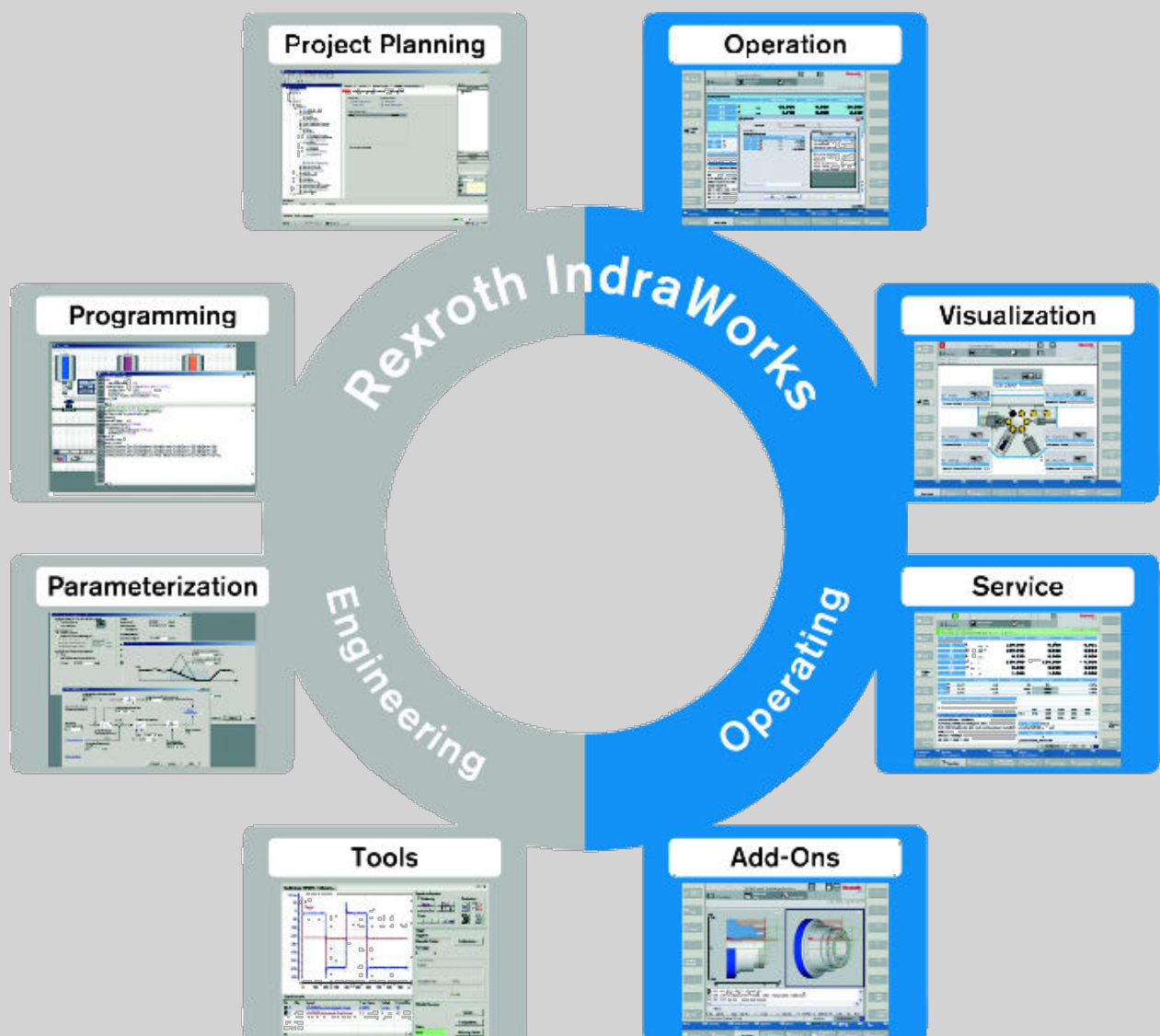


IndraWorks 14VRS

HMI

Application Description
R911343569

Edition 05



Title IndraWorks
14VRS
HMI

Type of Documentation Application Description

Document Typecode DOK-IWORKS-HMI*****V14-AP05-EN-P

Internal File Reference RS-2d199e89f4ec539d0a6846a500747a84-5-en-US-3

Change Record

Edition	Release Date	Note
Edition 01	2014-05	First edition
Edition 02	2014-10	Mobile panel support IndraControl VH2110.01 VEP15.6 Multi Touch: WinStudio Template
Edition 03	2015-05	Operating area-dependent OP-panel switching
Edition 04	2016-11	Adjusting the HMI IndraWorks Operation components to the new design and multi-touch operation External IndraWorks Operation language switching
Edition 05	2017-07	Adjusting the HMI editors to new touch devices in IndraWorks Engineering IndraControl PR/VR includes in the device library Extending HMI components to be visualized in the diff/merge framework Defdb00193127

Copyright © Bosch Rexroth AG 2017

This document, as well as the data, specifications and other information set forth in it, are the exclusive property of Bosch Rexroth AG. It may not be reproduced or given to third parties without its consent.

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 Visualization and Diagnostics, ThBe (SyKe/MePe)

Table of Contents

	Page
1 About this documentation.....	5
1.1 Validity of the documentation.....	5
1.2 Documentation structure.....	5
1.3 Required and supplementing documentation IndraWorks.....	6
1.3.1 Engineering.....	6
1.3.2 Engineering & visualization.....	7
1.3.3 Visualization.....	7
1.4 Information representation.....	7
1.4.1 Safety instructions.....	7
1.4.2 Names and abbreviations.....	8
1.5 Customer feedback.....	9
2 Important instructions on use.....	11
2.1 Intended use.....	11
2.1.1 Introduction.....	11
2.1.2 Areas of use and application.....	11
2.2 Unintended use.....	11
3 Overview.....	13
4 IndraWorks HMI user interface.....	15
4.1 Functional contents.....	15
4.2 Commissioning (only for VPx-/VSx devices).....	15
4.3 Starting IndraLogic V and the Operation Desktop on a visualization device.....	15
4.4 Screen layout.....	19
4.5 Header display.....	20
4.6 F-key operation.....	21
4.7 M-key operation.....	22
4.7.1 Visualization devices with foil keys	22
4.7.2 Visualization devices with touch screen	22
4.7.3 Error symbols on M-panel keys.....	23
4.8 User screens.....	23
4.9 Diagnostic display.....	24
4.9.1 Overview.....	24
4.9.2 Filter.....	26
4.9.3 Detailed view	26
4.9.4 HTML information.....	28
4.9.5 No communication.....	29
4.10 Logbook.....	29
4.10.1 Overview.....	29
4.10.2 Selecting Logbook.....	30
4.10.3 Deleting Logbook.....	31
4.10.4 Updating Display.....	31

Table of Contents

	Page
4.10.5 Export.....	31
4.10.6 Filter.....	32
4.11 Field bus diagnostics	33
4.12 Operating screens.....	36
4.13 Criteria analysis window.....	41
4.14 Machine overview.....	41
4.15 ACI screens.....	43
4.15.1 Configuration in IndraWorks Operation.....	43
4.15.2 Integration of User Controls.....	47
4.16 Version display.....	47
 5 Configuring the user interface.....	 49
5.1 Overview.....	49
5.1.1 Creating the visualization device.....	49
5.1.2 HMI wizard.....	51
5.1.3 Context menu of the visualization device.....	53
5.1.4 "Properties" dialog.....	54
5.1.5 Enabling the configuration data.....	60
5.1.6 Structure of the visualization device.....	65
5.2 Configuring screens.....	66
5.2.1 Overview.....	66
5.2.2 Screen editor	67
5.2.3 Configuring operating areas.....	71
5.2.4 Screen control by the PLC.....	74
5.2.5 External applications.....	75
5.2.6 Advanced copy function.....	76
5.3 Configuring F-keys.....	78
5.3.1 General information.....	78
5.3.2 Creating an F-panel.....	78
5.3.3 F-panel editor.....	79
5.3.4 Renaming an F-panel.....	82
5.3.5 Deleting an F-panel.....	82
5.3.6 Structuring F-panels in folders (from 13V06).....	82
5.4 Configuring OP-keys.....	83
5.4.1 General information.....	83
5.4.2 OP-panel editor	84
5.4.3 Additional OP-panels.....	86
5.5 Configuring M-keys.....	86
5.5.1 General information.....	86
5.5.2 Creating M-panels.....	87
5.5.3 M-panel editor	88
5.5.4 Renaming an M-panel.....	90
5.5.5 Deleting an M-panel	91
5.5.6 Structuring M-panels in folders (from 13V06).....	91
5.5.7 Programming M-keys in the PLC.....	91
5.5.8 Programming M-keys in the PLC for VEx devices.....	93

Table of Contents

	Page
5.6	Configuring default structures..... 95
5.7	OEM Data..... 108
5.8	HMI functions..... 111
5.9	Language-dependent user texts..... 112
5.10	Configuring logbooks..... 113
5.10.1	General information..... 113
5.10.2	Creating a new logbook..... 113
5.10.3	Logbook configuration..... 114
5.11	Configuring WinStudio screens..... 116
5.11.1	Overview..... 116
5.11.2	General WinStudio settings and functions..... 117
5.11.3	WinStudio screens..... 118
5.11.4	WinStudio variables..... 123
5.11.5	Linked symbols..... 124
5.11.6	WinStudio tasks..... 127
5.11.7	WinStudio tools..... 133
5.11.8	Database spy..... 134
5.11.9	LogWin..... 135
5.11.10	Cross reference search..... 136
5.11.11	Variable import wizard..... 136
5.11.12	Export/import..... 138
5.11.13	WinStudio licensing..... 138
5.11.14	Multilingualism 140
5.11.15	WinStudio applications..... 141
5.11.16	Further information..... 142
5.12	Configuring operating screens..... 142
5.13	Configuring ACI screens..... 147
5.13.1	General Information..... 147
5.13.2	Configuration in IndraWorks Engineering..... 148
5.13.3	Integrating User Controls into ACI Screens..... 165
5.13.4	F-Key Functions for ACI Screens and ACI Controls..... 175
5.14	User rights for screen and functions calls..... 178
5.15	Locking and hiding F- and M-keys by the user management..... 179
5.16	Converting configuration data of old IndraWorks versions..... 183
5.17	Configuring VCP devices..... 183
5.17.1	Overview..... 183
5.17.2	Inserting VCP devices..... 184
5.17.3	Calling the VI-Composer..... 185
5.17.4	Transferring project files..... 186
5.18	Configuring the Header Display..... 187
5.18.1	Overview..... 187
5.18.2	Configuring individual texts and symbols..... 189
5.18.3	Configuring for alternating display of messages..... 192
5.19	Export/import..... 193
5.20	Comparing visualization data 193
5.21	Validating visualization data..... 193

Table of Contents

	Page
5.22 Version control of the visualization devices.....	197
5.22.1 General information.....	197
5.22.2 Special features.....	198
6 IndraWorks process variables.....	199
6.1 Tool window for process variables.....	199
6.1.1 General information.....	199
6.1.2 Drag and drop.....	199
6.1.3 Filtering.....	199
6.1.4 Searching.....	199
6.1.5 Additional functions.....	199
6.1.6 Import mode.....	200
6.2 Process variables dialog.....	201
6.3 Browsing in WinStudio with OPC DA (OPC.IwSCP.1).....	201
6.3.1 Browsing with driver sheets.....	201
6.4 Browsing in WinStudio with OPC DA (IndraLogic OPC DA).....	201
6.4.1 Browsing with driver sheets.....	201
6.5 Browsing in WinStudio with OPC UA.....	201
6.5.1 Browsing in OPC UA driver sheets.....	201
6.6 Browsing in WinStudio with BR_WS driver.....	202
6.6.1 Browsing with driver sheets.....	202
7 Service and support.....	203
Index.....	205

1 About this documentation

1.1 Validity of the documentation

Overview on target groups and product phases

In the following illustration, the framed activities, product phases and target groups refer to the present documentation.

In the product phase "Engineering", the target group "programmer" can execute the activities "program" and "configure" using this documentation.

