		1	
Tool property:			Rec/Hd/BQ3
	○		Bit 5
Angle head tool	⊕	CH ²⁾	Bit 6
	○		Bit 15
Tool correction value:			Rec/UD/Ed/Geo/...
Length correction 1	● ¹⁾	L1 ¹⁾	L1
Length correction 2	● ¹⁾	L2 ¹⁾	L2
Length correction 3	●	L3	L3
Radius correction	●	R1	Rad
Edge position	● ¹⁾	O ¹⁾	Ori
Tool technology date:			Rec/UD/TI/Tech/...
Diameter	●	d	Tech1
	○		Tech2
	○		Tech3
	○		Tech4
Usable length	●	lu	Tech5
	○		Tech6
	○		Tech7
Number of teeth	●	nt	Tech8
Edge technology date:			Rec/UD/Ed/Tech/...
	○		Tech1
	○		Tech2

1) Active if CH = TRUE (angle head tool)

2) Value can be defined in the tool editor (⊕)

Tab. 10-35: 2011

Die-sinking cutter

Technology		Milling	
Tool type name		Die-sinking cutter	
Screen			
Tool type		2012	
Edge number		1	
Tool property:			Rec/Hd/BQ3
	○		Bit 5
Angle head tool	⊕	CH ²⁾	Bit 6
	○		Bit 15
Tool correction value:			Rec/UD/Ed/Geo/...
Length correction 1	● ¹⁾	L1 ¹⁾	L1
Length correction 2	● ¹⁾	L2 ¹⁾	L2
Length correction 3	●	L3	L3
Radius correction	●	R1	Rad
Edge position	● ¹⁾	O ¹⁾	Ori
Tool technology date:			Rec/UD/TI/Tech/...
Diameter	●	d	Tech1
	○		Tech2
Sphere radius	●	tc	Tech3
	○		Tech4
Usable length	●	lu	Tech5
Angle	●	tal	Tech6
	○		Tech7
Number of teeth	●	nt	Tech8
Edge technology date:			Rec/UD/Ed/Tech/...
	○		Tech1
	○		Tech2

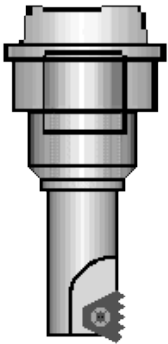
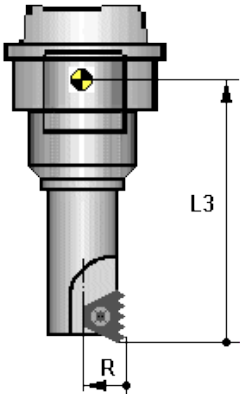
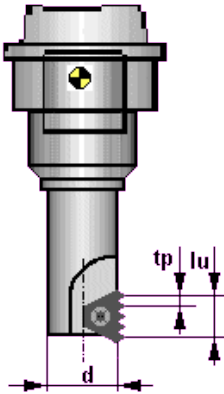
1) Active if CH = TRUE (angle head tool)

2) Value can be defined in the tool editor (⊕)

Tab. 10-36: 2012

Appendix

Thread milling cutter

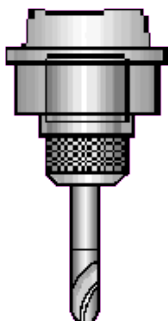
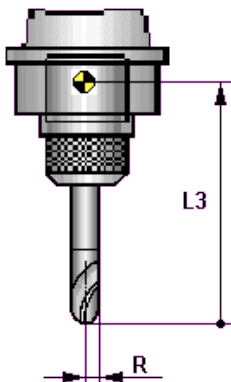
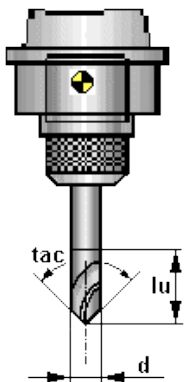
Technology		Milling	
Tool type name		Thread milling cutter	
Screen			
Tool type		2013	
Edge number		1	
Tool property:			Rec/Hd/BQ3
	○		Bit 5
Angle head tool	⊕	CH ²⁾	Bit 6
	○		Bit 15
Tool correction value:			Rec/UD/Ed/Geo/...
Length correction 1	● ¹⁾	L1 ¹⁾	L1
Length correction 2	● ¹⁾	L2 ¹⁾	L2
Length correction 3	●	L3	L3
Radius correction	●	R1	Rad
Edge position	● ¹⁾	O ¹⁾	Ori
Tool technology date:			Rec/UD/TI/Tech/...
Shaft diameter	●	d	Tech1
	○		Tech2
	○		Tech3
	○		Tech4
Usable length	●	lu	Tech5
Thread pitch	●	tp	Tech6
	○		Tech7
Number of teeth	●	nt	Tech8
Edge technology date:			Rec/UD/Ed/Tech/...
	○		Tech1
	○		Tech2

1) Active if CH = TRUE (angle head tool)

2) Value can be defined in the tool editor (⊕)

Tab. 10-37: 2013

Engraving tool

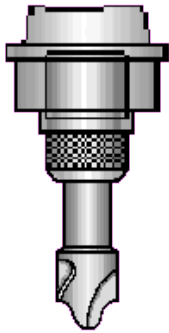
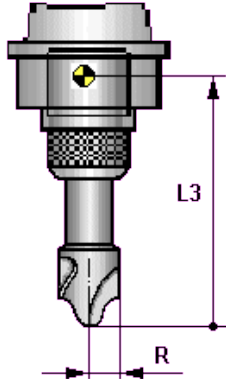
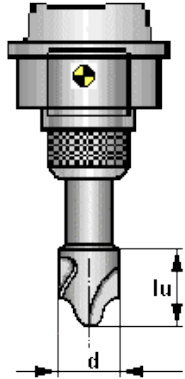
Technology		Milling	
Tool type name		Engraving tool	
Screen			
Tool type		2014	
Edge number		1	
Tool property:			Rec/Hd/BQ3
	○		Bit 5
Angle head tool	⊕	CH ²⁾	Bit 6
	○		Bit 15
Tool correction value:			Rec/UD/Ed/Geo/...
Length correction 1	● ¹⁾	L1 ¹⁾	L1
Length correction 2	● ¹⁾	L2 ¹⁾	L2
Length correction 3	●	L3	L3
Radius correction	●	R1	Rad
Edge position	● ¹⁾	O ¹⁾	Ori
Tool technology date:			Rec/UD/TI/Tech/...
Outer diameter	●	d	Tech1
	○		Tech2
	○		Tech3
	○		Tech4
Usable length	●	lu	Tech5
	○		Tech6
Apical angle	●	tac	Tech7
Number of teeth	●	nt	Tech8
Edge technology date:			Rec/UD/Ed/Tech/...
	○		Tech1
	○		Tech2

1) Active if CH = TRUE (angle head tool)

2) Value can be defined in the tool editor (⊕)

Tab. 10-38: 2014

Special milling tool

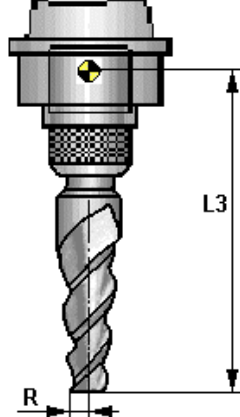
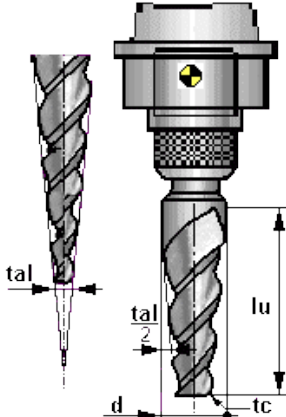
Technology		Milling	
Tool type name		Special milling tool	
Screen			
Tool type		2015	
Edge number		1	
Tool property:			Rec/Hd/BQ3
	☐		Bit 5
Angle head tool	☒	CH ²⁾	Bit 6
	☐		Bit 15
Tool correction value:			Rec/UD/Ed/Geo/...
Length correction 1	☒	L1 ¹⁾	L1
Length correction 2	☒	L2 ¹⁾	L2
Length correction 3	☐	L3	L3
Radius correction	☐	R1	Rad
Edge position	☒	O ¹⁾	Ori
Tool technology date:			Rec/UD/TI/Tech/...
Diameter	☐	d	Tech1
	☐		Tech2
	☐		Tech3
	☐		Tech4
Usable length	☐	lu	Tech5
	☐		Tech6
	☐		Tech7
Number of teeth	☐	nt	Tech8
Edge technology date:			Rec/UD/Ed/Tech/...
	☐		Tech1
	☐		Tech2

1) Active if CH = TRUE (angle head tool)

2) Value can be defined in the tool editor (©)

Tab. 10-39: 2015

Conical milling cutter

Technology		Milling	
Tool type name		Conical milling cutter	
Screen			
Tool type		2016	
Edge number		1	
Tool property:			Rec/Hd/BQ3
	○		Bit 5
Angle head tool	⊕	CH ²⁾	Bit 6
	○		Bit 15
Tool correction value:			Rec/UD/Ed/Geo/...
Length correction 1	● ¹⁾	L1 ¹⁾	L1
Length correction 2	● ¹⁾	L2 ¹⁾	L2
Length correction 3	●	L3	L3
Radius correction	●	R1	Rad
Edge position	● ¹⁾	O ¹⁾	Ori
Tool technology date:			Rec/UD/TI/Tech/...
Shaft diameter	●	d	Tech1
	○		Tech2
Sphere radius	●	tc	Tech3
	○		Tech4
Usable length	●	lu	Tech5
Opening angle	●	tal	Tech6
	○		Tech7
Number of teeth	●	nt	Tech8
Edge technology date:			Rec/UD/Ed/Tech/...
	○		Tech1
	○		Tech2

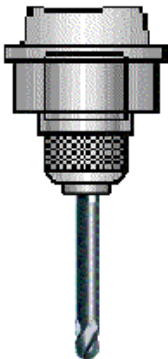
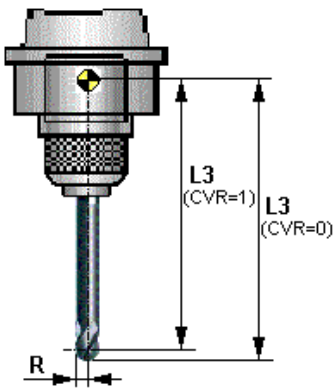
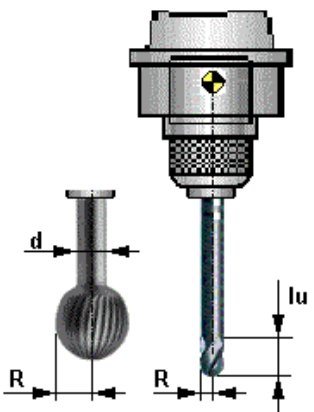
1) Active if CH = TRUE (angle head tool)

2) Value can be defined in the tool editor (⊕)

Tab. 10-40: 2016

Appendix

Ball cutter

Technology		Milling	
Tool type name		Ball cutter	
Screen			
Tool type		2017	
Edge number		1	
Tool property:			Rec/Hd/BQ3
Length reference of ball center	⊕	CVR ²⁾	Bit 5
Angle head tool	⊕	CH ²⁾	Bit 6
	○		Bit 15
Tool correction value:			Rec/UD/Ed/Geo/...
Length correction 1	● ¹⁾	L1 ¹⁾	L1
Length correction 2	● ¹⁾	L2 ¹⁾	L2
Length correction 3	●	L3	L3
Radius correction	●	R1	Rad
Edge position	● ¹⁾	O ¹⁾	Ori
Tool technology date:			Rec/UD/TI/Tech/...
Shaft diameter	●	d	Tech1
	○		Tech2
	○		Tech3
	○		Tech4
Usable length	●	lu	Tech5
	○		Tech6
	○		Tech7
Number of teeth	●	nt	Tech8
Edge technology date:			Rec/UD/Ed/Tech/...
	○		Tech1
	○		Tech2

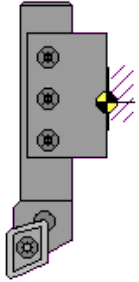
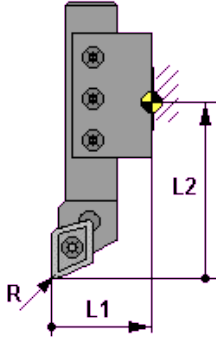
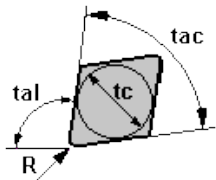
1) Active if CH = TRUE (angle head tool)

2) Value can be defined in the tool editor (⊕)

Tab. 10-41: 2017

Turning tools

Turning tool, general

Technology		Rotate	
Tool type name		Turning tool, general	
Screen			
Tool type		3000	
Edge number		can be set from 1 - max. edge number	
Tool property:			Rec/Hd/BQ3
	☐		Bit 5
Angle head tool	☒	CH ²⁾	Bit 6
Diameter input	☒	DIA	Bit 15
Tool correction value:			Rec/UD/Ed/Geo/...
Length correction 1	☒	L1	L1
Length correction 2	☒	L2	L2
Length correction 3	☒	L3 ¹⁾	L3
Radius correction	☒	R1	Rad
Edge position	☒	O	Ori
Tool technology date:			Rec/UD/TI/Tech/...
	☐		Tech1
	☐		Tech2
Tool tip size	☒	tc	Tech3
	☐		Tech4
	☐		Tech5
Tool setting angle	☒	tal	Tech6
Corner angle	☒	tac	Tech7
	☐		Tech8
Edge technology date:			Rec/UD/Ed/Tech/...
	☐		Tech1
	☐		Tech2

1)

Active if CH = TRUE (angle head tool)

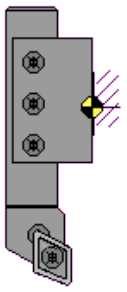
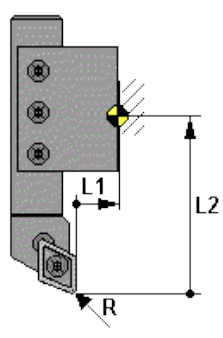
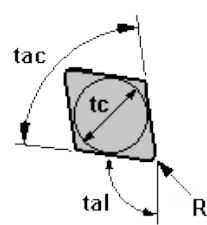
2)

Value can be defined in the tool editor (⊕)

Tab. 10-42:

3000

Turning tool, outside radial right

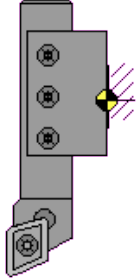
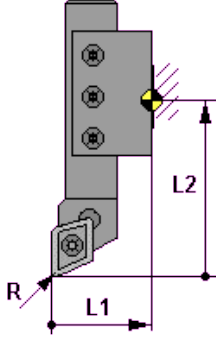
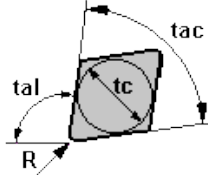
Technology		Rotate	
Tool type name		Turning tool, outside radial right	
Screen			
Tool type		3001	
Edge number		1	
Tool property:			Rec/Hd/BQ3
	☐		Bit 5
Angle head tool	☒	CH ²⁾	Bit 6
Diameter input	☒	DIA	Bit 15
Tool correction value:			Rec/UD/Ed/Geo/...
Length correction 1	☒	L1	L1
Length correction 2	☒	L2	L2
Length correction 3	☒	L3 ¹⁾	L3
Radius correction	☒	R1	Rad
Edge position	☒	O	Ori
Tool technology date:			Rec/UD/TI/Tech/...
	☐		Tech1
	☐		Tech2
Tool tip size	☒	tc	Tech3
	☐		Tech4
	☐		Tech5
Tool setting angle	☒	tal	Tech6
Corner angle	☒	tac	Tech7
	☐		Tech8
Edge technology date:			Rec/UD/Ed/Tech/...
	☐		Tech1
	☐		Tech2

1) Active if CH = TRUE (angle head tool)

2) Value can be defined in the tool editor (©)