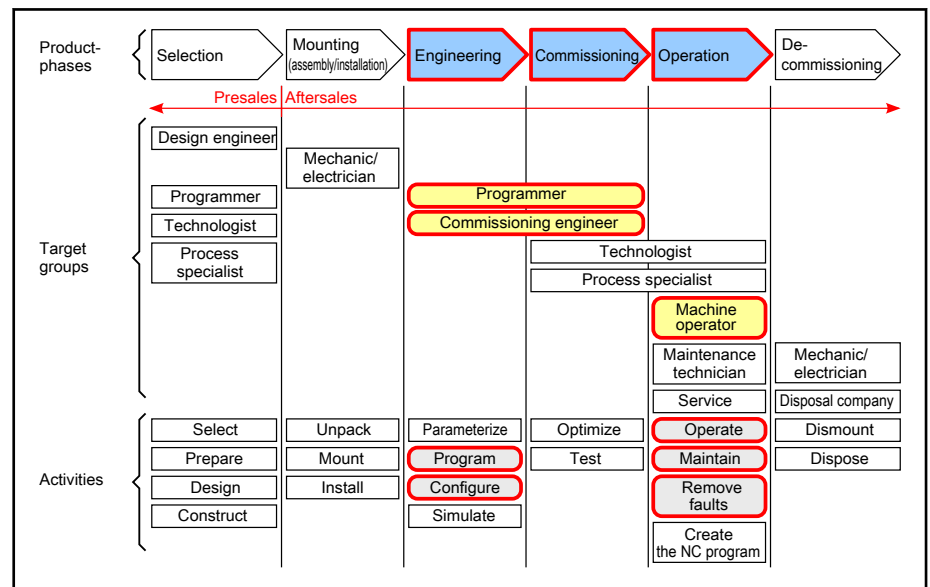


Fig. 1-1: Assigning this documentation to the target groups, product phases and target group activities

1.2 Documentation structure

The first part of the document contains important instructions on use ([chapter 2 "Important instructions on use" on page 11](#)).

[chapter 3 "Overview" on page 13](#) contains an overview of the visualization system.

[chapter 4 "IndraWorks HMI user interface" on page 15](#) describes the user interface.

[chapter 4 "IndraWorks HMI user interface" on page 15](#) describes the user interface used for process visualization and machine operation on VEx devices.

[chapter 5 "Configuring the user interface" on page 49](#) describes creating and configuring of visualization devices.

For information on the customer service helpdesk, refer to [chapter 7 "Service and support" on page 203](#).

About this documentation

1.3 Required and supplementing documentation IndraWorks

1.3.1 Engineering

Documentation titles with type codes and part numbers

IndraWorks 14VRS Software Installation DOK-IWORKS-SOFTINS*V14-CORS-EN-P, R911344286 This documentation describes the IndraWorks installation.
IndraWorks 14VRS Engineering DOK-IWORKS-ENGINEE*V14-APRS-EN-P, R911343566 This documentation describes the application 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 14VRS PLC Programming System IndraLogic 2G DOK-IWORKS-IL2GPRO*V14-APRS-EN-P, R911343571 This documentation describes the PLC programming tool IndraLogic 2G and its use. It includes the basic use, first steps, visualization, menu items and editors.
IndraWorks 14VRS Basic Libraries IndraLogic 2G DOK-IL*2G*-BASLIB**V14-LIRS-EN-P, R911343920 This documentation describes the system-comprehensive PLC libraries.
IndraWorks 14VRS WinStudio 7.4 DOK-IWORKS-WINSTUD*V14-APRS-EN-P, R911341585 This "User Manual and Technical Reference Book" facilitates working with the "Rexroth WinStudio"™ software for optimal results. This document provides technical information and step-by-step instructions to create web-enabled HMI/SCADA programs.
Rexroth IndraWorks 13VRS CamBuilder DOK-IWORKS-CAMBUIL*V13-APRS-EN-P, R911336291 This documentation describes the basic principles and operation of the CamBuilder, the cam editing tool.
IndraWorks 14VRS Field Buses DOK-IWORKS-FB*****V14-APRS-EN-P, R911341485 This documentation describes the field bus and local periphery connections supported by the IndraLogic XLC, IndraMotion MLC and IndraMotion MTX systems. The focus of this documentation is on the configuration, parameterization, commissioning and the diagnostics of the different periphery connections.
IndraWorks IndraMotion Service Tool DOK-IWORKS-IMST*****-APRS-EN-P, R911341383 This documentation describes the IndraMotion Service Tool (IMST). IMST is a web-based diagnostic tool used to access a control system via an Ethernet high-speed connection. The IMST allows OEMs, end users and service engineers to access and remotely diagnose a system. The PC has to use at least Internet Explorer 8, Firefox 3.5 or a higher version. The following control variants are supported: • IndraMotion MLC L25/L45/L65/L75/XM12/XM2/VPx • IndraLogic XLC L25/L45/L65/L75
Rexroth IndraDrive Service Tools IMST/IDST DOK-IM*MLD-IMSTIDSTV13-RERS-EN-P, R911342652 This documentation describes the IndraMotion Service Tools (IMST and IDST). It is a web-based diagnostic tool used to access a drive system via a high-speed Ethernet connection. Using these service tools OEMs, end users and service engineers can access and remotely diagnose a system from anywhere.

IndraWorks SafeLogic 14VRS First Steps DOK-IWORKS-SL*STEP*V14-CORS-EN-P, R911341520 This documentation provides information to facilitate an easy start in the safety control engineering SafeLogic. By using project examples, information about the installation, configuration, commissioning, troubleshooting and diagnostics is provided to the user.
Rexroth IndraWorks 12VRS FDT Container DOK-IWORKS-FDT*CON*V12-APRS-EN-P, R911334398 This documentation describes the IndraWorks FDT Container functionality. It includes the activation of the functionality in the project and working with DTMs.
IndraWorks 14VRS Field Buses Libraries DOK-IWORKS-FB*LIB**V14-LIRS-EN-P, R911343575 This documentation describes field bus libraries for the IndraLogic XLC, IndraMotion MLC and IndraMotion MTX systems.
Rexroth IndraWorks Remote Condition Monitoring RMB_TechRCM DOK-IWORKS-TEC*RCM*V14-LIRS-EN-E, R911343671 This documentation describes the individual functions of the "Remote Condition Monitoring" systems as well as the most important use cases for a quick status assessment of the connected machines/systems.
IndraWorks 14VRS SafeLogic - Project Configuration, Version 1.2.0.0. DOK-IWORKS-SL**PRJ*V14-APRS-EN-P, R911341694 This documentation describes the creation and programming of SafeLogic projects in IndraWorks Engineering.

Tab. 1-1: IndraWorks documentation overview - Engineering

1.3.2 Engineering & visualization

Documentation titles with type codes and part numbers

IndraWorks 14VRS Energy Efficiency Management DOK-IWORKS-4EE*****V14-APRS-EN-P, R911339229 This document describes the use of the Energy Efficiency Management function within the MLC, XLC and MTX systems. Using example applications, this documentation provides information about the project planning, configuration and visualization of the system.
--

Tab. 1-2: IndraWorks documentation overview - Engineering & visualization

1.3.3 Visualization

Documentation titles with type codes and part numbers

IndraWorks 14VRS HMI DOK-IWORKS-HMI*****V14-APRS-EN-P, R911343569 This documentation describes the HMI operating interface IndraWorks Operation, its operation and functions as well as the configuration in IndraWorks Engineering.

Tab. 1-3: IndraWorks documentation overview - Visualization

1.4 Information representation

1.4.1 Safety instructions

If there are safety instructions in the documentation, they contain certain signal words ("Danger", "Warning", "Caution", "Notice") and sometimes a safety alert symbol (according to ANSI Z535.6-2006).

The signal word draws attention to the safety instruction and indicates the risk potential.

About this documentation

The safety alert symbol (triangular safety reflector with exclamation marks), preceding the signal words "Danger", "Warning", "Caution" indicates hazards for persons.

The safety instructions are represented as follows in this documentation:

DANGER

In case of non-compliance with this safety instruction, death or serious injury **will** occur.

WARNING

In case of non-compliance with this safety instruction, death or serious injury **can** occur.

CAUTION

In case of non-compliance with this safety instruction, minor or moderate injury can occur.

NOTICE

In case of non-compliance with this safety instruction, material damage can occur.

1.4.2 Names and abbreviations

Term	Explanation
OEM	Original Equipment Manufacturer
Windows CE	Often shortened to WinCE, is a group of Microsoft operating systems for PDA and embedded systems
Windows XPe	Windows XP Embedded. It is a Microsoft operating system. Windows XP Embedded originates from Windows XP Professional, is based on the same source text but has to be differentiated from Windows CE/Pocket PC as it has a different device class. XP Embedded is only compatible with x86 hardware. Areas of use are devices based on an x86 processor
IndraWorks Engineering	Basic components of the Rexroth Automation platform in which the Engineering tools (IndraWorks HMI, IndraLogic etc.) are integrated in
IndraWorks Operation	Runtime environment (runtime) to operate and visualize PC-based Rexroth controls
Runtime environment	See: IndraWorks Operation
IndraWorks HMI	IndraWorks user interface existing of panels, header, system and user screens
IndraWorks project	Is created in IndraWorks Engineering and contains all configured machine or device data

About this documentation

Download	Configuration data transfer (runtime project) to the configuration directory of IndraWorks Operation
Panel	Key bars displayed on the screen (F-, M-, OP-keys) are labeled as panels
M-panel	Keypad with machine keys at the left and right screen margin
F-panel	Keypad with function keys above the OP panel to call a function
OP-panel	Keypad with operating keys at the lower screen margin to select an operating area
User screen	Screen created using Rexroth WinStudio
ACI	Application Container Interface
ProVi message	Diagnostic messages of the control (IndraLogic)
WinStudio	Configuration tool to generate user screens
IEC	International Electrotechnical Commission
Function block (FB)	Provides application-specific functionalities
BRWS	Bosch Rexroth WinStudio
OPC	OLE for Process Control, standardized software interfaces in the Automation engineering domain
HTML	The Hypertext Markup Language (HTML), also often called hypertext, is a text-based markup language to structure contents such as texts, images, hyperlinks in documents
XML	Extensible Markup Language (XML). It is a markup language to represent hierarchical structured data as text data
ASCII	American Standard Code for Information Interchange is a 7-bit encoding; it corresponds to the US-variant of ISO 646 and is used as basis for later encodings for character sets based on more bits

Tab. 1-4: Names and abbreviations used

1.5 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

Rexroth products stand for state-of-the-art development and manufacturing. The products are tested prior to delivery to ensure operating safety and reliability.

WARNING

Personal injury and material damages due to inappropriate use of the products!

The products have been designed for an industrial environment and may only be used according to their intended use. If the products are not used as intended, situations occur that result in damage to property or personal injury due to incorrect use of products.



As manufacturer, Rexroth does not assume any warranty, liability or compensation for damages which result from inappropriate use of the products. The risks of inappropriate use of the products are exclusively the risks of the user.

Before using Rexroth products the following conditions must be fulfilled in order to ensure an appropriate 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.
- The original condition of hardware products may not be altered; in other words, no structural modifications 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.

2.1.2 Areas of use and application

IndraWorks HMI is a Windows-based visualization and user interface for the IndraMotion XLC/MLC, IndraMotion MTX and IndraMotion MLD control systems and is used

- to commission tool and printing machines
- to operate and observe production processes
- diagnose machine states



The IndraWorks HMI package comprises several components. The versions of these components have to be compatible with each other.

2.2 Unintended use

Any use of IndraWorks HMI in applications other than those specified above or under operating conditions other than those described in the documentation and the technical specifications is considered to be "unintended".

The IndraWorks HMI may not be used if

Important instructions on use

- subjected to operating conditions that do not comply with the specified ambient conditions. For example, it must not be operated under water, under extreme temperature variations, or at extreme maximum temperatures.
- Furthermore, IndraWorks HMI must not be used in any applications which are not expressly released by Bosch Rexroth.

Please also note the information in the general safety instructions!

3 Overview

The IndraWorks automation platform provides a prepared visualization system which, with the help of integrated tools, can be extended to customize a user-specific user interface.

This documentation is intended for users wanting a quick overview of the IndraWorks HMI operating interface as well as for application engineers requiring a technical overview of the mode of operation and configuration of individual HMI components.

The user interface is based on the Microsoft Windows XP/Windows7/Windows10 operating system and the IndraWorks automation platform. Additionally, the Embedded terminals based on the Microsoft Windows CE operating systems are supported. The reduced functional scope of this operating system necessitates limited IndraWorks functionalities. Differences, if any, are appropriately indicated in the present documentation.

On delivery, the systems already provide all necessary software components ready for operation.

IndraWorks HMI is a Windows-based visualization and user interface for the IndraMotion MLC, IndraMotion MTX and IndraLogic XLC control systems.

4 IndraWorks HMI user interface

4.1 Functional contents

Diagnostics (ProVi)	Programming of ProVi messages in IndraWorks Engineering (or IndraLogic) and message display in the runtime environment.
Configuration	Configuration of F-panels, M-panels, OP-panels and screens with the IndraWorks Engineering.
User screens	User screens that can be programmed freely with WinStudio.
M-key operation	Operation by means of machine keys (eight keys left; eight keys right). PLC function: In IndraWorks Engineering, the "MKEYS" function block (FB) is provided in the form of a library. This function block has to be implemented in the PLC program. For each visualization device that should be triggered by the PLC, a function block instance has to be created.
Device types	The IndraWorks HMI user interface is available for PC-based devices (VPx/VSx devices), embedded terminals (VEH / VEP devices) as well as small control panels (VH21/VR21 devices). The functional scope for embedded terminals is limited as compared to that for PC-based devices. <i>The following functions are only provided for VPx devices and VSx devices:</i> • OP-keys
Operating screens	Configurable display of axis positions, variable states and machine operation statuses by means of machine keys.