Tab. 10-43: 3001

Turning tool, outside radial left

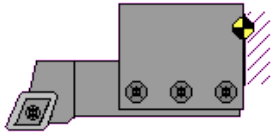
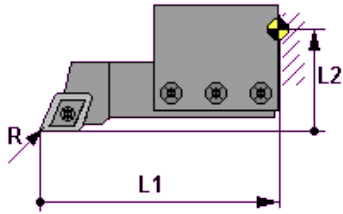
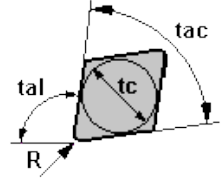
Technology		Rotate	
Tool type name		Turning tool, outside radial left	
Screen			
Tool type		3002	
Edge number		1	
Tool property:			Rec/Hd/BQ3
	☐		Bit 5
Angle head tool	☒	CH ²⁾	Bit 6
Diameter input	☒	DIA	Bit 15
Tool correction value:			Rec/UD/Ed/Geo/...
Length correction 1	☒	L1	L1
Length correction 2	☒	L2	L2
Length correction 3	☒	L3 ¹⁾	L3
Radius correction	☒	R1	Rad
Edge position	☒	O	Ori
Tool technology date:			Rec/UD/TI/Tech/...
	☐		Tech1
	☐		Tech2
Tool tip size	☒	tc	Tech3
	☐		Tech4
	☐		Tech5
Tool setting angle	☒	tal	Tech6
Corner angle	☒	tac	Tech7
	☐		Tech8
Edge technology date:			Rec/UD/Ed/Tech/...
	☐		Tech1
	☐		Tech2

1) Active if CH = TRUE (angle head tool)

2) Value can be defined in the tool editor (⊕)

Tab. 10-44: 3002

Turning tool, outside axial right

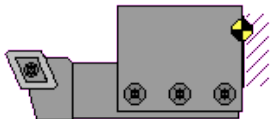
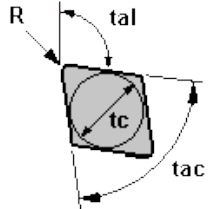
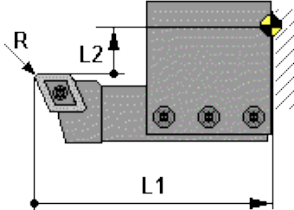
Technology		Rotate	
Tool type name		Turning tool, outside axial right	
Screen			
Tool type		3003	
Edge number		1	
Tool property:			Rec/Hd/BQ3
	☐		Bit 5
Angle head tool	☒	CH ²⁾	Bit 6
Diameter input	☒	DIA	Bit 15
Tool correction value:			Rec/UD/Ed/Geo/...
Length correction 1	☒	L1	L1
Length correction 2	☒	L2	L2
Length correction 3	☒	L3 ¹⁾	L3
Radius correction	☒	R1	Rad
Edge position	☒	O	Ori
Tool technology date:			Rec/UD/TI/Tech/...
	☐		Tech1
	☐		Tech2
Tool tip size	☒	tc	Tech3
	☐		Tech4
	☐		Tech5
Tool setting angle	☒	tal	Tech6
Corner angle	☒	tac	Tech7
	☐		Tech8
Edge technology date:			Rec/UD/Ed/Tech/...
	☐		Tech1
	☐		Tech2

1) Active if CH = TRUE (angle head tool)

2) Value can be defined in the tool editor (©)

Tab. 10-45: 3003

Turning tool, outside axial left

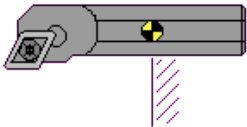
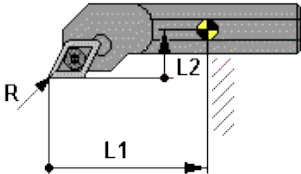
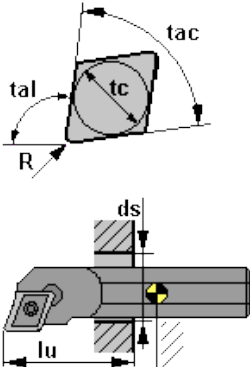
Technology		Rotate	
Tool type name		Turning tool, outside axial left	
Screen			
			
Tool type		3004	
Edge number		1	
Tool property:			Rec/Hd/BQ3
	☐		Bit 5
Angle head tool	☒	CH ²⁾	Bit 6
Diameter input	☒	DIA	Bit 15
Tool correction value:			Rec/UD/Ed/Geo/...
Length correction 1	☒	L1	L1
Length correction 2	☒	L2	L2
Length correction 3	☒	L3 ¹⁾	L3
Radius correction	☒	R1	Rad
Edge position	☒	O	Ori
Tool technology date:			Rec/UD/TI/Tech/...
	☐		Tech1
	☐		Tech2
Tool tip size	☒	tc	Tech3
	☐		Tech4
	☐		Tech5
Tool setting angle	☒	tal	Tech6
Corner angle	☒	tac	Tech7
	☐		Tech8
Edge technology date:			Rec/UD/Ed/Tech/...
	☐		Tech1
	☐		Tech2

1) Active if CH = TRUE (angle head tool)

2) Value can be defined in the tool editor (⊕)

Tab. 10-46: 3004

Turning tool, inside right

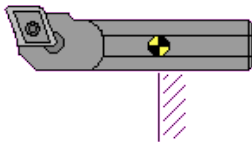
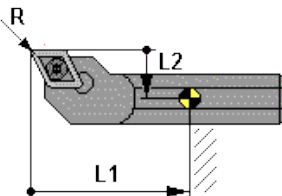
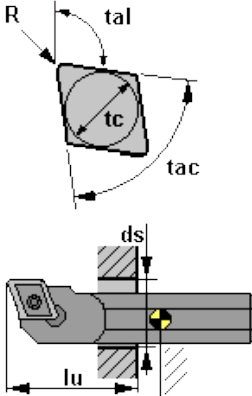
Technology		Rotate	
Tool type name		Turning tool, inside right	
Screen			
Tool type		3005	
Edge number		1	
Tool property:			Rec/Hd/BQ3
	○		Bit 5
Angle head tool	ⓔ	CH ²⁾	Bit 6
Diameter input	•	DIA	Bit 15
Tool correction value:			Rec/UD/Ed/Geo/...
Length correction 1	•	L1	L1
Length correction 2	•	L2	L2
Length correction 3	• ¹⁾	L3 ¹⁾	L3
Radius correction	•	R1	Rad
Edge position	•	O	Ori
Tool technology date:			Rec/UD/TI/Tech/...
	○		Tech1
Smallest diameter that can be machined	•	ds	Tech2
Tool tip size	•	tc	Tech3
	○		Tech4
Usable length	•	lu	Tech5
Tool setting angle	•	tal	Tech6
Corner angle	•	tac	Tech7
	○		Tech8
Edge technology date:			Rec/UD/Ed/Tech/...
	○		Tech1
	○		Tech2

1) Active if CH = TRUE (angle head tool)

2) Value can be defined in the tool editor (ⓔ)

Tab. 10-47: 3005

Turning tool, inside left

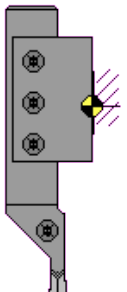
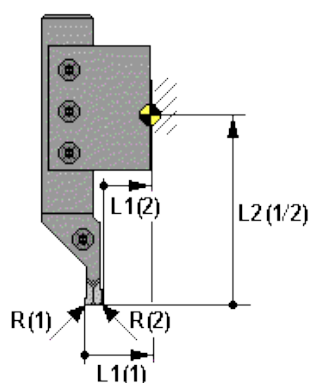
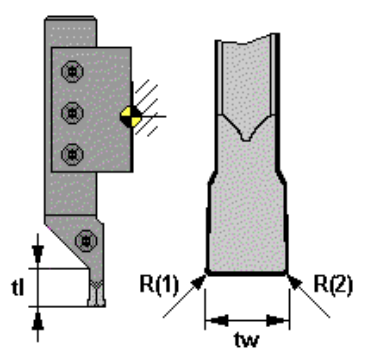
Technology		Rotate	
Tool type name		Turning tool, inside left	
Screen			
Tool type		3006	
Edge number		1	
Tool property:			Rec/Hd/BQ3
	○		Bit 5
Angle head tool	ⓔ	CH ²⁾	Bit 6
Diameter input	•	DIA	Bit 15
Tool correction value:			Rec/UD/Ed/Geo/...
Length correction 1	•	L1	L1
Length correction 2	•	L2	L2
Length correction 3	● ¹⁾	L3 ¹⁾	L3
Radius correction	•	R1	Rad
Edge position	•	O	Ori
Tool technology date:			Rec/UD/TI/Tech/...
	○		Tech1
Smallest diameter that can be machined	•	ds	Tech2
Tool tip size	•	tc	Tech3
	○		Tech4
Usable length	•	lu	Tech5
Tool setting angle	•	tal	Tech6
Corner angle	•	tac	Tech7
	○		Tech8
Edge technology date:			Rec/UD/Ed/Tech/...
	○		Tech1
	○		Tech2

1) Active if CH = TRUE (angle head tool)

2) Value can be defined in the tool editor (ⓔ)

Tab. 10-48: 3006

Plunging turn tool, outside right

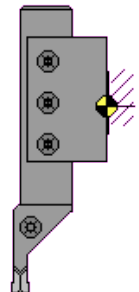
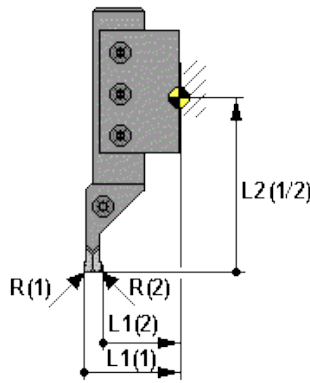
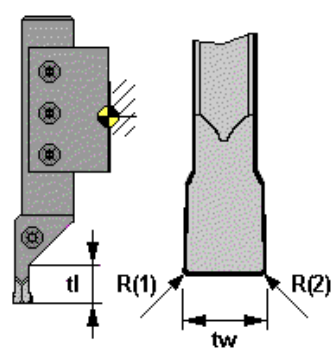
Technology		Rotate	
Tool type name		Plunging turn tool, outside right	
Screen			
Tool type		3007	
Edge number		2	
Tool property:			Rec/Hd/BQ3
	☐		Bit 5
Angle head tool	☒	CH ²⁾	Bit 6
Diameter input	☒	DIA	Bit 15
Tool correction value:			Rec/UD/Ed/Geo/...
Length correction 1	☒	L1	L1
Length correction 2	☒	L2	L2
Length correction 3	☒	L3 ¹⁾	L3
Radius correction	☒	R1	Rad
Edge position	☒	O	Ori
Tool technology date:			Rec/UD/TI/Tech/...
	☐		Tech1
	☐		Tech2
Cutting edge width	☒	tw	Tech3
Maximum grooving depth	☒	tl	Tech4
	☐		Tech5
	☐		Tech6
	☐		Tech7
	☐		Tech8
Edge technology date:			Rec/UD/Ed/Tech/...
	☐		Tech1
	☐		Tech2

1) Active if CH = TRUE (angle head tool)

2) Value can be defined in the tool editor (©)

Tab. 10-49: 3007

Plunging turn tool, outside left

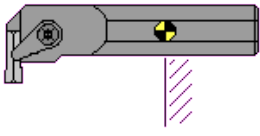
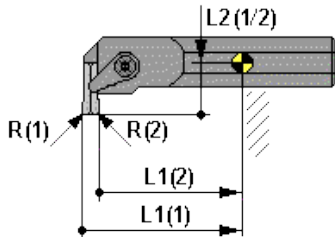
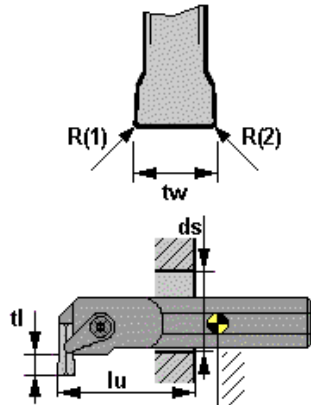
Technology		Rotate	
Tool type name		Plunging turn tool, outside left	
Screen			
Tool type		3008	
Edge number		2	
Tool property:			Rec/Hd/BQ3
	○		Bit 5
Angle head tool	⊕	CH ²⁾	Bit 6
Diameter input	•	DIA	Bit 15
Tool correction value:			Rec/UD/Ed/Geol/...
Length correction 1	•	L1	L1
Length correction 2	•	L2	L2
Length correction 3	● ¹⁾	L3 ¹⁾	L3
Radius correction	•	R1	Rad
Edge position	•	O	Ori
Tool technology date:			Rec/UD/TI/Tech/...
	○		Tech1
	○		Tech2
Cutting edge width	•	tw	Tech3
Maximum grooving depth	•	tl	Tech4
	○		Tech5
	○		Tech6
	○		Tech7
	○		Tech8
Edge technology date:			Rec/UD/Ed/Tech/...
	○		Tech1
	○		Tech2

1) Active if CH = TRUE (angle head tool)

2) Value can be defined in the tool editor (⊕)

Tab. 10-50: 3008

Plunging turn tool, inside right

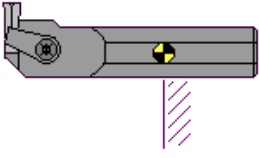
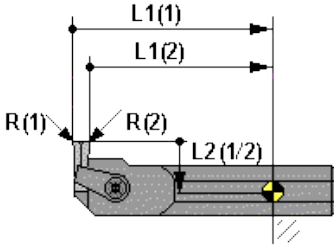
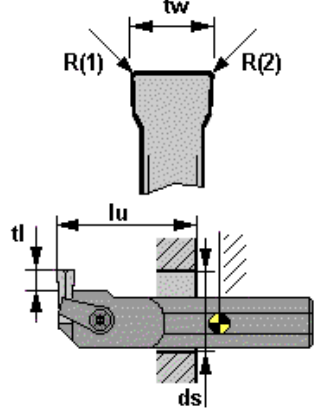
Technology		Rotate	
Tool type name		Plunging turn tool, inside right	
Screen			
Tool type		3009	
Edge number		2	
Tool property:			Rec/Hd/BQ3
	○		Bit 5
Angle head tool	ⓔ	CH ²⁾	Bit 6
Diameter input	•	DIA	Bit 15
Tool correction value:			Rec/UD/Ed/Geo/...
Length correction 1	•	L1	L1
Length correction 2	•	L2	L2
Length correction 3	• ¹⁾	L3 ¹⁾	L3
Radius correction	•	R1	Rad
Edge position	•	O	Ori
Tool technology date:			Rec/UD/TI/Tech/...
	○		Tech1
Smallest diameter that can be machined	•	ds	Tech2
Cutting edge width	•	tw	Tech3
Maximum grooving depth	•	tl	Tech4
Usable length	•	lu	Tech5
	○		Tech6
	○		Tech7
	○		Tech8
Edge technology date:			Rec/UD/Ed/Tech/...
	○		Tech1
	○		Tech2

1) Active if CH = TRUE (angle head tool)

2) Value can be defined in the tool editor (ⓔ)

Tab. 10-51: 3009

Plunging turn tool, inside left

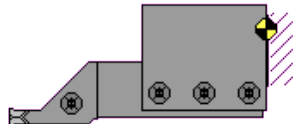
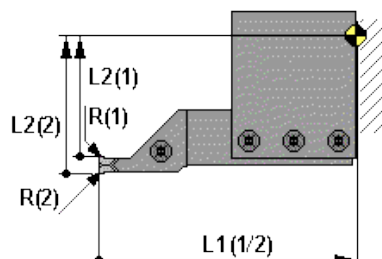
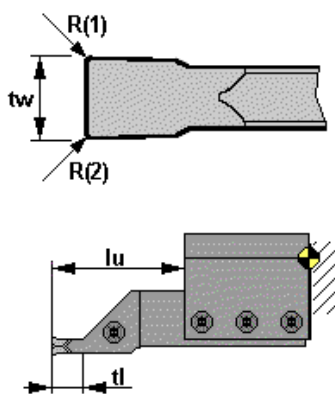
Technology		Rotate	
Tool type name		Plunging turn tool, inside left	
Screen			
Tool type		3010	
Edge number		2	
Tool property:			Rec/Hd/BQ3
	○		Bit 5
Angle head tool	ⓔ	CH ²⁾	Bit 6
Diameter input	•	DIA	Bit 15
Tool correction value:			Rec/UD/Ed/Geo/...
Length correction 1	•	L1	L1
Length correction 2	•	L2	L2
Length correction 3	● ¹⁾	L3 ¹⁾	L3
Radius correction	•	R1	Rad
Edge position	•	O	Ori
Tool technology date:			Rec/UD/TI/Tech/...
	○		Tech1
Smallest diameter that can be machined	•	ds	Tech2
Cutting edge width	•	tw	Tech3
Maximum grooving depth	•	tl	Tech4
Usable length	•	lu	Tech5
	○		Tech6
	○		Tech7
	○		Tech8
Edge technology date:			Rec/UD/Ed/Tech/...
	○		Tech1
	○		Tech2