4.2 Commissioning (only for VPx-/VSx devices)

Commissioning of the user interface of a VPx-/VSx device requires the following steps:

- Project creation in IndraWorks Engineering
- Transferring the visualization data
- IndraWorks Operation start

4.3 Starting IndraLogic V and the Operation Desktop on a visualization device

If the IndraLogic V and the Operation Desktop have to be operated simultaneously on one visualization device, the sequence of the applications must not be changed at startup. Before starting the Operation Desktop, booting of the IndraLogic V has to be completed. Only then, a reliable communication can be established to exchange visualization data between the IndraLogic V and the Operation Desktop.

To ensure the correct sequence at automatic startup, it is recommended to create a corresponding batch file (e.g. Boot_PLC_OPD.bat, refer to the following program listing) and to add it to the Windows AutoStart. Occurring errors are then diagnosed and output.

Batch file "Boot_PLC_OPD.bat"



The path specifications have to be adjusted to the current installation directory!

Program:

IndraWorks HMI user interface

```

@echo off
REM -----
REM IndraWorks Operation desktop is started after completed startup of the
REM IndraLogic V (soft control).
REM -----

REM -----
REM Check whether corrupt program exists.
REM NOTE: Adjust possible path to current installation directory.

cd "C:\Program Files\Rexroth\IndraWorks\SoftControl_Vxx3\Bin"
if EXIST "CORRUPT.PRG" goto :corrupt_exist
goto :start_plc_and_operation
REM -----

REM -----
REM Delete symbol file if program does not exist.
REM Delete file Corrupt.PRG if program exists.

:corrupt_exist
if NOT EXIST "DEFAULT.PRG" del /F "*.SDB"
del /F "CORRUPT.PRG"
goto :start_plc_only
REM -----

REM -----
REM File "ProjectLoaded.txt" is deleted is it exists and soft panel is started.
REM NOTE: Adjust possible path to current installation directory.

:start_plc_and_operation
IF EXIST "C:\Program Files\Rexroth\IndraWorks\SoftControl_Vxx3\Bin\ProjectLoaded.txt"
del "C:\Program Files\Rexroth\IndraWorks\SoftControl_Vxx3\Bin\ProjectLoaded.txt"
start "" "C:\Program Files\Rexroth\IndraWorks\SoftControl_Vxx3\Bin\SoftPanel.exe"
REM -----

REM -----
REM Wait till file "ProjectLoaded.txt" was written again,
REM maximum 40*5s = 200sc (ping runtime -n 6 ~ 5 seconds).
REM NOTE: Adjust possible path to current installation directory.

:try_again
IF EXIST "C:\Program Files\Rexroth\IndraWorks\SoftControl_Vxx3\Bin\ProjectLoaded.txt" GOTO valid
echo PLC program is loaded. Please wait.
ping 127.0.0.1 -n 6 >nul
set /a loop+=1
echo.|set /p =.
if %loop%==40 goto :FaultStartup
GOTO try_again
REM -----

REM -----
REM soft panel is started → Start Operation desktop.
REM NOTE: Adjust possible path to current installation directory.

:valid
start "" "C:\Program Files\Rexroth\IndraWorks\DDP.OperationDesktop.exe"
goto :end
REM -----

REM -----
REM Project data incorrect. IndraLogic V is started without Operation desktop
REM and an error message is output.
REM NOTE: Adjust possible path to current installation directory.

:start_plc_only
start "" "C:\Program Files\Rexroth\IndraWorks\SoftControl_Vxx3\Bin\SoftPanel.exe"
echo -----
echo Project data incorrect, IndraLogic V is started without Operation desktop!
echo Restart computer, Reload program is required!
echo If error message occurs again, check integrity of the
echo file directory (chkdsk)!
echo -----
echo.
pause
goto :end
REM -----

REM -----
REM Error when starting IndraLogic V. An error message is output.

:FaultStartup

```

IndraWorks HMI user interface

```
echo -----  
echo Error when starting IndraLogic V!  
echo Operation desktop is not started!  
echo Check IndraLogic V (soft panel)!  
echo -----  
echo.  
pause  
REM -----
```

```
:end
```

**Sequence diagram of the batch
file**

IndraWorks HMI user interface

The following figure shows the batch file sequence:

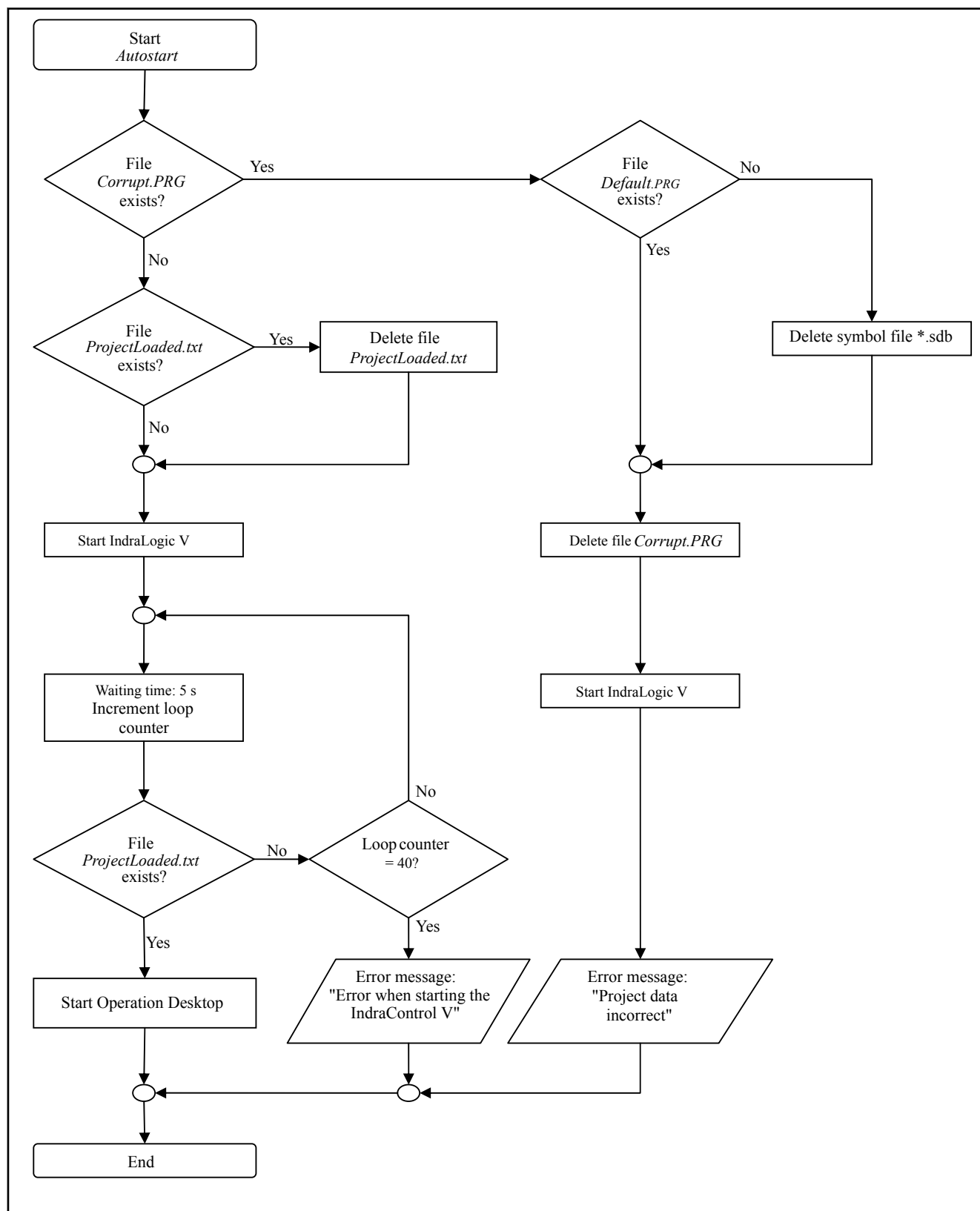


Fig. 4-1: Sequence of Boot_PLC_OPD.bat

4.4 Screen layout

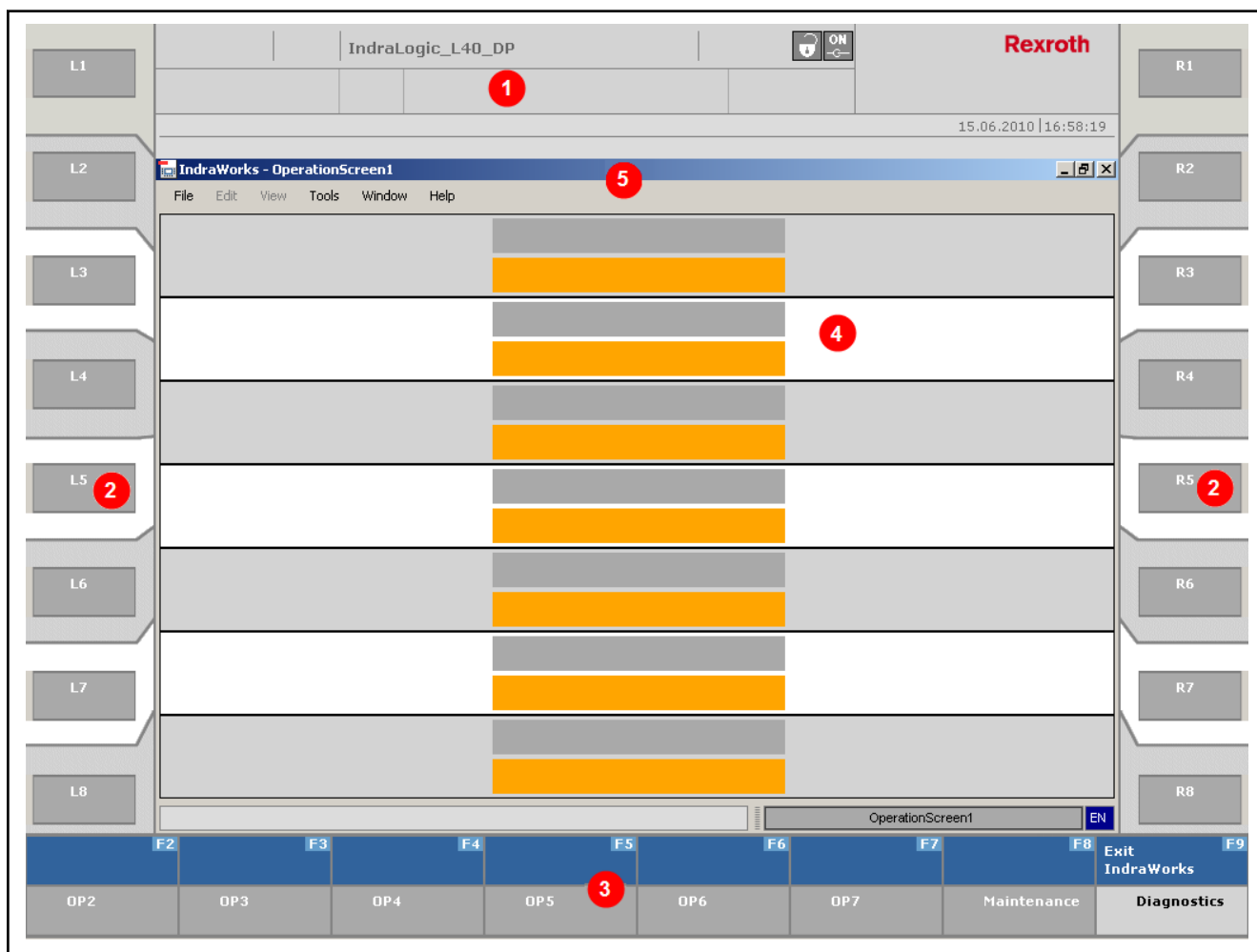


Fig. 4-2: IndraWorks HMI user interface

Header display ①

The header display is identical for all screens.

The following information is displayed:

- Control name or station name
- Diagnostic status
- ProVi error and ProVi warnings
- Date and time
- Name of active image

M-panel ②

M-panels are intended for machine operation or for execution of application-specific functions. The M-panels to the left and right of the screen show the assignment and the status of the M-keys.

OP- and F-panels ③

OP-panels are used to switch between the screens of different workspaces. Use the function keys to move back and forth between the screens, show an extended key assignment, or execute other application-specific functions. The currently active workspace and the screen selected are highlighted in color on the OP-panel and/or the F-panel.

Screen area ④

The display area shows the various applications.