1) Active if CH = TRUE (angle head tool)

2) Value can be defined in the tool editor (ⓔ)

Tab. 10-52: 3010

Plunging turn tool, axial right

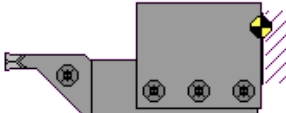
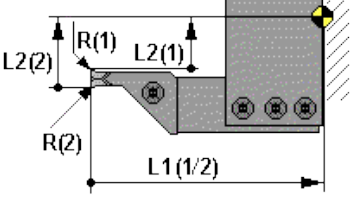
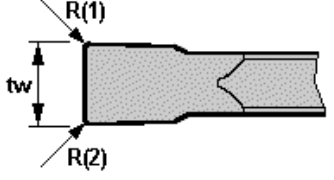
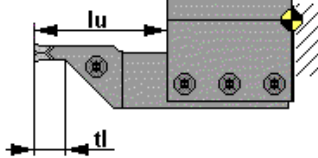
Technology		Rotate	
Tool type name		Plunging turn tool, axial right	
Screen			
Tool type		3011	
Edge number		2	
Tool property:			Rec/Hd/BQ3
	○		Bit 5
Angle head tool	⊕	CH ²⁾	Bit 6
Diameter input	●	DIA	Bit 15
Tool correction value:			Rec/UD/Ed/Geo/...
Length correction 1	●	L1	L1
Length correction 2	●	L2	L2
Length correction 3	● ¹⁾	L3 ¹⁾	L3
Radius correction	●	R1	Rad
Edge position	●	O	Ori
Tool technology date:			Rec/UD/TI/Tech/...
Outer limitation of the cutting diameter area	●	d	Tech1
Inner limitation of the cutting diameter area	●	ds	Tech2
Cutting edge width	●	tw	Tech3
Maximum grooving depth	●	tl	Tech4
Usable length	●	lu	Tech5
	○		Tech6
	○		Tech7
	○		Tech8
Edge technology date:			Rec/UD/Ed/Tech/...
	○		Tech1
	○		Tech2

1) Active if CH = TRUE (angle head tool)

2) Value can be defined in the tool editor (⊕)

Tab. 10-53: 3011

Plunging turn tool, axial left

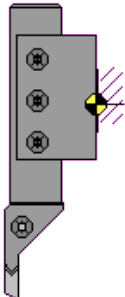
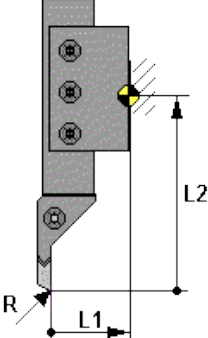
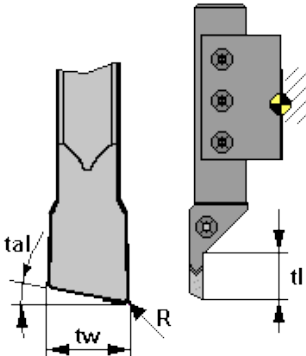
Technology		Rotate	
Tool type name		Plunging turn tool, axial left	
Screen			 
Tool type		3012	
Edge number		2	
Tool property:			Rec/Hd/BQ3
	○		Bit 5
Angle head tool	⊕	CH ²⁾	Bit 6
Diameter input	•	DIA	Bit 15
Tool correction value:			Rec/UD/Ed/Geo/...
Length correction 1	•	L1	L1
Length correction 2	•	L2	L2
Length correction 3	• ¹⁾	L3 ¹⁾	L3
Radius correction	•	R1	Rad
Edge position	•	O	Ori
Tool technology date:			Rec/UD/TI/Tech/...
Outer limitation of the cutting diameter area	•	d	Tech1
Inner limitation of the cutting diameter area	•	ds	Tech2
Cutting edge width	•	tw	Tech3
Maximum grooving depth	•	tl	Tech4
Usable length	•	lu	Tech5
	○		Tech6
	○		Tech7
	○		Tech8
Edge technology date:			Rec/UD/Ed/Tech/...
	○		Tech1
	○		Tech2

1) Active if CH = TRUE (angle head tool)

2) Value can be defined in the tool editor (⊕)

Tab. 10-54: 3012

Cut-off tool

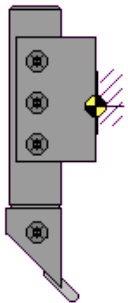
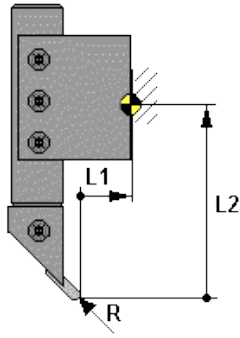
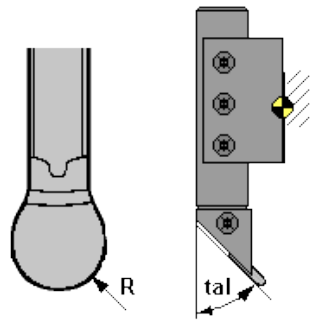
Technology		Rotate	
Tool type name		Cut-off tool	
Screen			
Tool type		3013	
Edge number		1	
Tool property:			Rec/Hd/BQ3
	○		Bit 5
Angle head tool	ⓔ	CH ²⁾	Bit 6
Diameter input	●	DIA	Bit 15
Tool correction value:			Rec/UD/Ed/Geo/...
Length correction 1	●	L1	L1
Length correction 2	●	L2	L2
Length correction 3	● ¹⁾	L3 ¹⁾	L3
Radius correction	●	R1	Rad
Edge position	●	O	Ori
Tool technology date:			Rec/UD/TI/Tech/...
	○		Tech1
	○		Tech2
Cutting edge width	●	tw	Tech3
Maximum grooving depth	●	tl	Tech4
	○		Tech5
Edge angle	●	tal	Tech6
	○		Tech7
	○		Tech8
Edge technology date:			Rec/UD/Ed/Tech/...
	○		Tech1
	○		Tech2

1) Active if CH = TRUE (angle head tool)

2) Value can be defined in the tool editor (ⓔ)

Tab. 10-55: 3013

Form turning tool, right

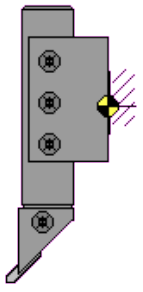
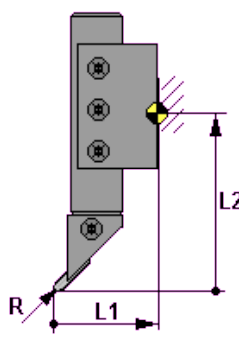
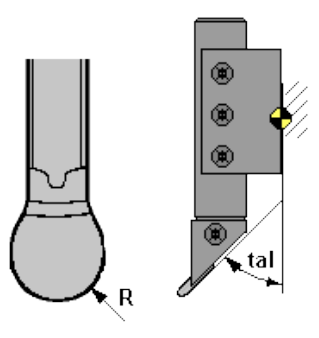
Technology		Rotate	
Tool type name		Form turning tool, right	
Screen			
Tool type		3014	
Edge number		1	
Tool property:			Rec/Hd/BQ3
	☐		Bit 5
Angle head tool	☒	CH ²⁾	Bit 6
Diameter input	☒	DIA	Bit 15
Tool correction value:			Rec/UD/Ed/Geo/...
Length correction 1	☒	L1	L1
Length correction 2	☒	L2	L2
Length correction 3	☒	L3 ¹⁾	L3
Radius correction	☒	R1	Rad
Edge position	☒	O	Ori
Tool technology date:			Rec/UD/TI/Tech/...
Outer limitation of the cutting diameter area	☒	d	Tech1
	☐		Tech2
	☐		Tech3
	☐		Tech4
	☐		Tech5
Orientation angle	☒	tal	Tech6
	☐		Tech7
	☐		Tech8
Edge technology date:			Rec/UD/Ed/Tech/...
	☐		Tech1
	☐		Tech2

1) Active if CH = TRUE (angle head tool)

2) Value can be defined in the tool editor (©)

Tab. 10-56: 3014

Form turning tool, left

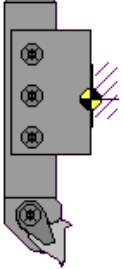
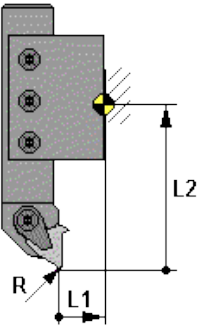
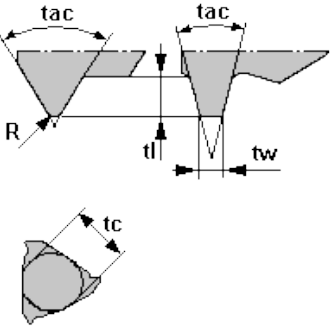
Technology		Rotate	
Tool type name		Form turning tool, left	
Screen			
Tool type		3015	
Edge number		1	
Tool property:			Rec/Hd/BQ3
	☐		Bit 5
Angle head tool	☒	CH ²⁾	Bit 6
Diameter input	☒	DIA	Bit 15
Tool correction value:			Rec/UD/Ed/Geo/...
Length correction 1	☒	L1	L1
Length correction 2	☒	L2	L2
Length correction 3	☒	L3 ¹⁾	L3
Radius correction	☒	R1	Rad
Edge position	☒	O	Ori
Tool technology date:			Rec/UD/TI/Tech/...
Outer limitation of the cutting diameter area	☒	d	Tech1
	☐		Tech2
	☐		Tech3
	☐		Tech4
	☐		Tech5
Orientation angle	☒	tal	Tech6
	☐		Tech7
	☐		Tech8
Edge technology date:			Rec/UD/Ed/Tech/...
	☐		Tech1
	☐		Tech2

1) Active if CH = TRUE (angle head tool)

2) Value can be defined in the tool editor (©)

Tab. 10-57: 3015

Thread turning tool, outside right

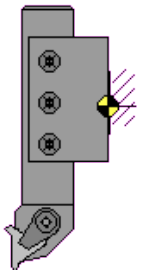
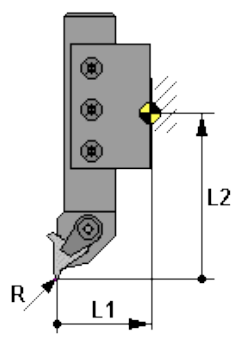
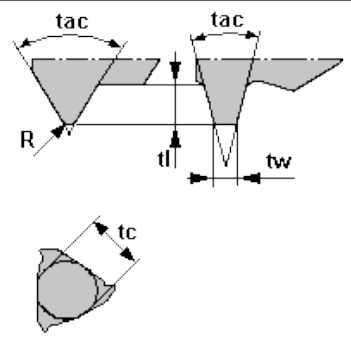
Technology		Rotate	
Tool type name		Thread turning tool, outside right	
Screen			
Tool type		3016	
Edge number		1	
Tool property:			Rec/Hd/BQ3
	○		Bit 5
Angle head tool	ⓔ	CH ²⁾	Bit 6
Diameter input	●	DIA	Bit 15
Tool correction value:			Rec/UD/Ed/Geo/...
Length correction 1	●	L1	L1
Length correction 2	●	L2	L2
Length correction 3	● ¹⁾	L3 ¹⁾	L3
Radius correction	●	R1	Rad
Edge position	●	O	Ori
Tool technology date:			Rec/UD/TI/Tech/...
Width of trapezoid thread	●	tw	Tech1
	○		Tech2
Tool tip size	●	tc	Tech3
Maximum thread depth	●	tl	Tech4
	○		Tech5
Thread pitch	●	tp	Tech6
Edge angle	●	tac	Tech7
	○		Tech8
Edge technology date:			Rec/UD/Ed/Tech/...
	○		Tech1
	○		Tech2

1) Active if CH = TRUE (angle head tool)

2) Value can be defined in the tool editor (ⓔ)

Tab. 10-58: 3016

Thread turning tool, outside left

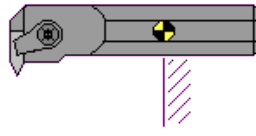
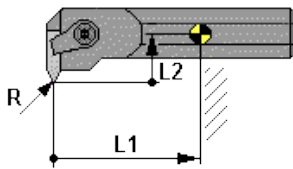
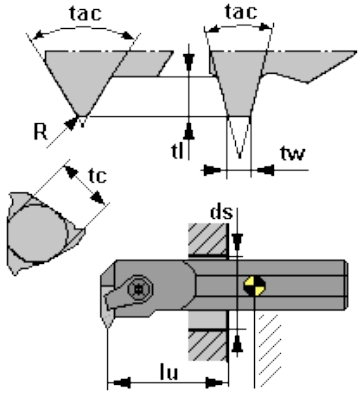
Technology		Rotate	
Tool type name		Thread turning tool, outside left	
Screen			
Tool type		3017	
Edge number		1	
Tool property:			Rec/Hd/BQ3
	○		Bit 5
Angle head tool	⊕	CH ²⁾	Bit 6
Diameter input	●	DIA	Bit 15
Tool correction value:			Rec/UD/Ed/Geo/...
Length correction 1	●	L1	L1
Length correction 2	●	L2	L2
Length correction 3	● ¹⁾	L3 ¹⁾	L3
Radius correction	●	R1	Rad
Edge position	●	O	Ori
Tool technology date:			Rec/UD/TI/Tech/...
Width of trapezoid thread	●	tw	Tech1
	○		Tech2
Tool tip size	●	tc	Tech3
Maximum thread depth	●	tl	Tech4
	○		Tech5
Thread pitch	●	tp	Tech6
Edge angle	●	tac	Tech7
	○		Tech8
Edge technology date:			Rec/UD/Ed/Tech/...
	○		Tech1
	○		Tech2

1) Active if CH = TRUE (angle head tool)

2) Value can be defined in the tool editor (⊕)

Tab. 10-59: 3017

Thread turning tool, inside right

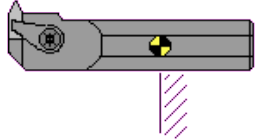
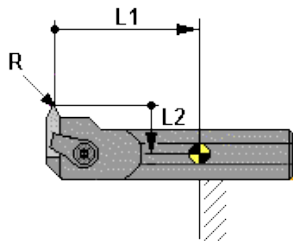
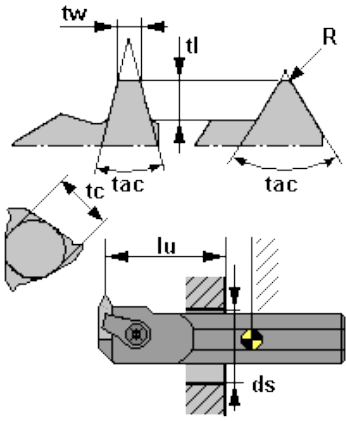
Technology		Rotate	
Tool type name		Thread turning tool, inside right	
Screen			
Tool type		3018	
Edge number		1	
Tool property:			Rec/Hd/BQ3
	○		Bit 5
Angle head tool	⊕	CH ²⁾	Bit 6
Diameter input	•	DIA	Bit 15
Tool correction value:			Rec/UD/Ed/Geo/...
Length correction 1	•	L1	L1
Length correction 2	•	L2	L2
Length correction 3	● ¹⁾	L3 ¹⁾	L3
Radius correction	•	R1	Rad
Edge position	•	O	Ori
Tool technology date:			Rec/UD/TI/Tech/...
Width of trapezoid thread	•	tw	Tech1
Smallest diameter that can be machined	•	ds	Tech2
Tool tip size	•	tc	Tech3
Maximum thread depth	•	tl	Tech4
Usable length	•	lu	Tech5
Thread pitch	•	tp	Tech6
Edge angle	•	tac	Tech7
	○		Tech8
Edge technology date:			Rec/UD/Ed/Tech/...
	○		Tech1
	○		Tech2

1) Active if CH = TRUE (angle head tool)