IndraWorks HMI user interface

Menu bar ⑤ The menu bar provides various general functions and can be shown and hidden as necessary by pressing <ALT>+<M>.

The following functions can be called via the menu bar:

- Logging users on and off
- IndraWorks Operation exit
- Option dialog call
- Changing the language (using the "Options" dialog)
- Overview on the open application windows
- Version display call

4.5 Header display

The header display contains several elements which provide information on existing diagnostic messages, on the connection and on network statuses as well as on Winlock activation. Additionally, user-specific texts and icons can be configured that are displayed depending on the PLC variables.

The header provides the following information:

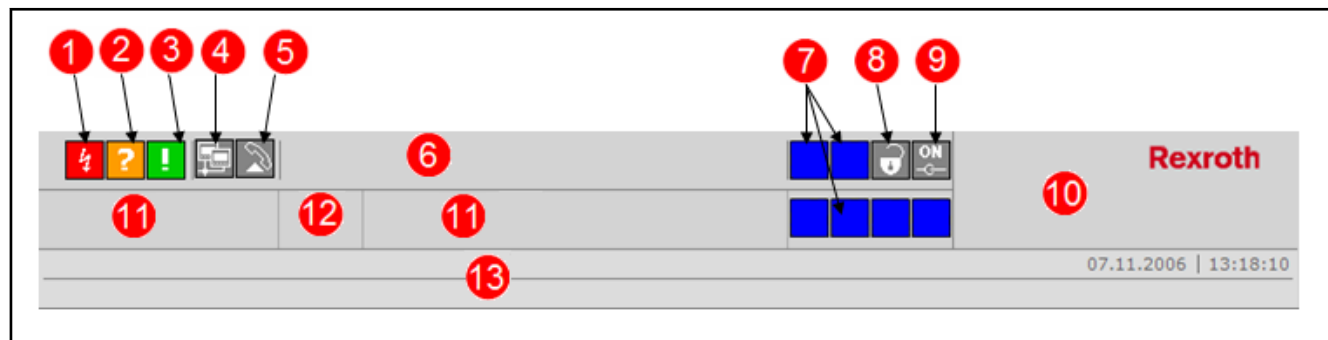


Fig. 4-3: Standard header

No.	Icon	Description
1		A ProVi error message is pending
2		A ProVi warning message is pending
3		A ProVi notice is pending
4		Link is active. There are no ProVi warnings or ProVi errors present at any network device
4		Link is active. A ProVi warning is present at at least one of the network devices or there is no connection
4		Link is active. A ProVi error is present at at least one of the network devices
5		Teleservice is switched on
5		Teleservice is switched off

IndraWorks HMI user interface

No.	Icon	Description
6	-	Name of the control connected to the HMI device. If the station name is available, it is displayed instead of the control name
7	-	Configurable application-specific symbols, which can be activated and deactivated via PLC variables
8		WinLock is enabled
8		WinLock is disabled
9		Connection status is online
9		Connection status is offline
10	-	Corporate logo.
11	-	Freely configured texts. Can be switched via the PLC variable. Configurable text, color and flashing status
12	-	Placeholder for primary and secondary operation modes
13	-	Message bars for ProVi errors, ProVi warnings, setup diagnostics, ProVi notices or start requirements (configurable)

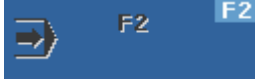
Tab. 4-1: Information displayed in the header

4.6 F-key operation

The devices are operated via the foil keys (F-panel).

The keys can have the following states:

- 

F-key "Normal"
- 

F-key "deactivated"
- 

F-key "disabled"
- 

F-key "hidden"

If the PLC variable is entered for "Write value", the value is set when clicking on this F-key.

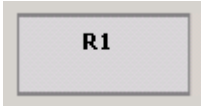
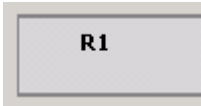
IndraWorks HMI user interface

4.7 M-key operation

4.7.1 Visualization devices with foil keys

The devices are operated via the foil keys (M-panels) at the sides of the screen.

The keys can have the following states:

- 
M-key "Normal"
- 
M-key "pressed"
- 
M-key "Active"
- 
M-key "pressed and active"
- 
M-key "deactivated"
- 
M-key "hidden"

IndraControl VPP 21 key layout

The IndraControl VPP 21 visualization device is provided with three foil keys to the left and three foil keys to the right of the screen. The upper and lower keys are navigation keys. The key in the middle is provided for selection and confirmation.

The keys are displayed as in the figures from "Normal" key... to "Deactivated" key.

Additionally, the selected key is highlighted using a red frame:

- 
Selected M-key

4.7.2 Visualization devices with touch screen

The user interface of a visualization device with a touch screen is operated by touching a key icon on the screen.

The M-keys are displayed as in the figures from "Normal" key... to "Deactivated" key.

⚠ WARNING

The touch screen key symbols can activate safety-critical machine functions.

For safety reasons, keys should be provided to lock the hardware for machine operation.

4.7.3 Error symbols on M-panel keys

PLC variable not found If a PLC variable which is not in the PLC program of the control or in the IndraLogic symbol file, (symbol configuration) has been configured for an M-key in the M-panel editor, a warning triangle is shown on the key:



Error display "PLC variable not found"

Key pressed If an M-key is pressed and held (e.g. for moving to a different screen) and a second M-key is pressed at the same time, the user is prompted to release the key. The new assignment is not active before this key is released:



Error symbol "M-key pressed"

4.8 User screens

IndraWorks HMI allows to integrate as many self-created user screens in the user interface as desired. The screens can be created with the WinStudio configuration tool.

User screens are called via F-keys, M-keys or OP-keys.

IndraWorks HMI user interface

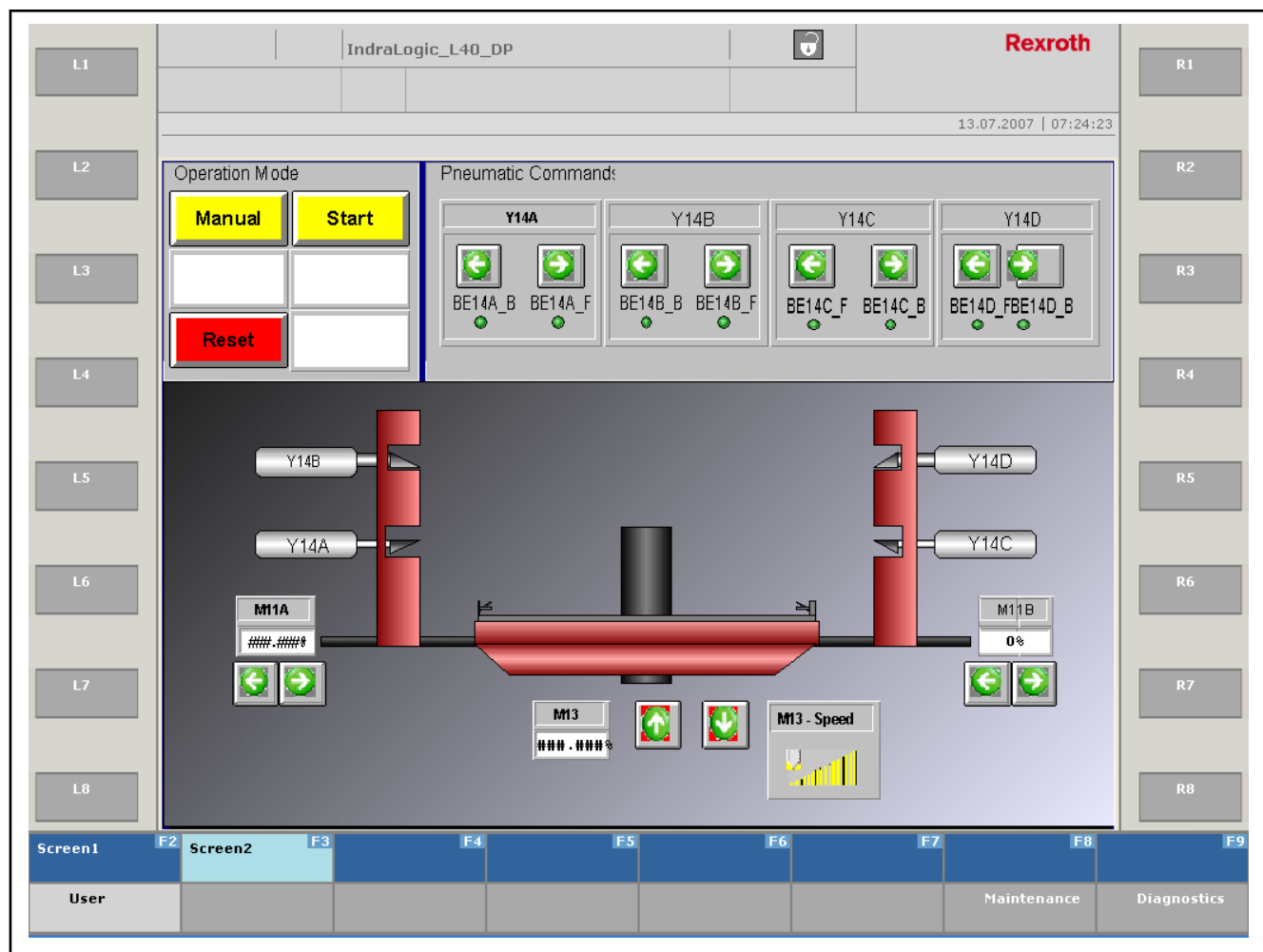


Fig. 4-4: Example of a user screen in the runtime environment

4.9 Diagnostic display

4.9.1 Overview

Pressing <OP9> activates the diagnostic screen.



The scope of delivery of IndraWorks HMI depends on the control system installed. While the IndraMotion MTX scope of delivery includes preconfigured screens for operating NC machines, the ndraMotion MLC/IndraLogic control only has the <OP9> key for calling the diagnostic screen.

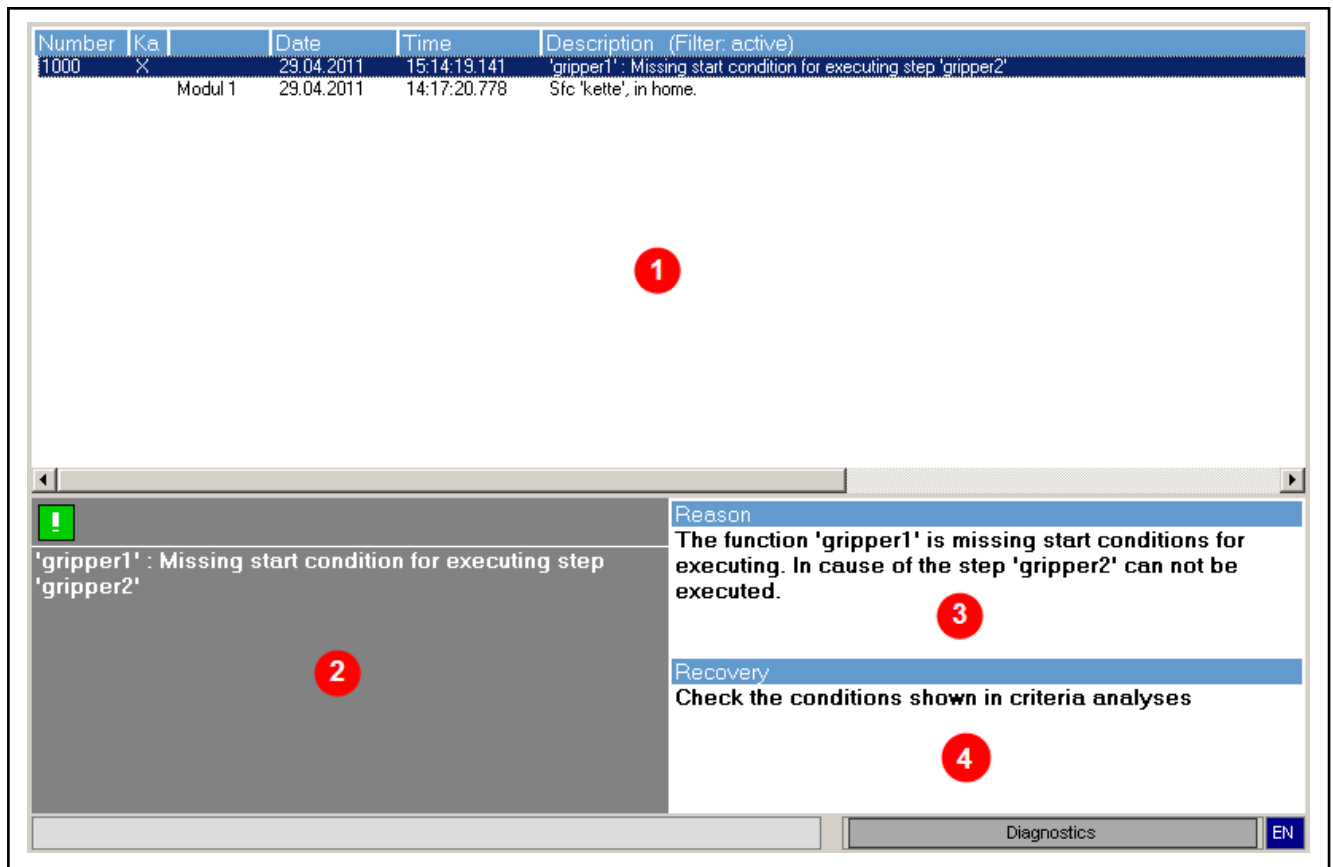


Fig. 4-5: Basic diagnostic screen

The basic diagnostic screen is divided in four areas:

- Message list ①** This area displays the diagnostic messages from the assigned control (see header). Listed are:
- the message number
 - the presence of the criteria analysis data
 - the module number of ProVi messages or the channel number of MTX messages
 - the time stamp indicating the time when the message occurred
 - and the brief description entered by the user
- Short info ②** This area shows the message selected as short text. A symbol represents the message types
- Error
 - Warning
 - Message
 - Setup diagnostics
 - Starting condition
- More information: Cause ③** This area shows more information. There, user texts illustrating the cause of the message in more detail, are filed.
- More information: Recovery ④** This area shows instructions for recovering the problem.