2) Value can be defined in the tool editor (⊕)

Tab. 10-60: 3018

Thread turning tool, inside left

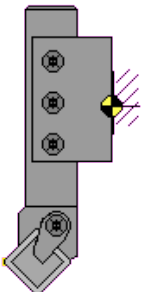
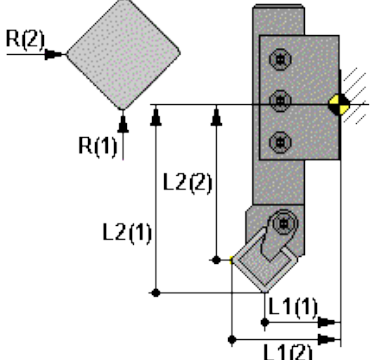
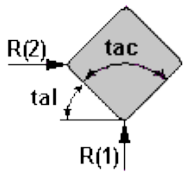
Technology		Rotate	
Tool type name		Thread turning tool, inside left	
Screen			
Tool type		3019	
Edge number		1	
Tool property:			Rec/Hd/BQ3
	○		Bit 5
Angle head tool	ⓔ	CH ²⁾	Bit 6
Diameter input	•	DIA	Bit 15
Tool correction value:			Rec/UD/Ed/Geo/...
Length correction 1	•	L1	L1
Length correction 2	•	L2	L2
Length correction 3	• ¹⁾	L3 ¹⁾	L3
Radius correction	•	R1	Rad
Edge position	•	O	Ori
Tool technology date:			Rec/UD/TI/Tech/...
Width of trapezoid thread	•	tw	Tech1
Smallest diameter that can be machined	•	ds	Tech2
Tool tip size	•	tc	Tech3
Maximum thread depth	•	tl	Tech4
Usable length	•	lu	Tech5
Thread pitch	•	tp	Tech6
Edge angle	•	tac	Tech7
	○		Tech8
Edge technology date:			Rec/UD/Ed/Tech/...
	○		Tech1
	○		Tech2

1) Active if CH = TRUE (angle head tool)

2) Value can be defined in the tool editor (ⓔ)

Tab. 10-61: 3019

Special turning tool

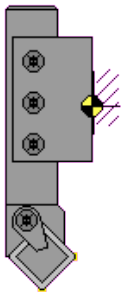
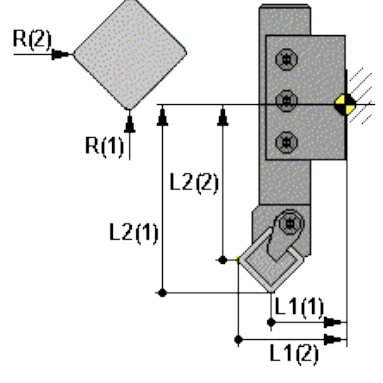
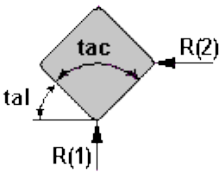
Technology		Rotate	
Tool type name		Special turning tool	
Screen			
Tool type		3020	
Edge number		can be set from 1 - max. edge number	
Tool property:			Rec/Hd/BQ3
	☐		Bit 5
Angle head tool	☒	CH ²⁾	Bit 6
Diameter input	☒	DIA	Bit 15
Tool correction value:			Rec/UD/Ed/Geo/...
Length correction 1	☒	L1	L1
Length correction 2	☒	L2	L2
Length correction 3	☒	L3 ¹⁾	L3
Radius correction	☒	R1	Rad
Edge position	☒	O	Ori
Tool technology date:			Rec/UD/TI/Tech/...
	☐		Tech1
	☐		Tech2
	☐		Tech3
	☐		Tech4
	☐		Tech5
Tool setting angle	☒	tal	Tech6
Corner angle	☒	tac	Tech7
	☐		Tech8
Edge technology date:			Rec/UD/Ed/Tech/...
	☐		Tech1
	☐		Tech2

1) Active if CH = TRUE (angle head tool)

2) Value can be defined in the tool editor (Ⓢ)

Tab. 10-62: 3020

Turning tool, outside radial right (2 edges)

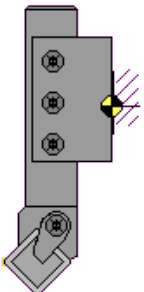
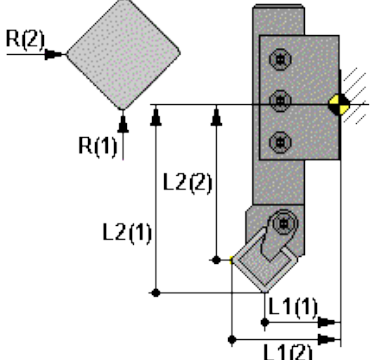
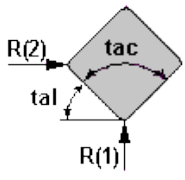
Technology		Rotate	
Tool type name		Turning tool, outside radial right (2S)	
Screen			
Tool type		3021	
Edge number		2	
Tool property:			Rec/Hd/BQ3
	☐		Bit 5
Angle head tool	☒	CH ²⁾	Bit 6
Diameter input	☒	DIA	Bit 15
Tool correction value:			Rec/UD/Ed/Geo/...
Length correction 1	☒	L1	L1
Length correction 2	☒	L2	L2
Length correction 3	☒	L3 ¹⁾	L3
Radius correction	☒	R1	Rad
Edge position	☒	O	Ori
Tool technology date:			Rec/UD/TI/Tech/...
	☐		Tech1
	☐		Tech2
	☐		Tech3
	☐		Tech4
	☐		Tech5
Tool setting angle	☒	tal	Tech6
Corner angle	☒	tac	Tech7
	☐		Tech8
Edge technology date:			Rec/UD/Ed/Tech/...
	☐		Tech1
	☐		Tech2

1) Active if CH = TRUE (angle head tool)

2) Value can be defined in the tool editor (©)

Tab. 10-63: 3021

Turning tool, outside radial left (2 edges)

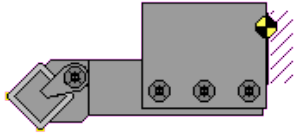
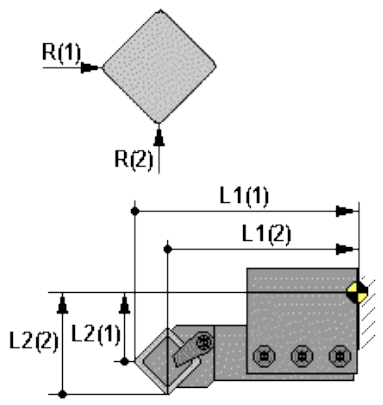
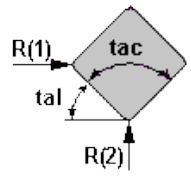
Technology		Rotate	
Tool type name		Turning tool, outside radial left (2S)	
Screen			
Tool type		3022	
Edge number		2	
Tool property:			Rec/Hd/BQ3
	○		Bit 5
Angle head tool	ⓔ	CH ²⁾	Bit 6
Diameter input	•	DIA	Bit 15
Tool correction value:			Rec/UD/Ed/Geo/...
Length correction 1	•	L1	L1
Length correction 2	•	L2	L2
Length correction 3	● ¹⁾	L3 ¹⁾	L3
Radius correction	•	R1	Rad
Edge position	•	O	Ori
Tool technology date:			Rec/UD/TI/Tech/...
	○		Tech1
	○		Tech2
	○		Tech3
	○		Tech4
	○		Tech5
Tool setting angle	•	tal	Tech6
Corner angle	•	tac	Tech7
	○		Tech8
Edge technology date:			Rec/UD/Ed/Tech/...
	○		Tech1
	○		Tech2

1) Active if CH = TRUE (angle head tool)

2) Value can be defined in the tool editor (ⓔ)

Tab. 10-64: 3022

Turning tool, outside axial right (2 edges)

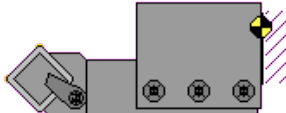
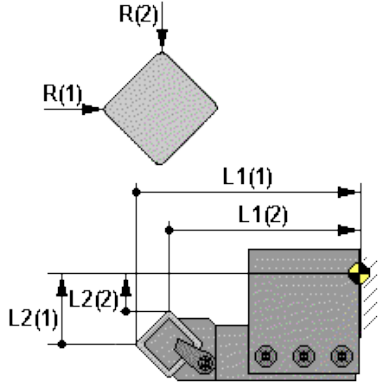
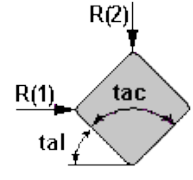
Technology		Rotate	
Tool type name		Turning tool, outside axial right (2S)	
Screen			
Tool type		3023	
Edge number		2	
Tool property:			Rec/Hd/BQ3
	☐		Bit 5
Angle head tool	☒	CH ²⁾	Bit 6
Diameter input	☒	DIA	Bit 15
Tool correction value:			Rec/UD/Ed/Geo/...
Length correction 1	☒	L1	L1
Length correction 2	☒	L2	L2
Length correction 3	☒	L3 ¹⁾	L3
Radius correction	☒	R1	Rad
Edge position	☒	O	Ori
Tool technology date:			Rec/UD/TI/Tech/...
	☐		Tech1
	☐		Tech2
	☐		Tech3
	☐		Tech4
	☐		Tech5
Tool setting angle	☒	tal	Tech6
Corner angle	☒	tac	Tech7
	☐		Tech8
Edge technology date:			Rec/UD/Ed/Tech/...
	☐		Tech1
	☐		Tech2

1) Active if CH = TRUE (angle head tool)

2) Value can be defined in the tool editor (©)

Tab. 10-65: 3023

Turning tool, outside axial left (2 edges)

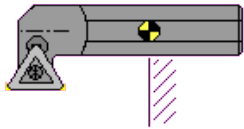
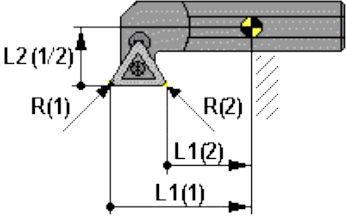
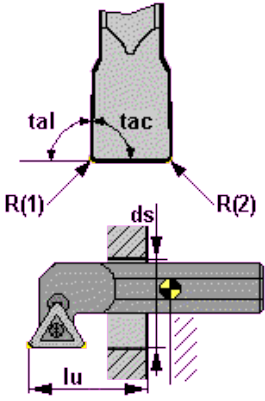
Technology		Rotate	
Tool type name		Turning tool, outside axial left (2S)	
Screen			
Tool type		3024	
Edge number		2	
Tool property:			Rec/Hd/BQ3
	☐		Bit 5
Angle head tool	☒	CH ²⁾	Bit 6
Diameter input	☐	DIA	Bit 15
Tool correction value:			Rec/UD/Ed/Geo/...
Length correction 1	☐	L1	L1
Length correction 2	☐	L2	L2
Length correction 3	☒	L3 ¹⁾	L3
Radius correction	☐	R1	Rad
Edge position	☐	O	Ori
Tool technology date:			Rec/UD/TI/Tech/...
	☐		Tech1
	☐		Tech2
	☐		Tech3
	☐		Tech4
	☐		Tech5
Tool setting angle	☐	tal	Tech6
Corner angle	☐	tac	Tech7
	☐		Tech8
Edge technology date:			Rec/UD/Ed/Tech/...
	☐		Tech1
	☐		Tech2

1) Active if CH = TRUE (angle head tool)

2) Value can be defined in the tool editor (Ⓢ)

Tab. 10-66: 3024

Turning tool, inside right (2 edges)

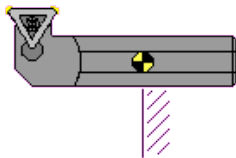
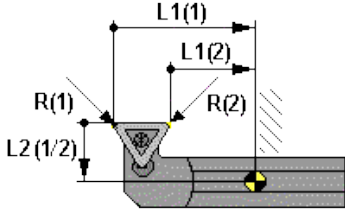
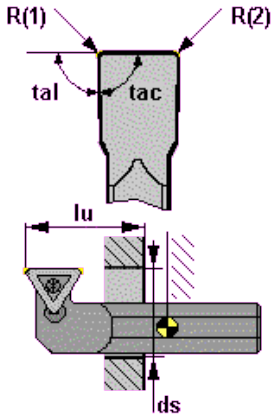
Technology		Rotate	
Tool type name		Turning tool, inside right (2S)	
Screen			
Tool type		3025	
Edge number		2	
Tool property:			Rec/Hd/BQ3
	☐		Bit 5
Angle head tool	☒	CH ²⁾	Bit 6
Diameter input	☒	DIA	Bit 15
Tool correction value:			Rec/UD/Ed/Geo/...
Length correction 1	☒	L1	L1
Length correction 2	☒	L2	L2
Length correction 3	☒	L3 ¹⁾	L3
Radius correction	☒	R1	Rad
Edge position	☒	O	Ori
Tool technology date:			Rec/UD/TI/Tech/...
	☐		Tech1
Smallest diameter that can be machined	☒	ds	Tech2
	☐		Tech3
	☐		Tech4
Usable length	☒	lu	Tech5
Tool setting angle	☒	tal	Tech6
Corner angle	☒	tac	Tech7
	☐		Tech8
Edge technology date:			Rec/UD/Ed/Tech/...
	☐		Tech1
	☐		Tech2

1) Active if CH = TRUE (angle head tool)

2) Value can be defined in the tool editor (©)

Tab. 10-67: 3025

Turning tool, inside left (2 edges)

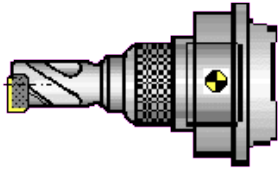
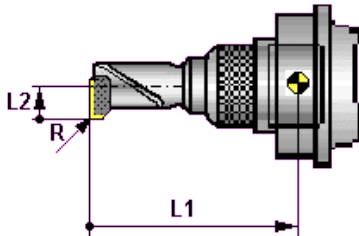
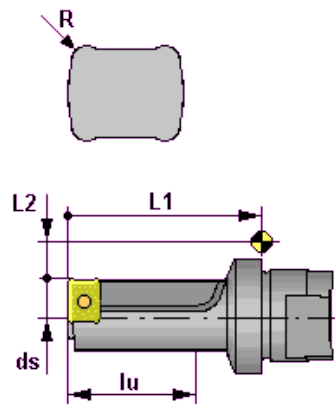
Technology		Rotate	
Tool type name		Turning tool, inside left (2S)	
Screen			
Tool type		3026	
Edge number		2	
Tool property:			Rec/Hd/BQ3
	☐		Bit 5
Angle head tool	☒	CH ²⁾	Bit 6
Diameter input	☒	DIA	Bit 15
Tool correction value:			Rec/UD/Ed/Geo/...
Length correction 1	☒	L1	L1
Length correction 2	☒	L2	L2
Length correction 3	☒	L3 ¹⁾	L3
Radius correction	☒	R1	Rad
Edge position	☒	O	Ori
Tool technology date:			Rec/UD/TI/Tech/...
	☐		Tech1
Smallest diameter that can be machined	☒	ds	Tech2
	☐		Tech3
	☐		Tech4
Usable length	☒	lu	Tech5
Tool setting angle	☒	tal	Tech6
Corner angle	☒	tac	Tech7
	☐		Tech8
Edge technology date:			Rec/UD/Ed/Tech/...
	☐		Tech1
	☐		Tech2

1) Active if CH = TRUE (angle head tool)

2) Value can be defined in the tool editor (Ⓢ)

Tab. 10-68: 3026

EcoCut

Technology		Rotate	
Tool type name		EcoCut	
Screen			
Tool type		3027	
Edge number		2	
Tool property:			Rec/Hd/BQ3
	☐		Bit 5
Angle head tool	☒	CH ²⁾	Bit 6
Diameter input	☒	DIA	Bit 15
Tool correction value:			Rec/UD/Ed/Geo/...
Length correction 1	☒	L1	L1
Length correction 2	☒	L2	L2
Length correction 3	☒	L3 ¹⁾	L3
Radius correction	☒	R1	Rad
Edge position	☒	O	Ori
Tool technology date:			Rec/UD/TI/Tech/...
	☐		Tech1
Drilling diameter	☒	ds	Tech2
	☐		Tech3
	☐		Tech4
Usable length	☒	lu	Tech5
	☐		Tech6
	☐		Tech7
	☐		Tech8
Edge technology date:			Rec/UD/Ed/Tech/...
	☐		Tech1
	☐		Tech2