IndraWorks HMI user interface

4.9.2 Filter

<F1> on the diagnostic screen allows filtering by errors, warnings and messages. The labeling of the F-key indicates that the filter is active:

Press <F1> to switch through the various filter modes:

- **All <F1>**
No filtering. All diagnostic messages are displayed
- **Error <F1>**
Only messages of the "Error" type are displayed
- **Warning <F1>**
Only messages of the "Warning" type are displayed
- **Message <F1>**
Only messages of the "Message" type are displayed

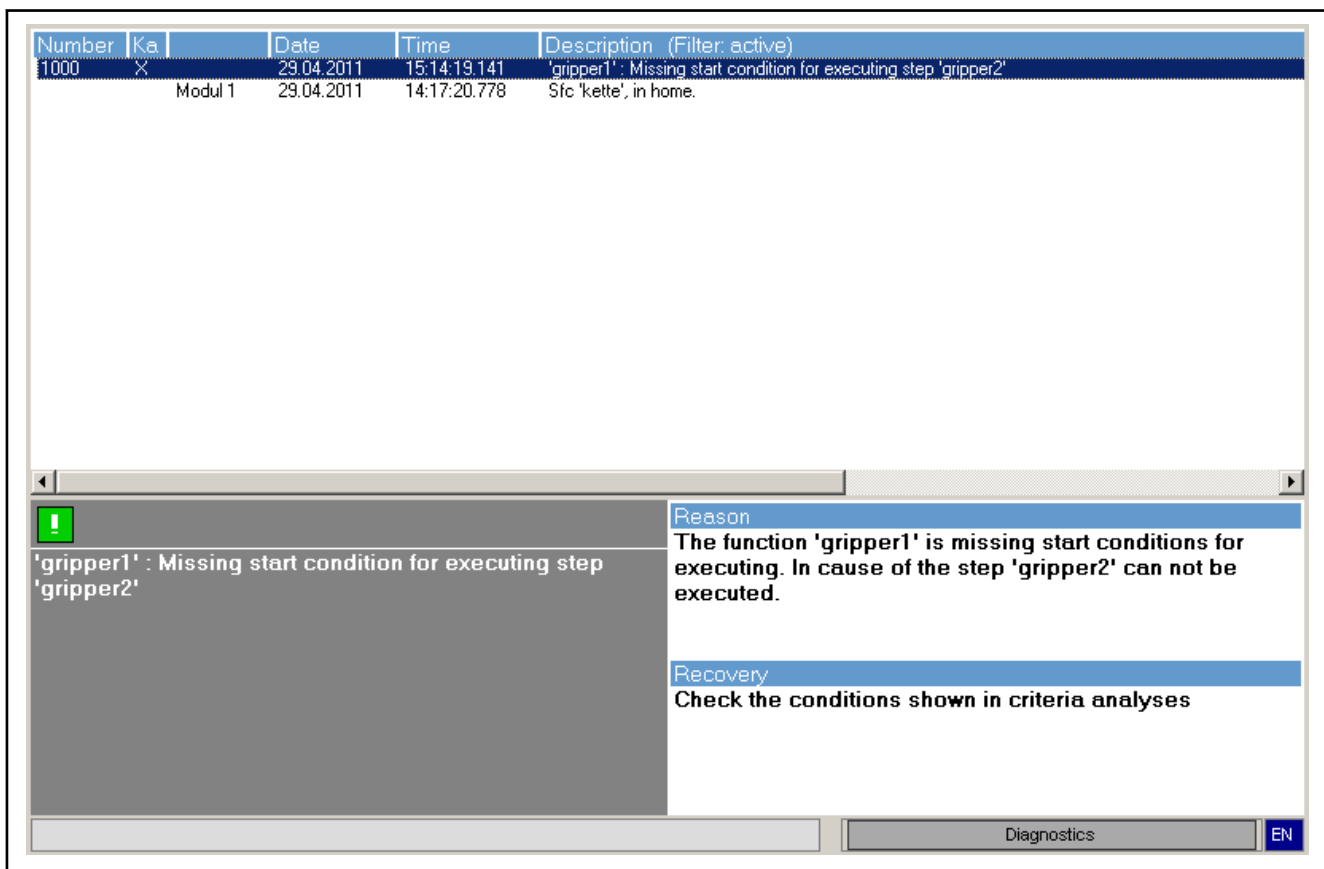


Fig. 4-6: Diagnostic screen filtered by errors

In the example above, the "error" filter is activated, so that only messages having the appropriate error characteristics are displayed.

4.9.3 Detailed view

Press <F2> to switch between the detailed view and the overview screen.

In the detailed view, the selected message is prepared again to contain more information.

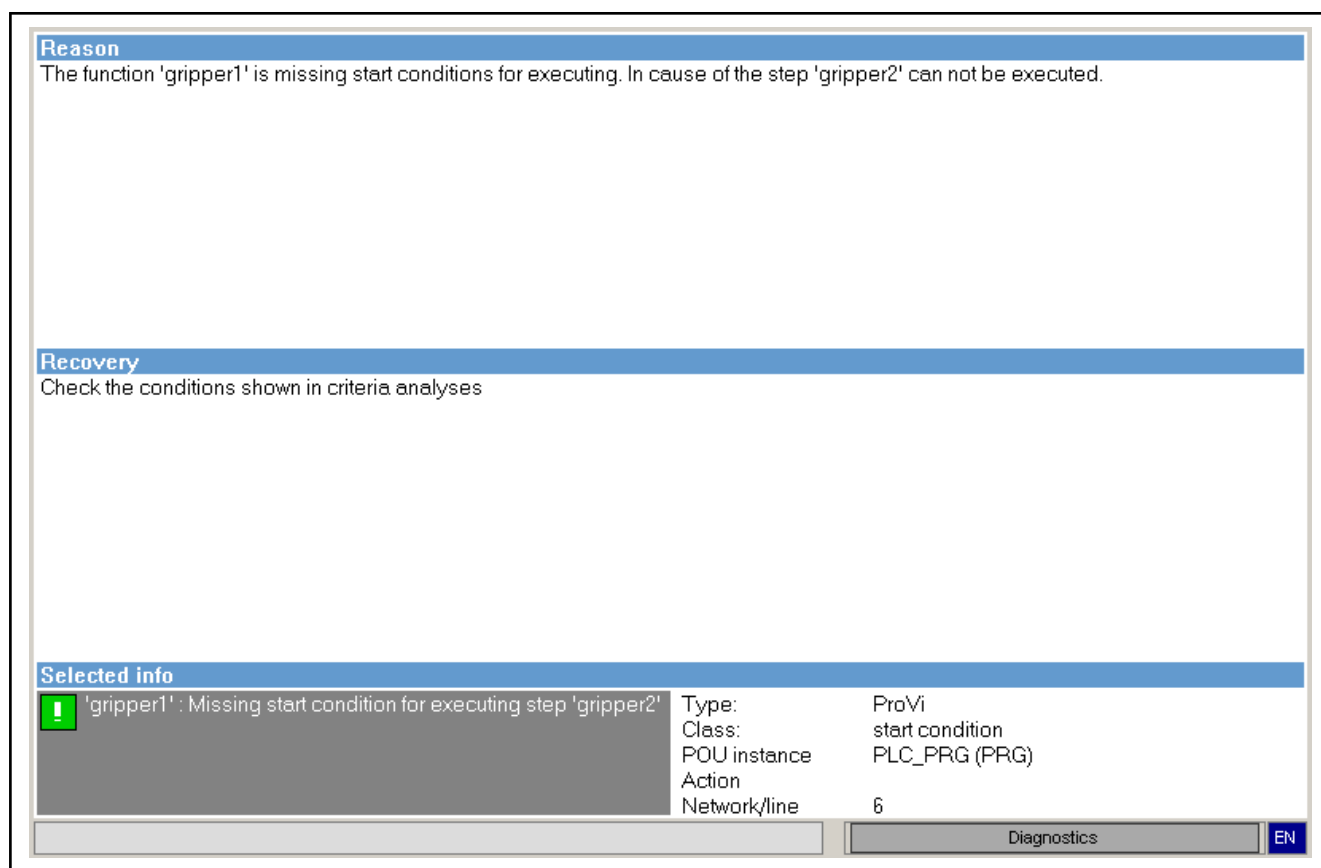


Fig. 4-7: Detailed view of the diagnostic screen

The "Cause" and "Recovery" windows are extended versions of the windows already shown in the overview view.

More information regarding the messages are displayed in the "Selected info". The additional information shown varies according to the type of message (e.g. ProVi).

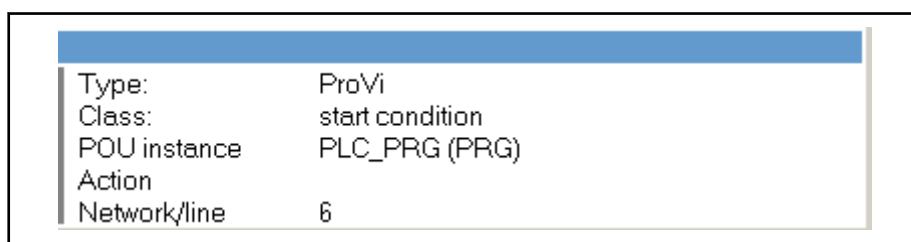


Fig. 4-8: Additional information about a ProVi message

The additional information describes the type of ProVi message and its origin in the program.

Use the cursor keys or the four navigation keys to scroll through the messages of the filtered overview list.

Press <F7> to move to "Overview".

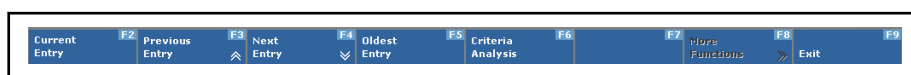


Fig. 4-9: Navigation keys for a detailed view of diagnostic messages

4.9.4 HTML information

The information on the cause and the recovery for a ProVi message cannot only be provided as text but also in HTML format. The HTML link is displayed in the recovery text of the overview. By selecting the ProVi message and pressing <F3> "Detail", the given file is displayed as part of the detailed view.

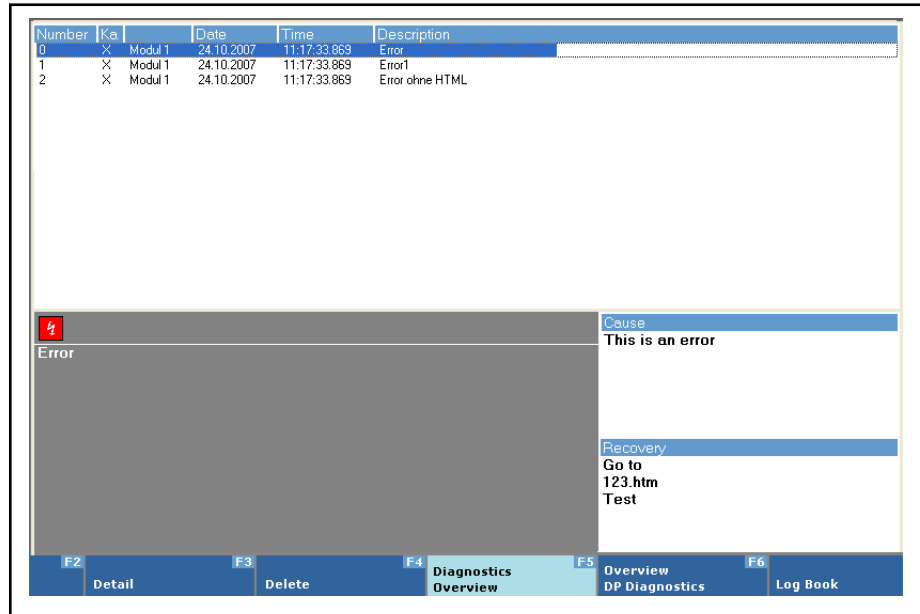


Fig. 4-10: Overview in the workspace Diagnostics with HTML information

The areas "Reason" and "Recovery" in the detailed view of the HTML information are substituted by an HTML page. An F-panel is shown for the detailed view with HTML information. The four F-keys in the F-panel are used for scrolling the HTML page. The F-panel for navigating between the individual diagnostics is available via a second F-panel level. If the detailed view including the HTML information is called from the diagnostic overview, the panel used for scrolling the HTML page is shown. The panel for navigating can be accessed via the level change. If the detailed view including the HTML information from the detailed view is called (e.g. by next entry, previous entry...), the navigation panel and the panel for scrolling can be accessed via a panel change.

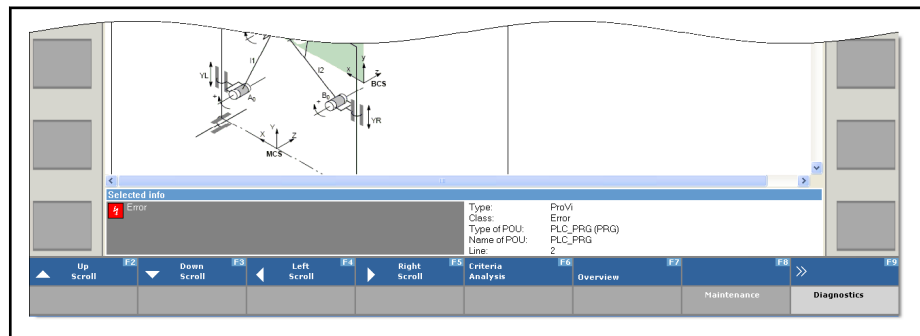


Fig. 4-11: Detailed view with HTML information (panel level: HTML pages)

IndraWorks HMI user interface

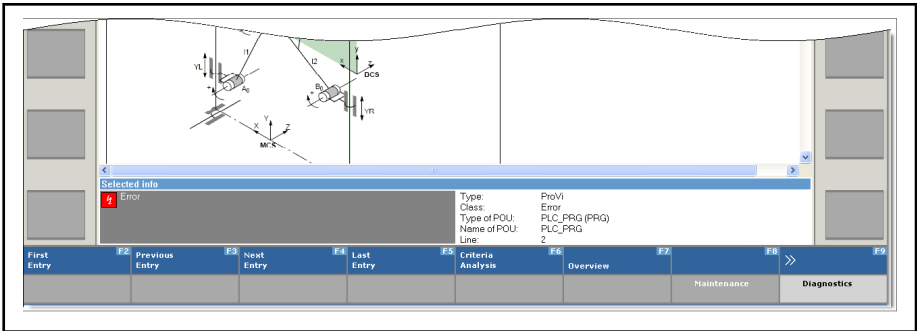


Fig. 4-12: Detailed view with HTML information (panel level: Diagnostic messages)



The level change  is only enabled if the ProVi message contains HTML information.

If the HTML file was not found, a default page appears containing the message "<Path>\<FileName> was not found".

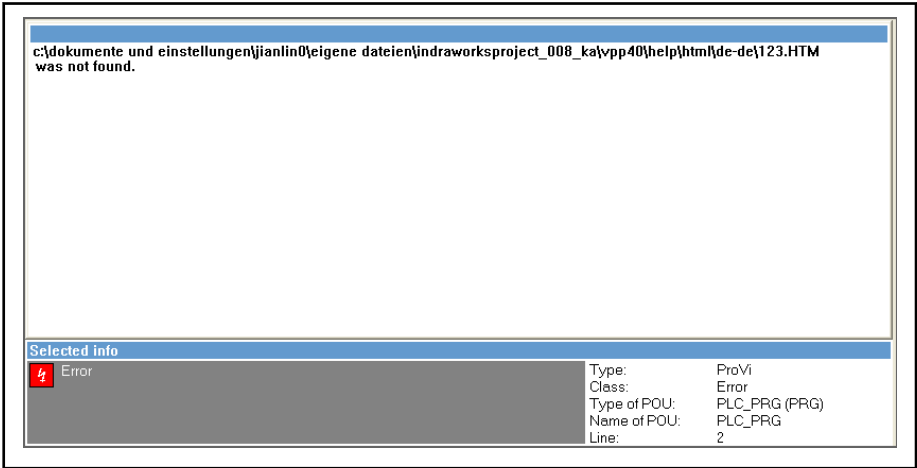


Fig. 4-13: Detailed view in the workspace Diagnostics - "File was not found"

4.9.5 No communication

If communication with the control is interrupted, diagnostic messages cannot be displayed. For example, this may happen if the connection cable has been removed. If communication is interrupted, the display of diagnostic messages is cleared. Once connection has been re-established, the display is automatically updated.

4.10 Logbook

4.10.1 Overview

The logbook is embedded in the diagnostic screen. After opening the diagnostic overview <F9>, press <F7> to change to logbook.

IndraWorks HMI user interface

Number	Ka		Date	Time	Description
1000	X		17.07.2007	11:48:01.962	'greifer1': Missing Start Condition to execute the step 'Greifer2_Zu'
1003	X	Modul 1	17.07.2007	11:48:01.961	'greifer1': the endpositions '%IX50.0; Greifer 1 offen' and '%IX50.1; Greifer 1 geschlossen' are not
	X	Modul 1	16.07.2007	18:16:40.569	Sfc 'Hub-Quer-Einheit erster Teile-Aufsetzer ', missing Startcondition
	X	Modul 1	16.07.2007	18:16:40.569	Sfc 'Hub-Quer-Einheit dritter Teile-Aufsetzer ', missing Startcondition
	X	Modul 2	16.07.2007	18:16:40.569	Sfc 'PR_HBQ_2', missing Startcondition

'greifer1': Missing Start Condition to execute the step 'Greifer2_Zu'

Cause

The working on release is missing to the function 'greifer1' for the execution, thus the step 'Greifer2_Zu' cannot be worked on

Recovery

Check the conditions, which are shown in the criteria analysis

Fig. 4-14: Basic diagnostic screen

4.10.2 Selecting Logbook

Opening logbook

Logbooks are configured in IndraWorks Engineering. As many logbooks as desired can be combined with any controls. If more than one logbook has been configured, a selection dialog opens when the logbook (<F7> key) is called.

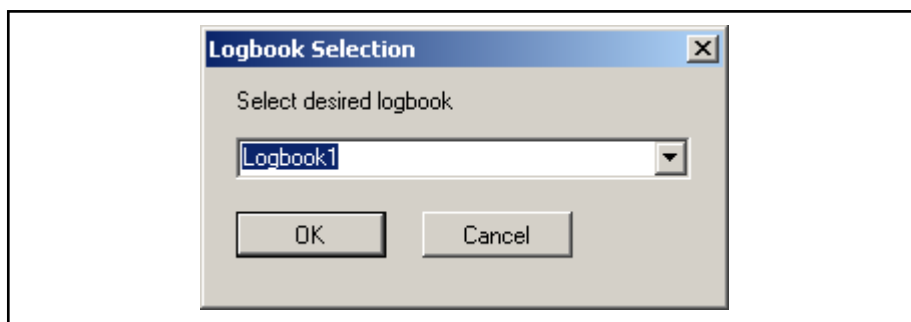


Fig. 4-15: Logbook selection dialog

If only one logbook is configured, the selection dialog is not displayed and the logbook is immediately activated.

If no logbook is configured at all, a warning is displayed.

Logbook functions

With <F5> "Selection", a logbook can be selected. With <F9>, the logbook display is closed and it is returned to the diagnostic overview.

Comes	Goes	Number	Source	Text
17.07.2007 07:37:11		1000	IndraLogic_L40...	'greifer1': Missing Start Condition to execute the s
17.07.2007 07:37:11		1003	Modul 1	no text available for message no 1003.
	17.07.2007 07:37:07	1000	IndraLogic_L40...	'greifer1': Missing Start Condition to execute the s
17.07.2007 07:37:07		1001	Modul 1	no text available for message no 1001.
	17.07.2007 07:36:36	1003	Modul 1	no text available for message no 1003.
16.07.2007 20:16:40			Modul 2	Sfc 'PR_HBQ_2', missing Startcondition
16.07.2007 20:16:40			Modul 1	Sfc 'Hub-Quer-Einheit dritter Teile-Aufsetzer ', mis
16.07.2007 20:16:40			Modul 1	Sfc 'Hub-Quer-Einheit erster Teile-Aufsetzer ', mis
16.07.2007 18:16:40			Modul 2	Sfc 'PR_HBQ_2', missing Startcondition
16.07.2007 18:16:40			Modul 1	Sfc 'Hub-Quer-Einheit dritter Teile-Aufsetzer ', mis
16.07.2007 18:16:40			Modul 1	Sfc 'Hub-Quer-Einheit erster Teile-Aufsetzer ', mis
16.07.2007 18:16:20		1000	Modul 1	ProVi Info zu Msg1000
16.07.2007 18:16:14	16.07.2007 18:16:20	1000	Modul 1	ProVi Info zu Msg1000
16.07.2007 18:16:07	16.07.2007 18:16:14	1000	Modul 1	ProVi Info zu Msg1000
16.07.2007 18:16:01	16.07.2007 18:16:07	1000	Modul 1	ProVi Info zu Msg1000
16.07.2007 18:15:55	16.07.2007 18:16:01	1001	Modul 1	no text available for message no 1001.
16.07.2007 18:15:49	16.07.2007 18:15:55	1001	Modul 1	no text available for message no 1001.
16.07.2007 18:15:43	16.07.2007 18:15:49	1001	Modul 1	no text available for message no 1001.
16.07.2007 18:15:37	16.07.2007 18:15:43	1001	Modul 1	no text available for message no 1001.
16.07.2007 18:15:30	16.07.2007 18:15:37	1000	Modul 1	ProVi Info zu Msg1000
16.07.2007 18:15:24	16.07.2007 18:15:30	1000	Modul 1	ProVi Info zu Msg1000
16.07.2007 18:15:18	16.07.2007 18:15:24	1000	Modul 1	ProVi Info zu Msg1000
16.07.2007 18:15:12	16.07.2007 18:15:18	1000	Modul 1	ProVi Info zu Msg1000
16.07.2007 18:15:06	16.07.2007 18:15:12	1001	Modul 1	no text available for message no 1001.

Text
'greifer1': Missing Start Condition to execute the step 'Greifer2_Zu'

Source
IndraLogic_L40_DP_1

Fig. 4-16: Selecting a logbook

4.10.3 Deleting Logbook

Press <F4> "Delete" to activate the delete function. This deletes all messages displayed with the currently set filters. The display is updated after completed deletion.

4.10.4 Updating Display

The logbook screen displays the messages statically. Press <F6> "Update" to update the screen.

4.10.5 Export

Press <F2> "Export" to show the "Export" dialog.