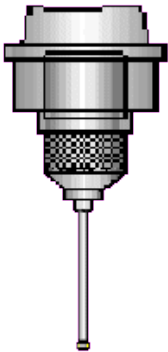
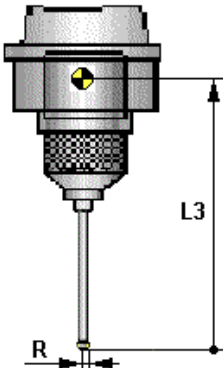
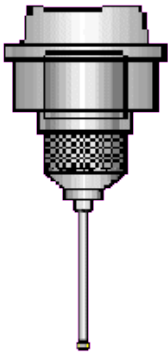
1) Active if CH = TRUE (angle head tool)

2) Value can be defined in the tool editor (©)

Tab. 10-69: 3027

Measuring probe

Probe, general

Technology		Measurement	
Tool type name		Probe, general	
Screen			
Tool type		5,000	
Edge number		1	
Tool property:			Rec/Hd/BQ3
	☐		Bit 5
	☐		Bit 6
	☐		Bit 15
Tool correction value:			Rec/UD/Ed/Geo/...
	☐		L1
	☐		L2
Length correction 3	☒	L3	L3
Radius correction	☒	R1	Rad
	☐		Ori
Tool technology date:			Rec/UD/TI/Tech/...
	☐		Tech1
	☐		Tech2
	☐		Tech3
	☐		Tech4
	☐		Tech5
	☐		Tech6
	☐		Tech7
	☐		Tech8
Edge technology date:			Rec/UD/Ed/Tech/...
	☐		Tech1
	☐		Tech2

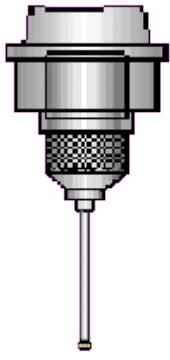
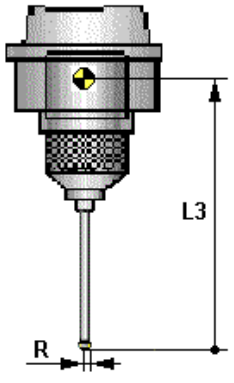
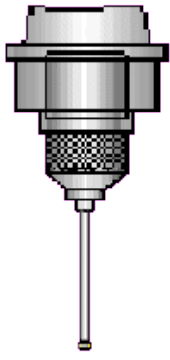
1) Active if CH = TRUE (angle head tool)

2) Value can be defined in the tool editor (©)

Tab. 10-70: 5,000

Appendix

Mono probe

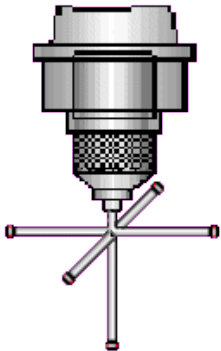
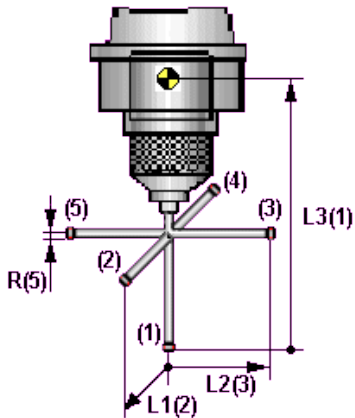
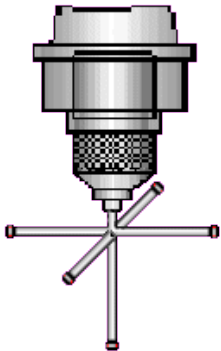
Technology		Measurement	
Tool type name		Mono probe	
Screen			
	Tool type		
	5001		
	Edge number		
Tool property:		Rec/Hd/BQ3	
		☐	Bit 5
		☐	Bit 6
		☐	Bit 15
Tool correction value:		Rec/UD/Ed/Geo/...	
		☐	L1
		☐	L2
Length correction 3	☒	L3	L3
Radius correction	☒	R1	Rad
		☐	Ori
Tool technology date:		Rec/UD/TI/Tech/...	
		☐	Tech1
		☐	Tech2
		☐	Tech3
		☐	Tech4
		☐	Tech5
		☐	Tech6
		☐	Tech7
		☐	Tech8
Edge technology date:		Rec/UD/Ed/Tech/...	
		☐	Tech1
		☐	Tech2

1) Active if CH = TRUE (angle head tool)

2) Value can be defined in the tool editor (©)

Tab. 10-71: 5001

Spring-return joystick switch

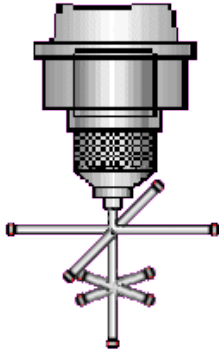
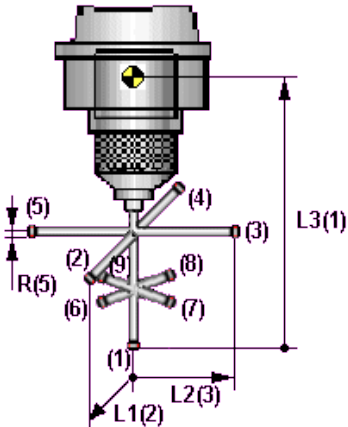
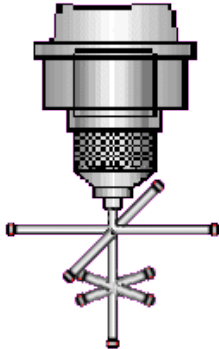
Technology		Measurement	
Tool type name		Spring-return joystick switch	
Screen			
Tool type		5002	
Edge number		5	
Tool property:			Rec/Hd/BQ3
			Bit 5
			Bit 6
			Bit 15
Tool correction value:			Rec/UD/Ed/Geo/...
Length correction 1	•	L1	L1
Length correction 2	•	L2	L2
Length correction 3	•	L3	L3
Radius correction	•	R1	Rad
			Ori
Tool technology date:			Rec/UD/TI/Tech/...
			Tech1
			Tech2
			Tech3
			Tech4
			Tech5
			Tech6
			Tech7
			Tech8
Edge technology date:			Rec/UD/Ed/Tech/...
			Tech1
			Tech2

1) Active if CH = TRUE (angle head tool)

2) Value can be defined in the tool editor (⊕)

Tab. 10-72: 5002

Special probe

Technology		Measurement	
Tool type name		Special probe	
Screen			
Tool type		5003	
Edge number		can be set from 1 - max. edge number	
Tool property:			Rec/Hd/BQ3
	☐		Bit 5
	☐		Bit 6
	☐		Bit 15
Tool correction value:			Rec/UD/Ed/Geo/...
Length correction 1	☒	L1	L1
Length correction 2	☒	L2	L2
Length correction 3	☒	L3	L3
Radius correction	☒	R1	Rad
	☐		Ori
Tool technology date:			Rec/UD/TI/Tech/...
	☐		Tech1
	☐		Tech2
	☐		Tech3
	☐		Tech4
	☐		Tech5
	☐		Tech6
	☐		Tech7
	☐		Tech8
Edge technology date:			Rec/UD/Ed/Tech/...
	☐		Tech1
	☐		Tech2

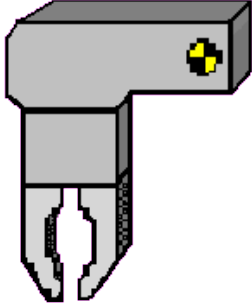
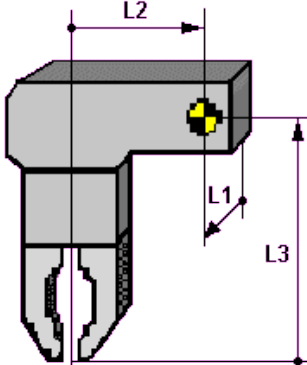
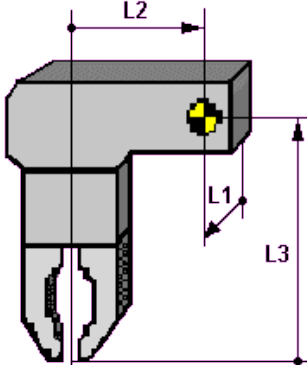
1) Active if CH = TRUE (angle head tool)

2) Value can be defined in the tool editor (©)

Tab. 10-73: 5003

Special tools

Grippers

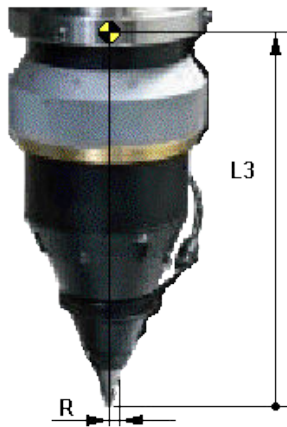
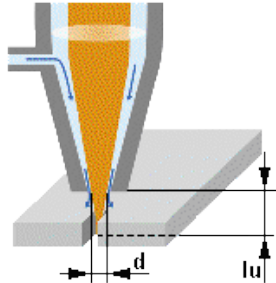
Technology		Special processing	
Tool type name		Grippers	
Screen			
Tool type		6000	
Edge number		can be set from 1 - max. edge number	
Tool property:			Rec/Hd/BQ3
	☐		Bit 5
	☐		Bit 6
	☐		Bit 15
Tool correction value:			Rec/UD/Ed/Geo/...
Length correction 1	☒	L1	L1
Length correction 2	☒	L2	L2
Length correction 3	☒	L3	L3
	☐		Rad
	☐		Ori
Tool technology date:			Rec/UD/TI/Tech/...
	☐		Tech1
	☐		Tech2
	☐		Tech3
	☐		Tech4
	☐		Tech5
	☐		Tech6
	☐		Tech7
	☐		Tech8
Edge technology date:			Rec/UD/Ed/Tech/...
	☐		Tech1
	☐		Tech2

1) Active if CH = TRUE (angle head tool)

2) Value can be defined in the tool editor (©)

Tab. 10-74: 6000

Jet cutting tool

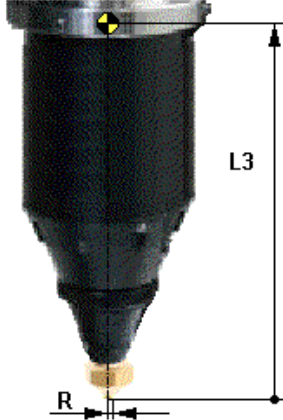
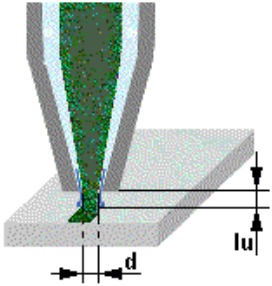
Technology		Special processing	
Tool type name		Jet cutting tool	
Screen			
Tool type		6001	
Edge number		1	
Tool property:			Rec/Hd/BQ3
	☐		Bit 5
	☐		Bit 6
	☐		Bit 15
Tool correction value:			Rec/UD/Ed/Geo/...
	☐		L1
	☐		L2
Length correction 3	☒	L3	L3
Radius correction	☒	R1	Rad
	☐		Ori
Tool technology date:			Rec/UD/TI/Tech/...
Diameter dimension 1	☒	d	Tech1
	☐		Tech2
	☐		Tech3
	☐		Tech4
Length dimension	☒	lu	Tech5
	☐		Tech6
	☐		Tech7
	☐		Tech8
Edge technology date:			Rec/UD/Ed/Tech/...
	☐		Tech1
	☐		Tech2

1) Active if CH = TRUE (angle head tool)

2) Value can be defined in the tool editor (©)

Tab. 10-75: 6001

AMT

Technology		Special processing	
Tool type name		AMT	
Screen			
Tool type		6002	
Edge number		1	
Tool property:			Rec/Hd/BQ3
	☐		Bit 5
	☐		Bit 6
	☐		Bit 15
Tool correction value:			Rec/UD/Ed/Geo/...
	☐		L1
	☐		L2
Length correction 3	☒	L3	L3
Radius correction	☒	R1	Rad
	☐		Ori
Tool technology date:			Rec/UD/TI/Tech/...
Diameter dimension 1	☒	d	Tech1
	☐		Tech2
	☐		Tech3
	☐		Tech4
Length dimension	☒	lu	Tech5
	☐		Tech6
	☐		Tech7
	☐		Tech8
Edge technology date:			Rec/UD/Ed/Tech/...
	☐		Tech1
	☐		Tech2

1)

Active if CH = TRUE (angle head tool)

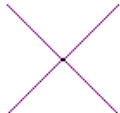
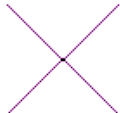
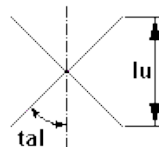
2)

Value can be defined in the tool editor (©)

Tab. 10-76: 6002

Punching and forming tool

Punching tool, general

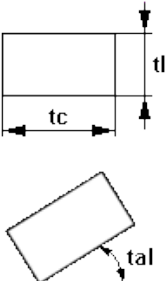
Technology		Punching	
Tool type name		Punching tool, general	
Screen			
Tool type		7000	
Edge number		1	
Tool property:			Rec/Hd/BQ3
	☐		Bit 5
	☐		Bit 6
	☐		Bit 15
Tool correction value:			Rec/UD/Ed/Geo/...
	☐		L1
	☐		L2
	☐		L3
Radius correction	☒	R1	Rad
	☐		Ori
Tool technology date:			Rec/UD/TI/Tech/...
	☐		Tech1
	☐		Tech2
	☐		Tech3
	☐		Tech4
Length dimension	☒	lu	Tech5
Angular dimension	☒	tal	Tech6
	☐		Tech7
	☐		Tech8
Edge technology date:			Rec/UD/Ed/Tech/...
	☐		Tech1
	☐		Tech2

1) Active if CH = TRUE (angle head tool)

2) Value can be defined in the tool editor (©)

Tab. 10-77: 7000

Rectangular stamp

Technology		Punching	
Tool type name		Rectangular stamp	
Screen			
Tool type		7001	
Edge number		1	
Tool property:			Rec/Hd/BQ3
	☐		Bit 5
	☐		Bit 6
	☐		Bit 15
Tool correction value:			Rec/UD/Ed/Geo/...
	☐		L1
	☐		L2
	☐		L3
	☐		Rad
	☐		Ori
Tool technology date:			Rec/UD/TI/Tech/...
	☐		Tech1
	☐		Tech2
Width dimension	☒	tc	Tech3
Depth gauge	☒	tl	Tech4
	☐		Tech5
Angular dimension	☒	tal	Tech6
	☐		Tech7
	☐		Tech8
Edge technology date:			Rec/UD/Ed/Tech/...
	☐		Tech1
	☐		Tech2

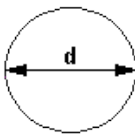
1) Active if CH = TRUE (angle head tool)

2) Value can be defined in the tool editor (Ⓢ)

Tab. 10-78: 7001

Appendix

Circular stamp

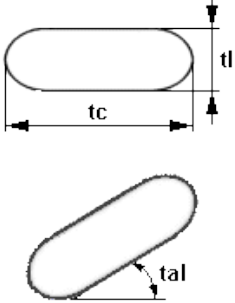
Technology		Punching	
Tool type name		Circular stamp	
Screen			
Tool type		7002	
Edge number		1	
Tool property:			Rec/Hd/BQ3
	☐		Bit 5
	☐		Bit 6
	☐		Bit 15
Tool correction value:			Rec/UD/Ed/Geo/...
	☐		L1
	☐		L2
	☐		L3
Radius correction	☒	R1	Rad
	☐		Ori
Tool technology date:			Rec/UD/TI/Tech/...
Diameter dimension	☒	d	Tech1
	☐		Tech2
	☐		Tech3
	☐		Tech4
	☐		Tech5
	☐		Tech6
	☐		Tech7
	☐		Tech8
Edge technology date:			Rec/UD/Ed/Tech/...
	☐		Tech1
	☐		Tech2

1) Active if CH = TRUE (angle head tool)

2) Value can be defined in the tool editor (©)