IndraWorks HMI user interface

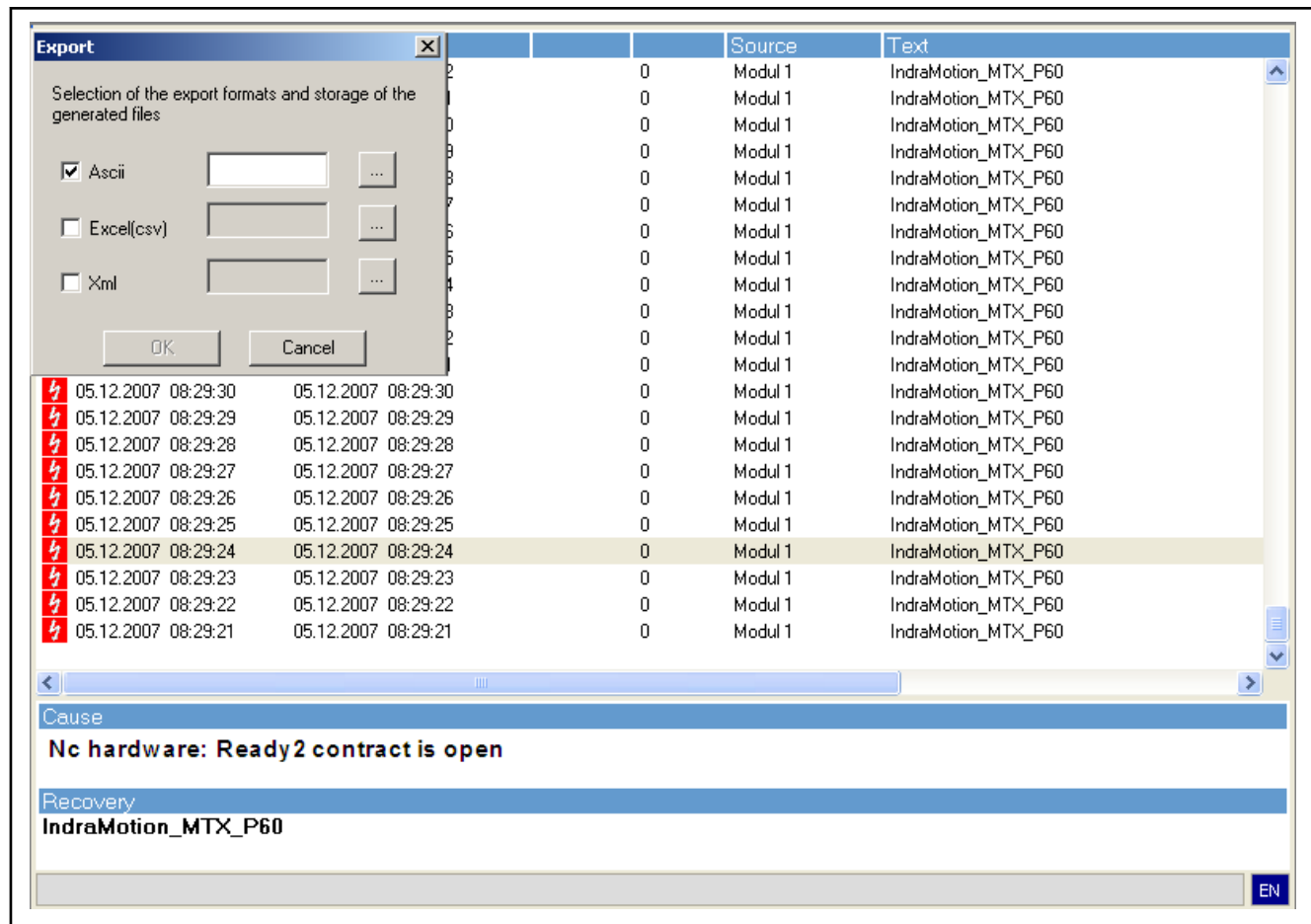


Fig. 4-17: Export dialog with a selection list of target files

In the export dialog, the following formats can be selected:

- ASCII
- CSV
- XML

Define path and name of the file to be exported in the input field. Press **OK** to export the file with the currently set filters. ASCII and XML files contain information on the filter settings.

4.10.6 Filter

Press <F3> "Filter" to display the Filter dialog.

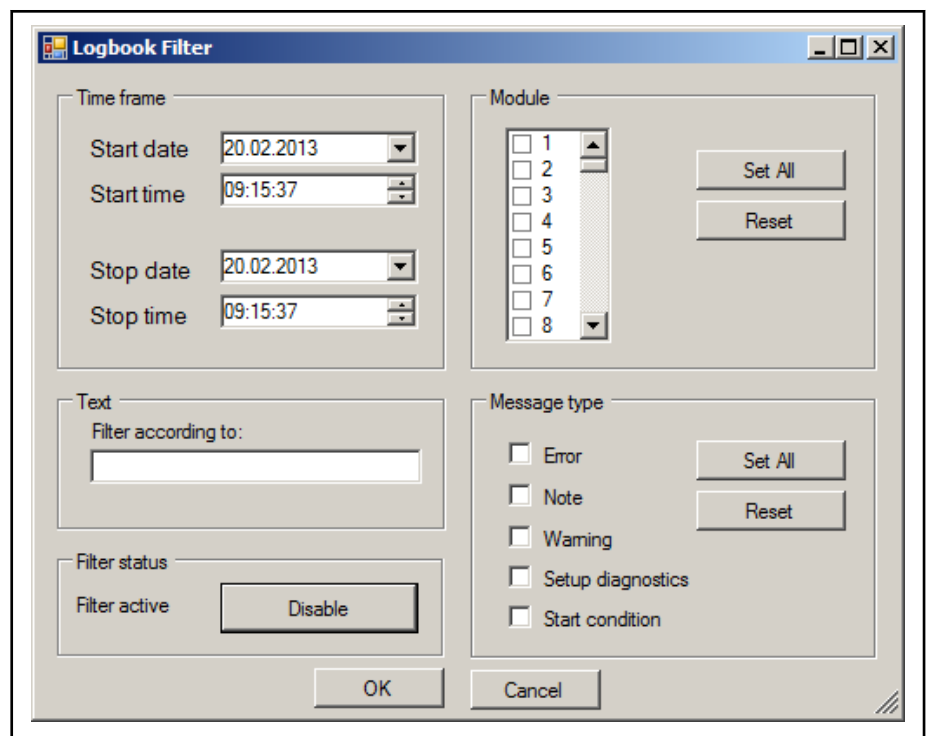


Fig. 4-18: Example: Filter dialog

The Filter dialog allows to limit the selection list of logbook messages.

Enter the start time and end time in the "Time frame" field. Only messages within this time period are displayed.

The "Source" field defines the control whose messages are considered.

Define the module numbers and the type of messages that are included in the log in "Module" and/or "Message type". The five message classes and the 99 modules may be combined in any way.

In the "Text" field, the messages available are searched for a defined text. Only messages within the text specified are displayed.

Enable or disable the filter with the **Enable** and/or **Disable** button.

Pressing **OK** enables the filters.

All settings are deleted when IndraWorks Operation is closed and the filter is disabled by default.

4.11 Field bus diagnostics

The field bus diagnostic display is embedded in the general representation of the diagnostics. The bus states of a Profibus DP connected to the control are displayed in the field bus diagnostics. Two views are available. The overview display provides an overview of the field buses configured in IndraWorks Engineering. The detailed view contains more detailed information about the individual bus devices selected in the overview.

The field bus diagnostics is called in two steps from the basic diagnostics view. In the first step, the key assignment of the function keys is switched using <F8>.

IndraWorks HMI user interface

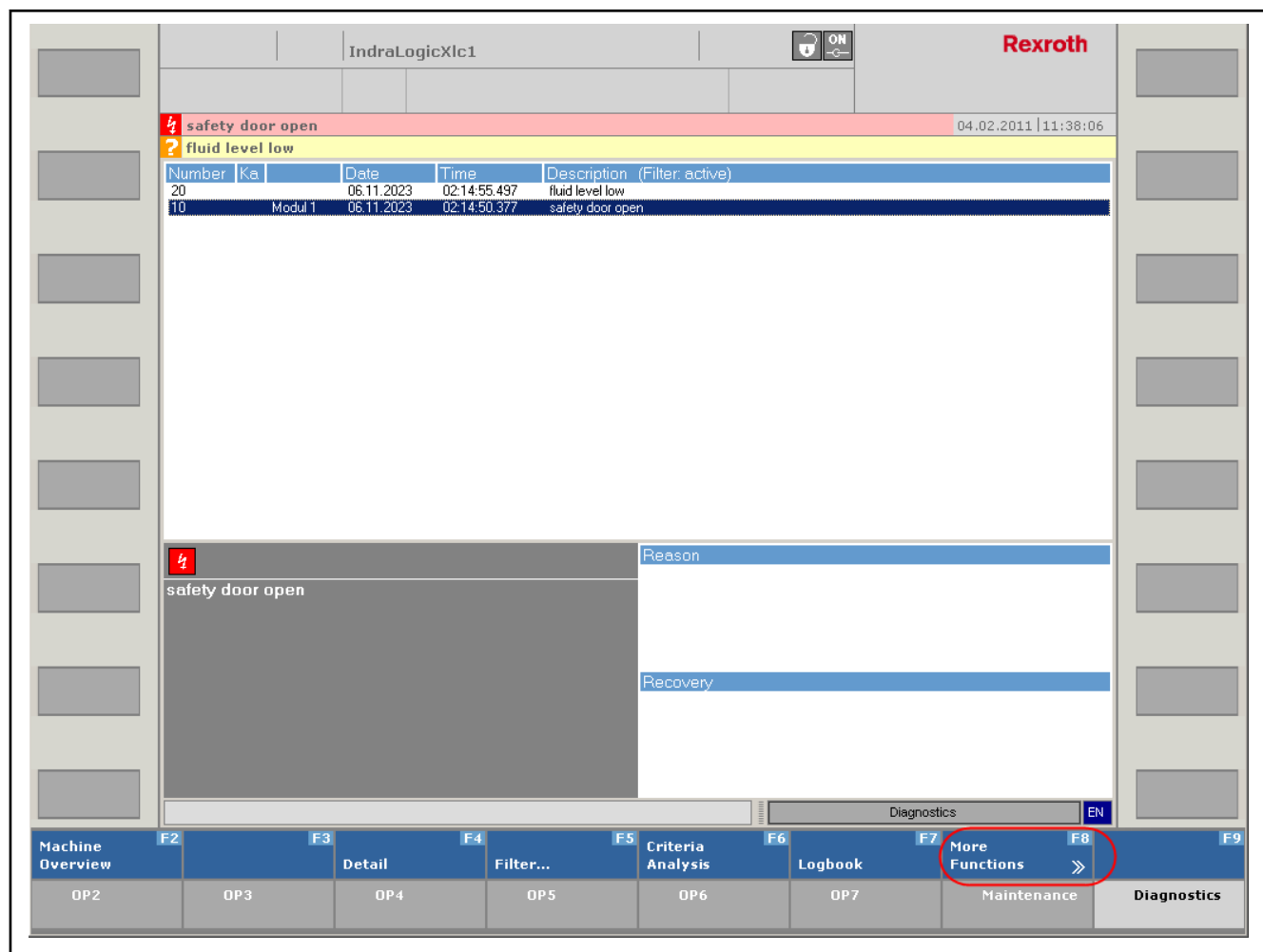


Fig. 4-19: Start screen of the diagnostics

Subsequently, the field bus diagnostics is called using <F2>.

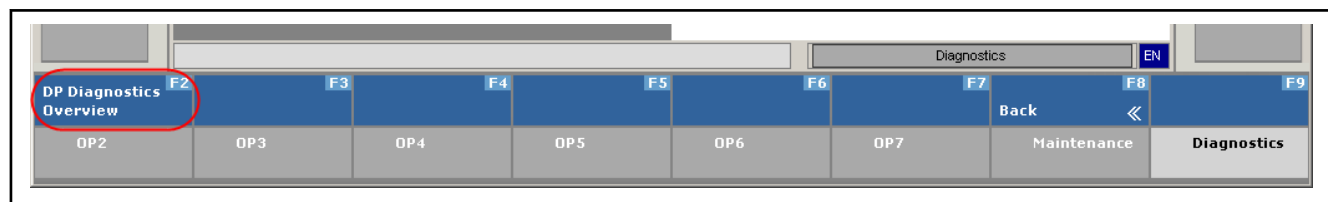


Fig. 4-20: Switched function key assignment in the diagnostic overview

After calling the field bus diagnostics, an overview of the currently configured Profibus DP is displayed.

The window is divided into three areas:

- In the upper area, the bus master state, the name of the control as well as the ident number are displayed
- In the center part of the screen contains the legend of the DP slave states as well as an overview of the number of the failed and the mal-functioning slaves
- In the lower part of the screen, the current bus configuration including all slaves is displayed. Each slave is displayed with the bus address and its current state

Number	Ka		Date	Time	Description
1000	X		17.07.2007	11:48:01.962	'greifer1': Missing Start Condition to execute the step 'Greifer2_Zu'
1003	X	Modul 1	17.07.2007	11:48:01.961	'greifer1': the endpositions '%IX50.0; Greifer 1 offen' and '%IX50.1; Greifer 1 geschlossen' are not
	X	Modul 1	16.07.2007	18:16:40.569	Sfc 'Hub-Quer-Einheit erster Teile-Aufsetzer ', missing Startcondition
	X	Modul 1	16.07.2007	18:16:40.569	Sfc 'Hub-Quer-Einheit dritter Teile-Aufsetzer ', missing Startcondition
	X	Modul 2	16.07.2007	18:16:40.569	Sfc 'PR_HBQ_2', missing Startcondition

 'greifer1': Missing Start Condition to execute the step 'Greifer2_Zu'	Cause The working on release is missing to the function 'greifer1' for the execution, thus the step 'Greifer2_Zu' cannot be worked on Recovery Check the conditions, which are shon in the criteria analysis
--	--

Fig. 4-21: Overview Profibus DP diagnostics

For more information about each bus device, refer to the detailed view. The detailed view is available for each slave and is displayed for the selected (black frame) bus device. Select the slave using the mouse or arrow keys. Open the detailed view using <F4>.

Once activated, the detailed view is displayed on the center and lower area of the overview screen. The representation provides information about the current bus device state, the slave address, the ident number and the status of the station. An error number transmitted by the control as well as the hexadecimal coded channel error text is displayed depending on the bus device.