Tab. 10-79: 7002

Elongated hole stamp

Technology		Punching	
Tool type name		Elongated hole stamp	
Screen			
Tool type		7003	
Edge number		1	
Tool property:			Rec/Hd/BQ3
	☐		Bit 5
	☐		Bit 6
	☐		Bit 15
Tool correction value:			Rec/UD/Ed/Geo/...
	☐		L1
	☐		L2
	☐		L3
	☐		Rad
	☐		Ori
Tool technology date:			Rec/UD/TI/Tech/...
	☐		Tech1
	☐		Tech2
Width dimension	☒	tc	Tech3
Depth gauge	☒	tl	Tech4
	☐		Tech5
Angular dimension	☒	tal	Tech6
	☐		Tech7
	☐		Tech8
Edge technology date:			Rec/UD/Ed/Tech/...
	☐		Tech1
	☐		Tech2

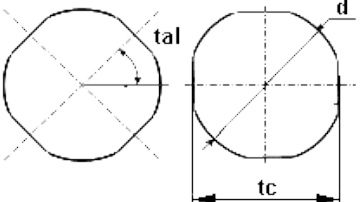
1) Active if CH = TRUE (angle head tool)

2) Value can be defined in the tool editor (⊕)

Tab. 10-80: 7003

Appendix

Circle fourfold flattened

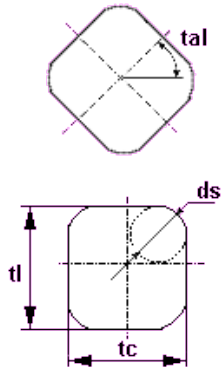
Technology		Punching	
Tool type name		Circle fourfold flattened	
Screen			
Tool type		7004	
Edge number		1	
Tool property:			Rec/Hd/BQ3
	☐		Bit 5
	☐		Bit 6
	☐		Bit 15
Tool correction value:			Rec/UD/Ed/Geo/...
	☐		L1
	☐		L2
	☐		L3
	☐		Rad
	☐		Ori
Tool technology date:			Rec/UD/TI/Tech/...
Diameter dimension	☒	d	Tech1
	☐		Tech2
Width dimension	☒	tc	Tech3
	☐		Tech4
	☐		Tech5
Angular dimension	☒	tal	Tech6
	☐		Tech7
	☐		Tech8
Edge technology date:			Rec/UD/Ed/Tech/...
	☐		Tech1
	☐		Tech2

1) Active if CH = TRUE (angle head tool)

2) Value can be defined in the tool editor (©)

Tab. 10-81: 7004

Square, rounded

Technology		Punching	
Tool type name		Rectangle, rounded	
Screen			
Tool type		7005	
Edge number		1	
Tool property:			Rec/Hd/BQ3
	☐		Bit 5
	☐		Bit 6
	☐		Bit 15
Tool correction value:			Rec/UD/Ed/Geo/...
	☐		L1
	☐		L2
	☐		L3
	☐		Rad
	☐		Ori
Tool technology date:			Rec/UD/TI/Tech/...
	☐		Tech1
Diameter dimension	☒	ds	Tech2
Width dimension	☒	tc	Tech3
Depth gauge	☒	tl	Tech4
	☐		Tech5
Angular dimension	☒	tal	Tech6
	☐		Tech7
	☐		Tech8
Edge technology date:			Rec/UD/Ed/Tech/...
	☐		Tech1
	☐		Tech2

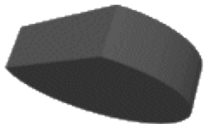
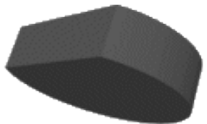
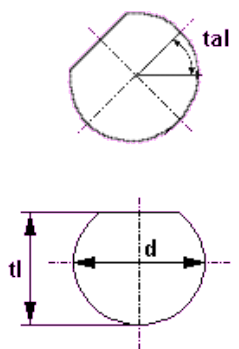
1) Active if CH = TRUE (angle head tool)

2) Value can be defined in the tool editor (☺)

Tab. 10-82: 7005

Appendix

Circle single flattened

Technology		Punching	
Tool type name		Circle single flattened	
Screen			
Tool type		7006	
Edge number		1	
Tool property:			Rec/Hd/BQ3
	☐		Bit 5
	☐		Bit 6
	☐		Bit 15
Tool correction value:			Rec/UD/Ed/Geo/...
	☐		L1
	☐		L2
	☐		L3
	☐		Rad
	☐		Ori
Tool technology date:			Rec/UD/TI/Tech/...
Diameter dimension	☒	d	Tech1
	☐		Tech2
	☐		Tech3
Depth gauge	☒	tl	Tech4
	☐		Tech5
Angular dimension	☒	tal	Tech6
	☐		Tech7
	☐		Tech8
Edge technology date:			Rec/UD/Ed/Tech/...
	☐		Tech1
	☐		Tech2

1)

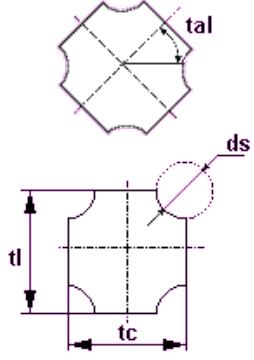
Active if CH = TRUE (angle head tool)

2)

Value can be defined in the tool editor (©)

Tab. 10-83: 7006

Square, fourfold circular slit

Technology		Punching	
Tool type name		Rectangle, fourfold circular slit	
Screen			
Tool type		7007	
Edge number		1	
Tool property:			Rec/Hd/BQ3
	☐		Bit 5
	☐		Bit 6
	☐		Bit 15
Tool correction value:			Rec/UD/Ed/Geo/...
	☐		L1
	☐		L2
	☐		L3
	☐		Rad
	☐		Ori
Tool technology date:			Rec/UD/TI/Tech/...
	☐		Tech1
Diameter dimension	☒	ds	Tech2
Width dimension	☒	tc	Tech3
Depth gauge	☒	tl	Tech4
	☐		Tech5
Angular dimension	☒	tal	Tech6
	☐		Tech7
	☐		Tech8
Edge technology date:			Rec/UD/Ed/Tech/...
	☐		Tech1
	☐		Tech2

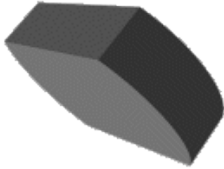
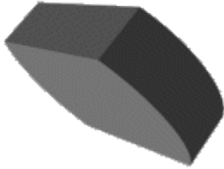
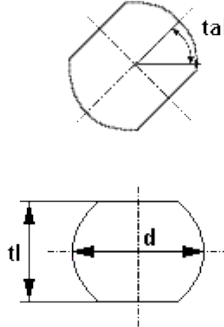
1) Active if CH = TRUE (angle head tool)

2) Value can be defined in the tool editor (☺)

Tab. 10-84: 7007

Appendix

Circle double flattened

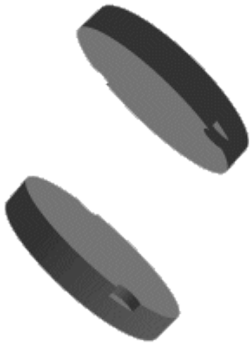
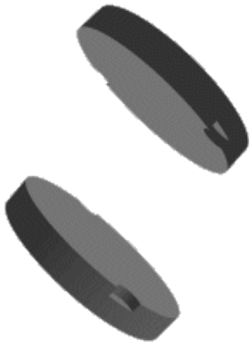
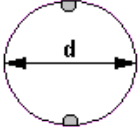
Technology		Punching	
Tool type name		Circle double flattened	
Screen			
Tool type		7008	
Edge number		1	
Tool property:			Rec/Hd/BQ3
	☐		Bit 5
	☐		Bit 6
	☐		Bit 15
Tool correction value:			Rec/UD/Ed/Geo/...
	☐		L1
	☐		L2
	☐		L3
	☐		Rad
	☐		Ori
Tool technology date:			Rec/UD/TI/Tech/...
Diameter dimension	☒	d	Tech1
	☐		Tech2
	☐		Tech3
Depth gauge	☒	tl	Tech4
	☐		Tech5
Angular dimension	☒	tal	Tech6
	☐		Tech7
	☐		Tech8
Edge technology date:			Rec/UD/Ed/Tech/...
	☐		Tech1
	☐		Tech2

1) Active if CH = TRUE (angle head tool)

2) Value can be defined in the tool editor (©)

Tab. 10-85: 7008

Printing plate

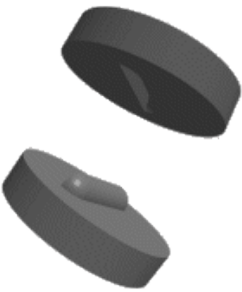
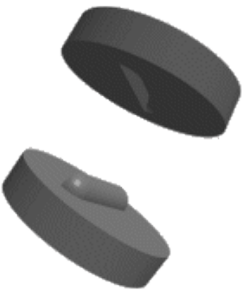
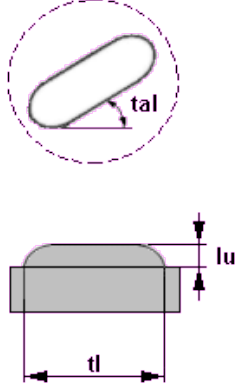
Technology		Punching	
Tool type name		Printing plate	
Screen			
Tool type		7009	
Edge number		1	
Tool property:			Rec/Hd/BQ3
	☐		Bit 5
	☐		Bit 6
	☐		Bit 15
Tool correction value:			Rec/UD/Ed/Geo/...
	☐		L1
	☐		L2
	☐		L3
	☐		Rad
	☐		Ori
Tool technology date:			Rec/UD/TI/Tech/...
Diameter dimension	☒	d	Tech1
	☐		Tech2
	☐		Tech3
	☐		Tech4
	☐		Tech5
	☐		Tech6
	☐		Tech7
	☐		Tech8
Edge technology date:			Rec/UD/Ed/Tech/...
	☐		Tech1
	☐		Tech2

1) Active if CH = TRUE (angle head tool)

2) Value can be defined in the tool editor (⊕)

Tab. 10-86: 7009

Ventilation slot

Technology		Punching	
Tool type name		Ventilation slot	
Screen			
Tool type		7010	
Edge number		1	
Tool property:			Rec/Hd/BQ3
	☐		Bit 5
	☐		Bit 6
	☐		Bit 15
Tool correction value:			Rec/UD/Ed/Geo/...
	☐		L1
	☐		L2
	☐		L3
Radius correction	☒	R1	Rad
	☐		Ori
Tool technology date:			Rec/UD/TI/Tech/...
	☐		Tech1
	☐		Tech2
	☐		Tech3
Depth gauge	☒	tl	Tech4
Length dimension	☒	lu	Tech5
Angular dimension	☒	tal	Tech6
	☐		Tech7
	☐		Tech8
Edge technology date:			Rec/UD/Ed/Tech/...
	☐		Tech1
	☐		Tech2

1)

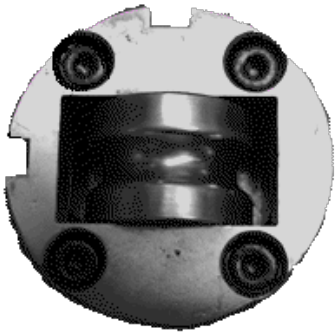
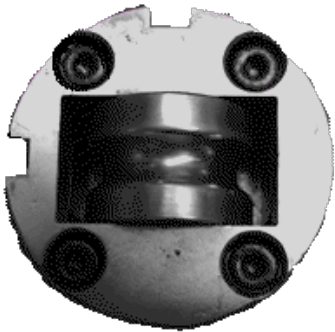
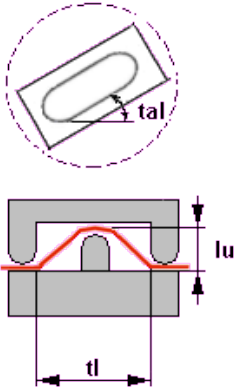
Active if CH = TRUE (angle head tool)

2)

Value can be defined in the tool editor (©)

Tab. 10-87: 7010

Rolling attachment

Technology		Reshaping	
Tool type name		Rolling attachment	
Screen			
Tool type		7011	
Edge number		1	
Tool property:			Rec/Hd/BQ3
	☐		Bit 5
	☐		Bit 6
	☐		Bit 15
Tool correction value:			Rec/UD/Ed/Geo/...
	☐		L1
	☐		L2
	☐		L3
	☐		Rad
	☐		Ori
Tool technology date:			Rec/UD/TI/Tech/...
	☐		Tech1
	☐		Tech2
	☐		Tech3
Depth gauge	☒	tl	Tech4
Length dimension	☒	lu	Tech5
Angular dimension	☒	tal	Tech6
	☐		Tech7
	☐		Tech8
Edge technology date:			Rec/UD/Ed/Tech/...
	☐		Tech1
	☐		Tech2

1)

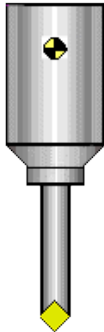
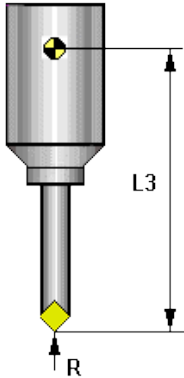
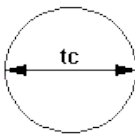
Active if CH = TRUE (angle head tool)

2)

Value can be defined in the tool editor (⊕)

Tab. 10-88: 7011

Signing tool

Technology		Reshaping	
Tool type name		Signing tool	
Screen			
Tool type		7012	
Edge number		1	
Tool property:			Rec/Hd/BQ3
	☐		Bit 5
	☐		Bit 6
	☐		Bit 15
Tool correction value:			Rec/UD/Ed/Geo/...
	☐		L1
	☐		L2
Length correction 3	☒	L3	L3
Radius correction	☒	R1	Rad
	☐		Ori
Tool technology date:			Rec/UD/TI/Tech/...
	☐		Tech1
	☐		Tech2
Width dimension	☒	tc	Tech3
	☐		Tech4
	☐		Tech5
	☐		Tech6
	☐		Tech7
	☐		Tech8
Edge technology date:			Rec/UD/Ed/Tech/...
	☐		Tech1
	☐		Tech2

1)

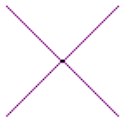
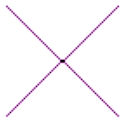
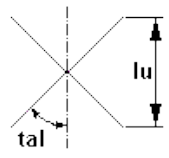
Active if CH = TRUE (angle head tool)

2)

Value can be defined in the tool editor (©)

Tab. 10-89: 7012

Forming tool, general

Technology		Reshaping	
Tool type name		Forming tool, general	
Screen			
Tool type		7013	
Edge number		1	
Tool property:			Rec/Hd/BQ3
	☐		Bit 5
	☐		Bit 6
	☐		Bit 15
Tool correction value:			Rec/UD/Ed/Geo/...
	☐		L1
	☐		L2
	☐		L3
	☐		Rad
	☐		Ori
Tool technology date:			Rec/UD/TI/Tech/...
	☐		Tech1
	☐		Tech2
	☐		Tech3
	☐		Tech4
Length dimension	☒	lu	Tech5
Angular dimension	☒	tal	Tech6
	☐		Tech7
	☐		Tech8
Edge technology date:			Rec/UD/Ed/Tech/...
	☐		Tech1
	☐		Tech2

1) Active if CH = TRUE (angle head tool)

2) Value can be defined in the tool editor (©)

Tab. 10-90: 7013

Appendix

Key

Element	Techn. label	Description	Type dependency of the meaning
Rec/Hd/BQ3 (bit 5)	CVR	Length reference of ball center	Ball cutter
Rec/Hd/BQ3 (bit 6)	CH	Angle head tool (all corrections applied)	Drilling, milling or turning tools
Rec/Hd/BQ3 (bit 15)	DIA	Tool with diameter input (for L2)	Turning tools
Rec/UD/Ed/Geo/L1	L1	1st length correction value	-
Rec/UD/Ed/Geo/L2	L2	2nd length correction value	-
Rec/UD/Ed/Geo/L3	L3	3rd length correction value	-
Rec/UD/Ed/Geo/Rad	R1	Large radius	-
Rec/UD/Ed/Geo/Ori	O	Edge position	-
TI/Tech/Tech1	d	Diameter	Drilling and milling tools
		Large diameter	
		Milling diameter	
		Outer diameter	
		Outer limitation of the cutting diameter area	Turning tools
		Edge radius	
		Diameter dimension 1	User tool
Rec/UD/TI/Tech/Tech2	ds	Small diameter	Drilling tools
		Shaft diameter	
		Drilling diameter	
		Smallest diameter that can be machined	Turning tools
		Inner limitation of the cutting diameter area	
		Diameter dimension 2	User tool
Rec/UD/TI/Tech/Tech3	lfc	Chamfer length	Drilling and milling tools
	ll	Loss length	
	tw	Cutting edge width	Various turning tools
		Width of trapezoid thread	
	tc	Corner radius	Various milling tools
		Tool tip size	Various turning tools
		Width dimension	User tool
Rec/UD/TI/Tech/Tech4	tl	Max. cutting height	Turning tools
		Max. thread height	
		Depth gauge	User tool

Element	Techn. label	Description	Type dependency of the meaning
Rec/UD/TI/Tech/ Tech5	lu	Usable length	-
		Length dimension	User tool
Rec/UD/TI/Tech/ Tech6	tp	(Thread) lead	Drilling, milling or turning tools
	ta	Setting angle	Div. drilling tools
	tal	Orientation angle	Various turning tools
		Angle (edge angle)	Various milling or turning tools
		Tool setting angle	Various turning tools
		Angle (kappa)	Various milling tools
		Angle	
		Angular dimension 1	User tool
Rec/UD/TI/Tech/ Tech7	tac	Apical angle	Drilling and milling tools
		Countersink angle	
		Corner angle	Turning tools
		Edge angle	
		Angular dimension 2	User tool
Rec/UD/TI/Tech/ Tech8	nt	Number of teeth	Milling tools and user tool
Rec/UD/Ed/Tech/ Tech1	de	Diameter (edge-dependent)	Step drill
		Edge dimension 1	User tool
Rec/UD/Ed/Tech/ Tech2	lfe	Cut length	Step drill
		Edge dimension 2	User tool

Tab. 10-91: Tool catalog key

10.3.3 Bitmap libraries

The following bitmap libraries are contained when delivered:

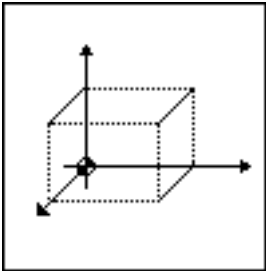
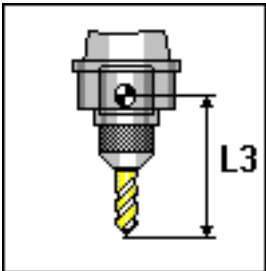
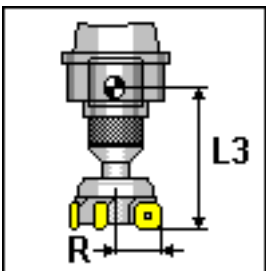
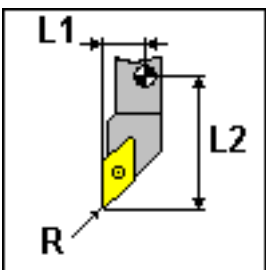
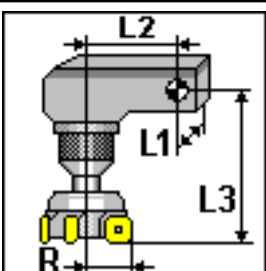
BitmapResourceName	Bitmap name (value)	Example (original size)	Explanation
ToolStorageBmp.resx	spindle16_g.gif		Spindle
	slot16_g.gif		Turret
	magazin16_g.gif		Magazine
	grip16_g.gif		Grippers
	schränk.gif		Tool cabinet

Tab. 10-92: Bitmap library of tool memory

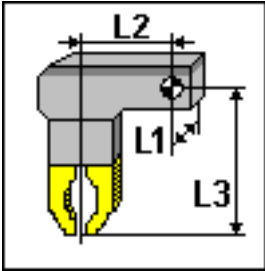
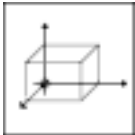
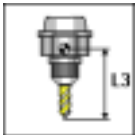
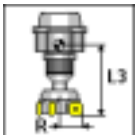
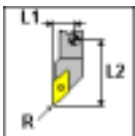
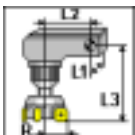
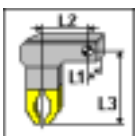
Appendix

BitmapResourceName	Bitmap name (value)	Example (original size)	Explanation
ToolTecTypesBmp.resx	drill16_g.gif		Drilling
	mill16_g.gif		Milling
	turn16_g.gif		Rotate
	grind16_g.gif		Grinding
	cut16_g.gif		Tool edges
	form16_g.gif		Reshaping
	erode16_g.gif		Eroding
	nibble16_g.gif		Nibbling
	punch16_g.gif		Punching
	screw16_g.gif		Screwing
	weld16_g.gif		Welding
	wrench16_g.gif		-
	mtctds_TOOL_g.gif		-
	special16_g.gif		-
	tool16_g.gif		-

Tab. 10-93: Bitmap library of tool technology

BitmapResourceName	Bitmap name (value)	Example (original size)	Explanation
ToolCorrTypeBmp.resx	CorrType_0.gif		Correction type 0 (General tool type)
	CorrType_1.gif		Correction type 1 (Drilling tools)
	CorrType_2.gif		Correction type 2 (Milling tools)
	CorrType_3.gif		Correction type 3 (Turning tools)
	CorrType_4.gif		Correction type 4 (Special milling tools)

Appendix

BitmapResourceName	Bitmap name (value)	Example (original size)	Explanation
	CorrType_5.gif		Correction type 5 (Special tools)
	CorrType_0_k.gif		Correction type 0 (General tool type)
	CorrType_1_k.gif		Correction type 1 (Drilling tools)
	CorrType_2_k.gif		Correction type 2 (Milling tools)
	CorrType_3_k.gif		Correction type 3 (Turning tools)
	CorrType_4_k.gif		Correction type 4 (Special milling tools)
	CorrType_5_k.gif		Correction type 5 (Special tools)
	CorrType_0_kk.gif		Correction type 0 (General tool type)
	CorrType_1_kk.gif		Correction type 1 (Drilling tools)
	CorrType_2_kk.gif		Correction type 2 (Milling tools)
	CorrType_3_kk.gif		Correction type 3 (Turning tools)
	CorrType_4_kk.gif		Correction type 4 (Special milling tools)

BitmapResourceName	Bitmap name (value)	Example (original size)	Explanation
	CorrType_5_kk.gif		Correction type 5 (Special tools)
	Kt1k.gif		Correction type 1 (Drilling tools)
	Kt2k.gif		Correction type 2 (Milling tools)
	Kt3k.gif		Correction type 3 (Turning tools)
	Kt4k.gif		Correction type 4 (Special milling tools)
	Kt5k.gif		Correction type 5 (Special tools)

Tab. 10-94: Bitmap library of tool correction types

BitmapResourceName	Bitmap name (value)	Example (original size)	Explanation
ToolStatusBmp.resx	checkbox_false.gif		
	checkbox_true.gif		
	hookred.gif		
	hookwhite.gif		
	ledgreen.gif		
	ledred.gif		
	ledyellow.gif		
	stateundefined16.gif		Undefined
	statealarm16.gif		Alarm
	statedisabled16.gif		Inactive
	stateok16.gif		OK

Tab. 10-95: Bitmap library of status messages



In addition, there are three more bitmap libraries (see [tab. 7-30 "Provided bitmap libraries" on page 273](#)) to display tool types in different sizes.

11 Service and support

Our worldwide service network provides an optimized and efficient support. Our experts offer you advice and assistance should you have any queries. You can contact us **24/7**.

Service Germany Our technology-oriented Competence Center in Lohr, Germany, is responsible for all your service-related queries for electric drive and controls.

Contact the **Service Hotline** and **Service Helpdesk** under:

Phone: **+49 9352 40 5060**
Fax: **+49 9352 18 4941**
E-mail: service.svc@boschrexroth.de
Internet: <http://www.boschrexroth.com>

Additional information on service, repair (e.g. delivery addresses) and training can be found on our internet sites.

Service worldwide Outside Germany, please contact your local service office first. For hotline numbers, refer to the sales office addresses on the internet.

Preparing information To be able to help you more quickly and efficiently, please have the following information ready:

- Detailed description of malfunction and circumstances
- Type plate specifications of the affected products, in particular type codes and serial numbers
- Your contact data (phone and fax number as well as your e-mail address)

Index

Symbols

[default Produktname] XM.....	29
//%C%.....	109
//%CHBEGIN%.....	104
//%CHEND%.....	104
//%DCS%.....	119
//%GF%.....	108
//%GROUP%.....	106
//%HELPPFILE%.....	109
//%HELPTOKEN%.....	109
//%LANG%.....	105
//%N%.....	120
//%NAME%.....	105
//%P%.....	109
//%V%.....	110
//%VAR%.....	120
%@%.....	110
%A1%.....	110, 119
%D%.....	114
%GF%.....	116
%P%.....	120
%SIGN%.....	105
%VALID%.....	117

A

About this documentation.....	1
Validity of the documentation.....	1
Access protection	
Control data.....	50
Project planning.....	49
ACI principle.....	282
ACI screens.....	76
Activate second database table	204
Activating the SD variable.....	333
Active channel number.....	137
Address parameter.....	110
Address parameters.....	100
Adjusting the search condition.....	331
Advanced archiving	
Archive advanced data.....	59
ALT.....	118
Archive advanced data.....	59
User input.....	59
Archive and restore.....	53
Archive control data.....	56
User input.....	56
Archive extended data	
Restore extended data.....	60
Archive/restore data	
Archive control data.....	56
Control data.....	56
Drive data.....	61
IndraWorks project.....	53
Restore control data.....	58
Axis commissioning.....	358

Commissioning tools.....	358
Control commissioning.....	376
Drive commissioning.....	371
General axis commissioning.....	375
Offline parameterization.....	390
PLC commissioning.....	389
Sercos fast startup.....	391

B

Backup battery.....	155
Basic data management.....	255
Initialize tool data block during tool re-start.....	256
Basic data type collection.....	224, 443
Basic projects.....	48
Battery.....	20
BOOL.....	112
BQ1 - Place status.....	459
BQ2 - Tool status.....	458
BQ3 - Technology status bits.....	460

C

Call, components.....	98
Calling TCH jobs.....	319
Calling the tool change.....	310
Calls, multiple lines.....	99
Calls, single line.....	99
Channel dialog.....	95
Channel overview.....	95
Commission.....	173
NC block sequences.....	95
Parameter settings for cycle calls.....	173
SD variables.....	174
Subroutines.....	174
Use existing projects.....	174
Commissioning	
F-keys for input assistance.....	123
Commissioning a simple tool management	
Commissioning.....	320
Functional description of the tool change....	307
Operating interface.....	304
Providing the NC subroutines.....	333
SD setup variables.....	329
Settings: Optional calls.....	336
Commissioning M-keys.....	70
Commissioning the simple tool management	
Configuring the database.....	321
Commissioning the tool management with standard tool change.....	303
Commissioning tool	
IW-Drive - Sercos master.....	362
Commissioning tools.....	92, 358
NC configurator - machine parameters.....	358
NC Editor - SCS files.....	362
Communication parameter	

Index

Properties.....	77	Commissioning the tool management with standard tool change.....	303
Communication parameters.....	76, 77	Configuring tool management Communicate with the PLC.....	178
NC Configurator.....	77	Control	
Compatibility.....	48	Demount.....	15
Compatibility mode.....	49	Control commissioning.....	141, 376
Complaints.....	6	Configure Sercos file.....	382
Conditional display of cell contents.....	272	Control-side machine parameters.....	376
Conditional editability of cell contents.....	276	Create Sercos files.....	384
Configuration of MTX		Log Sercos timing.....	388
Configuration and communication.....	142	Log transmitted Sercos parameters.....	387
Configuration of several different applica- tions.....	209	SCS handling.....	381
Configuration to use second database table.....	206	Sercos initialization.....	381
Configuration tools.....	179	Control, mounting.....	11
Menu items.....	182	Core dump.....	29, 42
Properties window.....	181	Core dump, SysError.....	29
Schema editor.....	179	CPL interfaces.....	291
Text editor.....	181	CPL parameter list.....	100
Tree view.....	181	Create contents of individual rows.....	187
ULC configurator.....	183	Comment.....	188
XML file editor.....	201	Copy cell definition.....	191
Configure data blocks.....	214	Edit text list.....	196
Configure database.....	213	Empty cell.....	188
Basic data type collection.....	224	Logic of cell contents.....	187
Configure data blocks.....	214	Merge cells.....	191
Default values for data blocks.....	238	Process variable.....	189
Define basic settings of tool manage- ment.....	224	Set general table properties.....	199
Define sector and place distribution of database table.....	213	Syntax of editing mask.....	198
Tool data type collection.....	220	Create DBT2 screen in IndraWorks Engi- neering.....	206
Tool system data structure.....	217	Create MTX control.....	147
Tool user data structure.....	219	Create number of lines and columns.....	185
Configure MTX.....	141	Create Sercos files.....	384
Configure Ethernet interface.....	146	Criticism.....	6
Configure function modules.....	144	Customer Feedback.....	6
Device data.....	146	Cycle calls, parameters settings.....	173
Configure Sercos file.....	382		
Configure several DBT visualizations.....	207	D	
Configure tool editors.....	282	D-corrections.....	175
Configuration screen.....	286	Data block.....	177
Configure M-keypads in tool editor.....	288	Data management.....	176
Configure tool editor controls.....	288	Database.....	177
New edit/Insert screen.....	285	DB configuration.....	204
Configure tool management.....	175	Configure several DBT visualizations.....	207
Basic data management.....	255	Number of DBT used.....	204
Configuration tools.....	179	DBLOAD.....	293
CPL interfaces.....	291	DBMOVE.....	293
Data management.....	176	DBSAVE.....	293
Database.....	213	DBSEA.....	292
Interfaces.....	291	DBSEAX.....	292
PLC interfaces.....	293	DBT1 screens.....	90
State upon delivery.....	294	DBTAB.....	292
Tool catalog.....	242	DBTABX.....	292
Tool corrections.....	175	DBTABXL.....	293
User interface.....	259	DCS.....	110, 119
Configuring the tool management		DCT.....	292

Default parameter, option-dependent with output condition (GNP).....	115
Default parameters.....	114
Default parameters, constant.....	114
Default parameters, option-dependent (GNP).....	115
Default values for data blocks.....	238
Define basic settings of tool management.....	224
Activate second database table.....	237
Activate value change in data block by PLC.....	231
Check input value via PLC.....	227
Define data-related user rights.....	227
Define input type.....	226
Define limit and enumeration values.....	227
Increase definition.....	226
Low-pass filter for change events.....	238
Message to PLC on value change in data block.....	230
Post-import function.....	236
Define basic tool management settings	
Set change events to be buffered in the control.....	238
Define sector and place distribution of database table.....	213
Demount.....	15
Demounting.....	9
Device data.....	55
Device version	49
Diagnostic application.....	153
Diagnostic messages on small operator panels.....	171
Dimensions.....	11
DIN parameter list.....	99
DIN words.....	100
Directory function	
Copy directory.....	352
Create new directory.....	351
Cut directory.....	352
Delete directory.....	351
Insert directory.....	352
Rename directory.....	351
Directory functions	
MTX Navigator.....	350
Display MZA messages.....	163
Distances.....	10
Distances for cooling.....	10
Documentation	
Change record.....	1
Drive commissioning.....	371
Drive-side parameterization.....	372
I/O wiring.....	371
E	
Edit.....	181
Edit tree view.....	181
Electric Installation.....	16

Electrical isolation.....	18
End clamp.....	9
Engineering Desktop.....	358
ENUM.....	112
Error case	
Core dump.....	29
Example: Pickup.....	321
Example: Turret.....	322
Examples for adjusting the physical process....	313
Export elements of project tree.....	148
Extension modules	
Mounting.....	14
External power supply unit.....	16

F

F panel Overview on functions.....	71
F-/M-/OP-panel function overview.....	71
F-keys for input assistance.....	123
F-panel.....	69
Feedback.....	6
Field bus.....	155
File function	
Copy file.....	353
Cut file.....	354
Delete file.....	353
Edit file.....	353
Insert file.....	354
Rename file.....	353
File functions	
Create new file.....	352
MTX Navigator.....	352
File manager.....	41
File structure.....	101
Bosch Rexroth, input masks.....	101
Graphics.....	102
Language support.....	102
Machine manufacturer, input masks.....	101
Mask definition files.....	101
Overlapping concept.....	101
Search strategy.....	101
User, input masks.....	101
Firmware download	
Sercos devices.....	368
Floating setup.....	19
Functional earth.....	19, 20
Functions	
MTX Navigator.....	345

G

G-code display.....	88
NC configurator.....	88
Properties.....	88
General axis commissioning.....	375
Global mask elements.....	108
Explanatory text.....	108
Graphics.....	108
Help.....	109

Index

GNP.....	115	I	
GNP variable address name.....	110	Import elements of project tree.....	148
Graphics.....	102	Import/export	
GritAttributes.....	199	Export MTX nodes.....	148
Grounding		Import MTX nodes.....	149
.....	19	IndraWorks - Engineering	
Grounding screw.....	20	Engineering.....	47
Group input masks.....	106	Inline CPLs.....	100
H		Input assistance display.....	87
Handling instruction		NC configurator.....	87
Activate E-stop function.....	372	Properties.....	87
Activate NC restart.....	389	Input dialogs.....	96
Activate safety engineering in drive.....	403	File structure.....	101
Adjust PLC program.....	400	Input mask.....	96
Backup drive parameters.....	61	Mask definition.....	104
Commission and program ProVi messages.....	171	Signature of calls.....	98
Commission safety engineering.....	400	Systematization of calls.....	99
Configure axis parameters.....	379	Input help display	
Configure list content.....	281	Properties.....	89
Configure list display.....	280	Input masks without signature.....	99
Configure SCS files.....	402	Inputs dialogs	
Configure tool editor controls.....	289	Input masks, overlapping.....	98
Create channel/axis.....	378	Inputs mask, definition.....	96
Create channel/axis/spindle interface.....	389	Installation.....	9
Create new list configuration.....	281	Installation and PLC interface.....	397
Create SCS files.....	388	Application structure.....	397
Data backup.....	53	PLC program part.....	398
Default values for data blocks.....	240	SCS configuration.....	399
Define data-relevant user privileges.....	291	Installation notes.....	9
Define new user text.....	197	Installation, electric.....	16
General drive commissioning.....	372	INT.....	111, 113
I/O wiring.....	371	Integrated safety engineering	394
Modify data block schema.....	239	Intended use	
Modify data element-related limit values.....	239	Areas of application.....	7
MZA configuration.....	170	Introduction.....	7
MZA interface.....	169	Use cases.....	7
Parameterize drives in IW drive.....	374	Interfaces.....	291
Restore data.....	54	CPL interfaces.....	291
Restore drive parameters.....	63	PLC interfaces.....	293
Sector and place distribution of a data-base table.....	214	State upon delivery.....	294
Set machine parameters.....	402	IW Operation desktop.....	45
Tool-specific limit value monitoring.....	240	IW-Drive - Sercos master.....	362
Traverse axis with control.....	375	Context menu.....	362
Hardware.....	45	Go online.....	363
Hardware data.....	36	Sercos device address assignment.....	364
Header.....	68	Using Sercos UC channel.....	366
Helpdesk.....	553	L	
HMI commissioning.....	67	Language-dependent text.....	103
HMI Setup		Languages - Support of multiple national and regional languages.....	102, 105
Commissioning M-keys.....	70	Layout template.....	286
Hotline.....	553	Licenses.....	31
Housing dimensions.....	11	List screens.....	94
		Log Sercos parameters.....	387
		Log Sercos timing.....	388

M

M-code display.....	88	Navigate in tree.....	344
NC configurator.....	88	Navigate list.....	344
Properties.....	88	Object types.....	342
M-keypad left and right.....	268	Operating possibilities.....	343
M-panel.....	70	Place file filter.....	345
M-panel Overview on functions.....	71	Program selection functions.....	350
Machine fault and status display.....	163	Properties dialog of file.....	347
Machine manufacturer.....	107	Properties dialog of the directory.....	346
Machine parameters - control-side.....	376	Set directory filter.....	346
MAND.....	117	Status bar.....	342
Mask definition.....	104	Toggling between tree and list.....	344
Structure.....	104	Tree + list.....	341
Text file, language-dependent.....	105	MTX project data.....	53
Title.....	105	MTX schema editor.....	180
Mask definition file.....	104	MTX system status.....	153
Mask definition files.....	101	Details.....	156
Mask definitions		Device information.....	156
Global mask elements.....	108	Functions.....	158
Output format.....	120	Info field with long status name.....	155, 156
Parameter types.....	109	IP address.....	154
Special parameter types.....	119	Name.....	154
Menu items.....	182	NC restart, button.....	155
Edit menu.....	182	Resources.....	157
File menu.....	182	Start/stop PLC, button.....	156
Help menu.....	183	Startup configuration.....	158
Schema menu.....	183	Startup mode.....	159
Search menu.....	182	Status field of NC.....	155
View menu.....	182	Status field of PLC.....	156
Window menu.....	183	Status info.....	155
Minimum distances.....	10	MTX version control.....	52
Mounting.....	9	MTX XM commissioning.....	23
Mounting control.....	11	Commissioning steps.....	23
Mounting extension modules.....	14	IndraWorks Engineering via the program-	
Mounting location.....	9	ming interface XF31.....	24
Mounting rail.....	9	Set the IP address for the XF5 port.....	23
Mounting S20 I/O modules.....	13	MTX XM42	
MT_DbData.....	293	Behavior in case of fatal exception (Sy-	
MT_DbLoad.....	294	sError).....	28
MT_DbRecList.....	293	Bootling.....	31
MT_DbSave.....	294	Control-internal data on "/OEM".....	28
MTX - Setting the time and date.....	160	Fallback and recovery.....	32
MTX diagnostics (MZA, ProVi, VCPs).....	163	IndraControl First Touch.....	36
MTX Emulation.....	151	IndraControl First Touch, core dump.....	42
MTX Navigator.....	341	IndraControl First Touch, file manager.....	41
Add directory filter.....	346	IndraControl First Touch, firmware man-	
Context-dependent functions.....	343	agement.....	37
Directory functions.....	350	IndraControl First Touch, general infor-	
Export dialog.....	349	mation.....	36
Export directories/files.....	349	IndraControl First Touch, hardware data.....	36
F-key bar.....	344	IndraControl First Touch, network settings....	37
File function.....	352	Initial firmware.....	29
Functions.....	345	Initial firmware, Install backup from exter-	
Import dialog.....	348	nal SD card.....	31
Import directories/files.....	348	Initial firmware, LED states.....	29
List.....	342	Initial firmware, starting and opening the	
Menu bar.....	344	initial firmware.....	30
		Runtime menu.....	35
		Select the start mode.....	32

Index

MTX-Navigator		
Context menu.....	343	
MZA.....	163	
MZA files.....	166	
MZA message system, commissioning.....	165	
N		
Navigate list.....	344	
NC block sequences.....	95	
NC call.....	319	
NC configuration.....	356	
NC configurator - machine parameters		
Select parameters.....	360	
NC Configurator - machine parameters.....	358	
Find parameters.....	361	
NC Editor - SCS files.....	362	
NC file system.....	341	
NC note.....	110	
NC notes.....	121	
NC ready bit.....	405	
NC simulation.....	137	
NC Simulation		
Basic license.....	137	
Simulation virtual machine.....	138	
Tool path.....	137	
NC subroutine TCH.....	310	
Calling TCH jobs.....	319	
Calling the tool change.....	310	
Examples for adjusting the physical process.....	313	
General information.....	310	
Physical process.....	312	
NC Subroutine TCH		
TCH at block pre-run.....	318	
New screen.....	285	
Notes to create a user-defined tool change.....	338	
Identifying the active tool.....	341	
Notes to create a user-defined tool replacement		
Relevant system data.....	338	
Number of the DBT used.....	204	
O		
Offline editor.....	284	
Offset display.....	82	
NC configurator.....	82	
Properties.....	83	
Offset table display.....	83	
NC configurator.....	83	
Properties.....	83	
Online correction display.....	89	
Online editor.....	284	
OP-panel (operating keys).....	69	
OP-panel Overview on functions.....	71	
Open or create configuration.....	183	
Create new configuration.....	183	
Open existing configuration.....	183	
Save changed configuration.....	185	
Open row editor.....	186	
Operating and display elements		
LEDs on the XD1 plug.....	24	
Operation mode switches SF1 and LEDs.....	25	
Operating screens.....	133	
Operation		
MTX Navigator.....	341, 343	
Output format.....	120	
Overlap input masks.....	98, 105	
Overlapping concept.....	99	
Overlapping concept in file structure.....	101	
P		
PAIR.....	117	
Parameter composition.....	99	
Parameter list.....	94	
Parameter pre-assignment.....	114	
Parameter type.....	109	
Binary type BOOL.....	112	
Character string STRING.....	113	
Parameter type - Integer.....	113	
Parameter types		
Default values and pre-assignment.....	114	
Enumeration type ENUM.....	112	
Graphics, parameter-related.....	116	
Integer INT.....	111	
Parameter address.....	110	
Real number REAL.....	111	
Syntax.....	109	
Title.....	110	
Validation information.....	117	
Value range of a parameter.....	111	
Parameterize MZA message system.....	165	
Parameters without address.....	113	
Parameters without addresses.....	100	
Placement display.....	83	
NC configurator.....	83	
Properties.....	83	
Placement table display.....	84	
NC configurator.....	84	
Properties.....	84	
PLC call.....	319	
PLC commissioning.....	389	
Activate axes/spindles in PLC user program.....	389	
PLC interface.....	167	
Configuration.....	168	
Programming.....	168	
PLC interfaces.....	293	
Function blocks.....	293	
Structures.....	294	
PLC list view.....	94	
Position display.....	77	
NC configurator.....	77	
Properties - Load: velocity / acceleration.....	79	
Properties - View.....	78	
Position parameter.....	111	

Position parameters.....	100
Potential equalization.....	20
Power connector XD1.....	17
Power supply unit, external.....	16
Precision correction display.....	88
NC configurator.....	88, 89
Process displays.....	76
NC configurator.....	76
Properties.....	76
Product phase.....	1
Program coordinate display.....	87
NC configurator.....	87
Properties.....	87
Program nesting display.....	82
NC configurator.....	82
Properties.....	82
Program section display.....	82
NC configurator.....	82
Properties.....	82
Program selection function	
MTX Navigator.....	350
Programming screens.....	95
Project data MTX.....	53
Project node IndraMotion MTX.....	141
Properties of input mask.....	106
Protective conductor.....	18
ProVi.....	171
ProVi messages.....	171

R

Ready contact.....	441
REAL.....	111
Reference conductor.....	18
Remaining tool time counting.....	336
Remaining tool time monitoring.....	336
Restore control data.....	58
User input.....	58
Restore extended data.....	60
User input (1).....	60
Runtime menu.....	35

S

S20 I/O modules, mounting.....	13
Safe motion.....	396
Safely-limited velocity.....	396
Safe standstill.....	395
Safe Stop 1.....	395
Safe Stop 1 (EMERGENCY STOP).....	396
Safe Stop 2.....	395
Safety.....	410
Safety engineering.....	394
Safety functions	
Safe motion.....	396
Safe standstill.....	395
Schema editor.....	179
SCP Analyzer.....	65
Display elements.....	65

SCP process tree.....	65
Value interpretation.....	66
screen application.....	67
Screen application.....	75
screen frames.....	67
Screen segments.....	286
ScreenApplication.....	67
ScreenFrame.....	67, 68
Screens MTX.....	134
Validate.....	135
SCS handling.....	381
SD setup variables.....	329
SD variables.....	174
Sercos fast startup.....	391
Sercos initialization.....	381
Sercos master.....	362
Service hotline.....	553
Set compressed display mode.....	271
Set process-dependent bitmap display.....	273
Set process-dependent text display.....	275
Set width and height of rows and columns.....	187
SET1 milling MANUAL with dialog.....	313
SET2 Milling AUTOMATIC without motions.....	314
SET3 Milling AUTOMATIC with motions.....	315
SET4 Turning AUTOMATIC with motions.....	316
Setting the date.....	160
Setting the maximum number of edges.....	322
Setting the time.....	160
Shielding.....	20
Signature.....	98, 104, 105
Signature for the correction.....	98
Single line call.....	121
Special parameter types.....	119
Description coordinate system "DCS".....	119
Parameters with variable binding.....	120
Startup mode.....	159
Startup of MTX.....	158
State upon delivery.....	294
Bitmap libraries.....	547
Database.....	294
Geometry list.....	298
Status bits.....	458
Status list.....	301
Tool editor.....	302
Tool life list.....	300
User interface.....	298
Wear list.....	299
STRING.....	113
Sublist.....	93, 179
Subroutines.....	174
Suggestions.....	6
Support.....	553
System message display.....	89
NC configurator.....	89
Properties.....	89
Systematization of calls.....	99
SystemData screens.....	92

Index

T

Target group.....	1
TCH.....	310
TCH at block pre-run.....	318
TCP/IP.....	155
TCV.....	292
Technology display.....	80
NC configurator.....	80
Properties - spindle power/torque.....	81
Properties - View.....	80
Text editor.....	181
Text file.....	103, 105
Title of an input mask.....	105
TL_SPS_Copy.....	265
TL_SPS_Delete.....	266
TL_SPS_Move.....	266
Token.....	103, 105
Tool bitmap control	
Define display of coordinate system.....	289
Tool catalog.....	242, 461
Basic data block.....	243
Master data block.....	244
Primary tool.....	243
Replacement or alternate tool.....	243
Tool class.....	243
Tool technology.....	243
Tool type.....	243
Tool types.....	244
Tool change.....	307
Notes about the creation of a tool change...	338
Tool correction display.....	86
NC configurator.....	86
Properties.....	86
Tool correction register display.....	86
NC configurator.....	86
Properties.....	86
Tool corrections.....	175
Tool data.....	175
Tool data type collection.....	220
Tool display.....	84
NC configurator.....	84
Properties.....	84
Tool length corrections.....	176
Tool list display	
Channel-dependent database query.....	262
Tool lists configuration file.....	270, 449
Tool lists configurator.....	269
Tool management screens.....	259
Tool management with standard tool change	
General information.....	303
Tool system data structure.....	217
Tool types.....	244
Tool user data structure.....	219
ToolCursor.....	267
ToolCursorEditor.....	267
Tree view.....	181

U

ULC.....	93, 179
ULC configurator.....	183
Create contents of individual cells.....	187
Create desired number of lines and col- umns.....	185
Open and create configuration.....	183
Open row editor.....	186
Set width and height of rows and col- umns.....	187
Unintended use.....	8
Consequences, non-liability.....	7
Upgrading the CMLx5 control to the XM42 control.....	429
Use existing projects.....	174
User.....	107
User interface.....	259
Configure tool editors.....	282
Configure tool lists.....	260
Configure user management.....	291
F-key panel definition.....	264
List call - list identification.....	261
List content - definition.....	261
List display - definition.....	269
M-key panel definition.....	268
User management.....	49
Access protection.....	49, 50

V

Validation information.....	117
Alternative parameter.....	118
Obligatory parameter.....	117
Parameter pair.....	117
Value range checks during the entry of val- ues - definition.....	278
VAM simulator.....	425
Configurator.....	426
Variable address name.....	119
Variable list.....	94
Voltage supply.....	17, 18
Voltage supply, dimensioning.....	19

X

XD1.....	17
XF5.....	23
XF31.....	24
XM42 commissioning.....	23
XML file editor.....	201
Copy nodes.....	203
Copy XPath to clipboard.....	203
Delete nodes.....	203
Edit value.....	203
Expand and collapse node levels.....	204
Move nodes down.....	203
Move nodes up.....	203
Paste known nodes.....	202
Paste nodes.....	203

Paste nodes with freely definable names....	203
Paste text node.....	203
Search for strings.....	203

Notes

Notes

Bosch Rexroth AG

P.O. Box 13 57

97803 Lohr a.Main, Germany

Bgm.-Dr.-Nebel-Str. 2

97816 Lohr a.Main, Germany

Phone +49 9352 18 0

Fax +49 9352 18 8400

www.boschrexroth.com/electrics



R911393281