IndraWorks HMI user interface

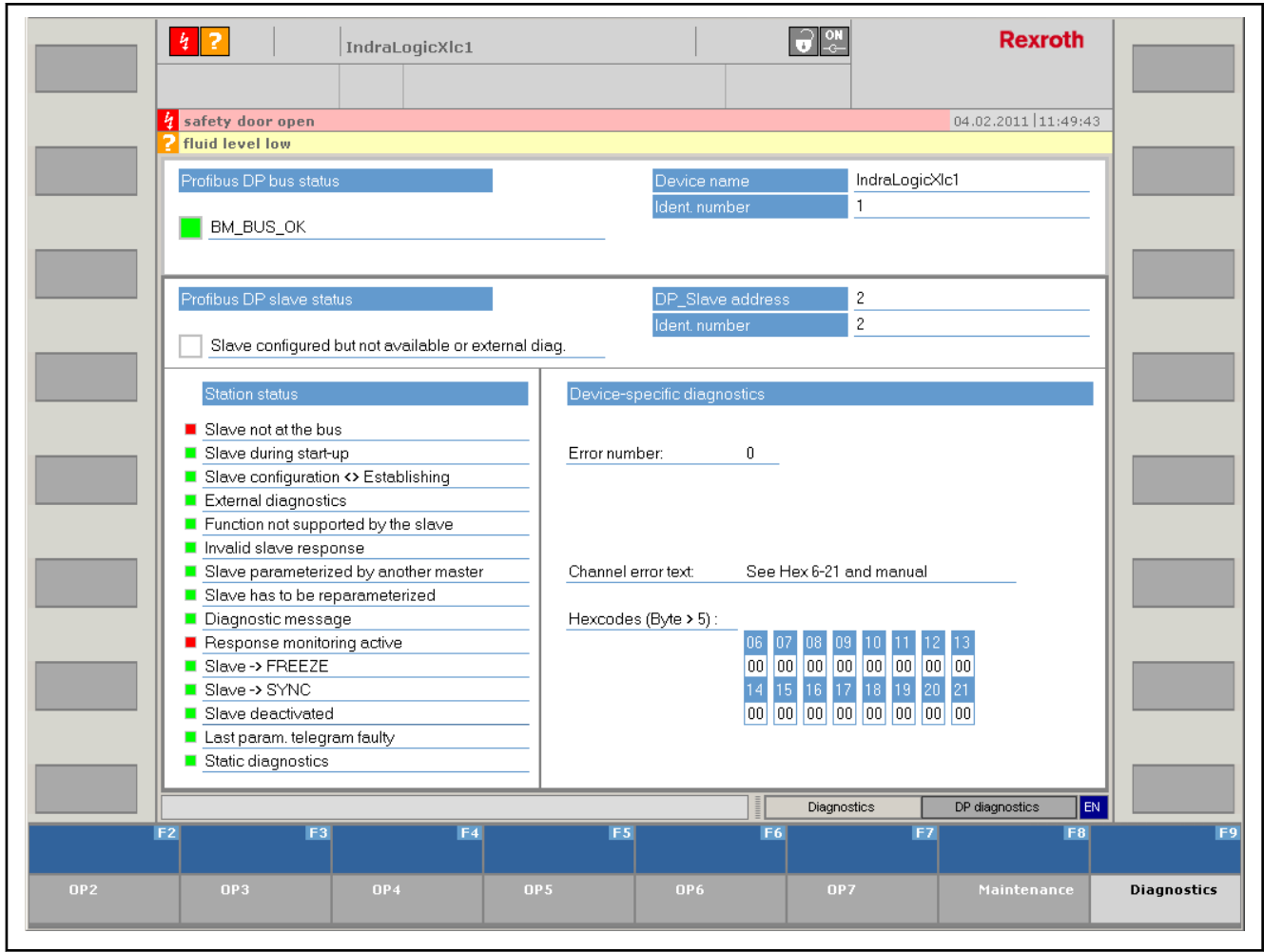


Fig. 4-22: Detailed view of a bus device

To access the detailed view of another slave, the user does not have to go back to the overview but can toggle between the detailed views of the bus devices using <F2> to <F5>.

Go back to the overview with <F9>. This key closes the detailed diagnostic view.

4.12 Operating screens

Operating screens manually operate the machine with the help of M-keys. The current position of the moving axis as well as the states of freely configurable PLC variables can be displayed. The meaning of individual states is determined while configuring the operating screen. They also show whether a specific action can be executed.

F-keys switch between operating screens.

IndraWorks HMI user interface



Fig. 4-23: Operating screen
Depending on its configuration, an operating screen may contain 5 or 7 operating lines.

IndraWorks HMI user interface

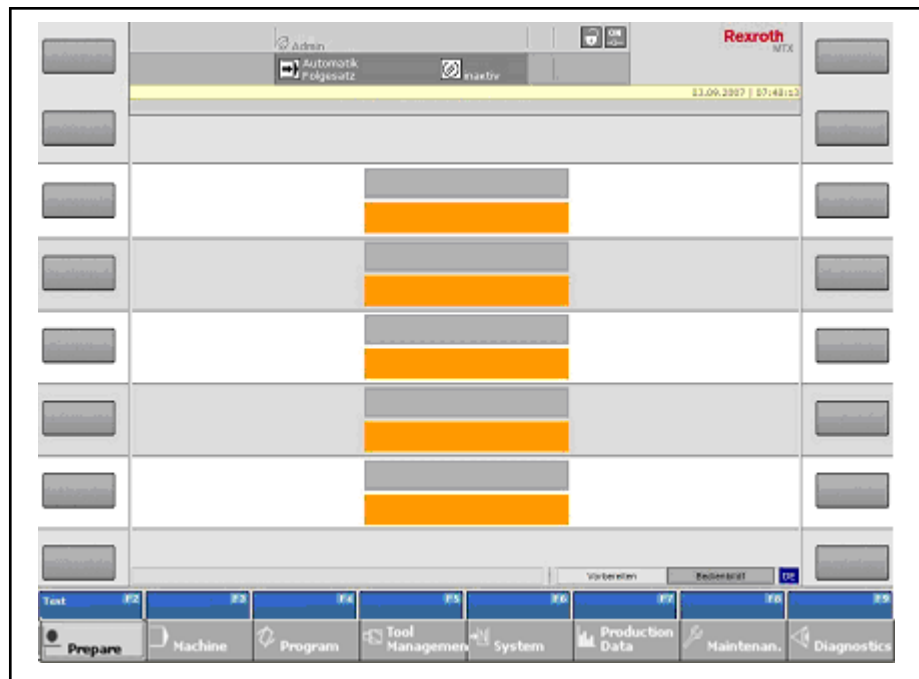


Fig. 4-24: 5-line user screen

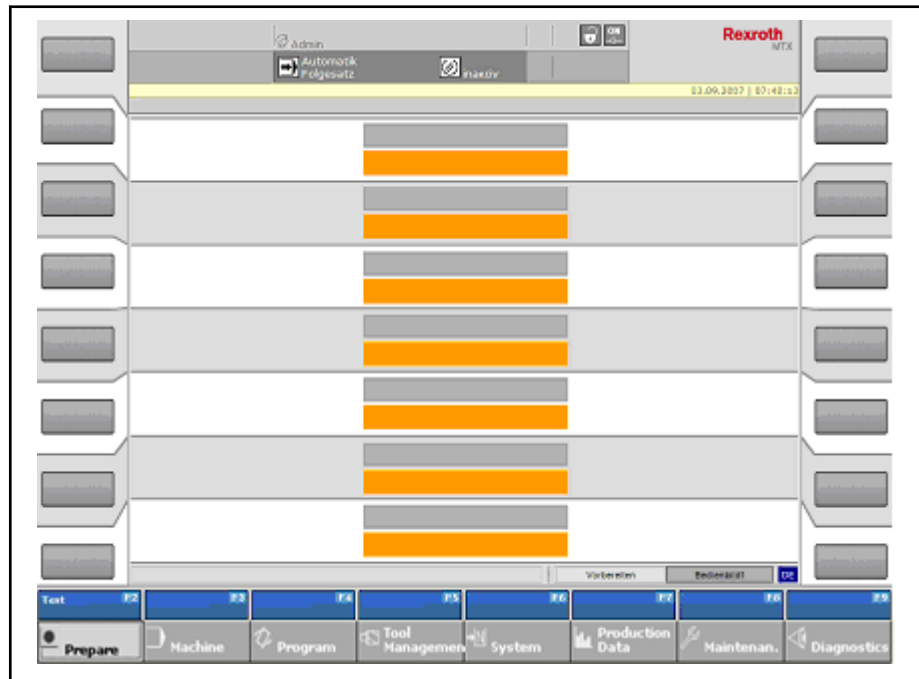


Fig. 4-25: 7-line user screen

Configure the following sections for each operating line with the operating screen editor:

- title
- a position display
- four status displays
- Two executability displays

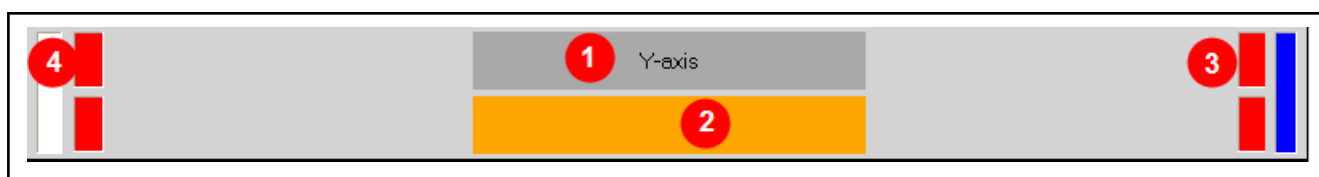


Fig. 4-26: Operating line

- Title ①** Description of the operating line.
- Position display ②** This display shows the current position of moving axes. The display value can be configured either via a PLC variable or by entering the axis number and the channel (for NC machines).
- Status display ③** This display shows the states of freely configurable PLC variables. This field is available only if a PLC variable is assigned to it.
- Feasibility display ④** To activate the feasibility display, either a freely configurable PLC variable can be used or an SFC flag can be configured. If an SFC flag is used, the POU and the step or action have to be defined first by the operating screen editor. The SFC flag specified in this manner is subject to the criteria analysis at runtime. The result of the criteria analysis (feasible or not feasible) is then visualized via the feasibility display.
- If the motion cannot be carried out, press the M-key assigned to the operating line (<L1>-<L8> or <R1>-<R8>), in order to display the criteria analysis window.
- This field is available only if a PLC variable is assigned to it.

Operating screen information

The operating screen displays the variables used for the status display. The variable display can be found in the "Presettings" using the <INFO> button at the operating panel.

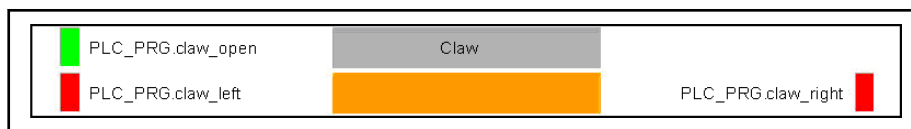


Fig. 4-27: Status line of an operating screen after switching to the variable display

When pressing <INFO> again, the I/O addresses assigned to the PLC variables are displayed. The PLC variables not linked with an I/O address, remain in the variable representation. If the operating screen does not use any I/O address, the representation mode only switches between text and PLC variable display.

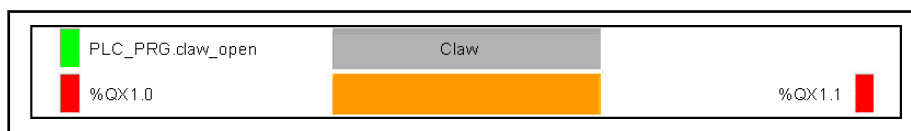


Fig. 4-28: Status line of an operating screen after switching to the I/O address display

Pressing <INFO> again, it is returned to the display of the status texts.

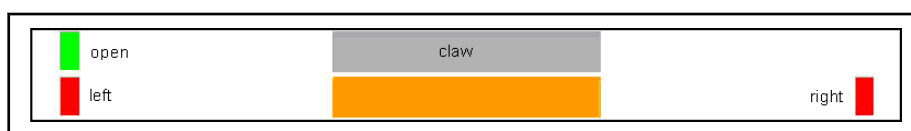


Fig. 4-29: Status line of an operating screen after switching to the text display

If there is no <INFO> button at the operating panel or if the user screen information should be reached via another button, it can be specified in the settings of IndraWorks Operation. The "Options" dialog to change different set-

IndraWorks HMI user interface

tings can be opened via the menu **Tools ► Options**. If the menu is not visible, it can be shown with <ALT> + <M>. In the "Options" dialog, select the folder "Shortcuts" and then in the list "Categories" the "HMI functions" entry. Subsequently, select the "Control panel information" in the "Commands" list and specify another shortcut.

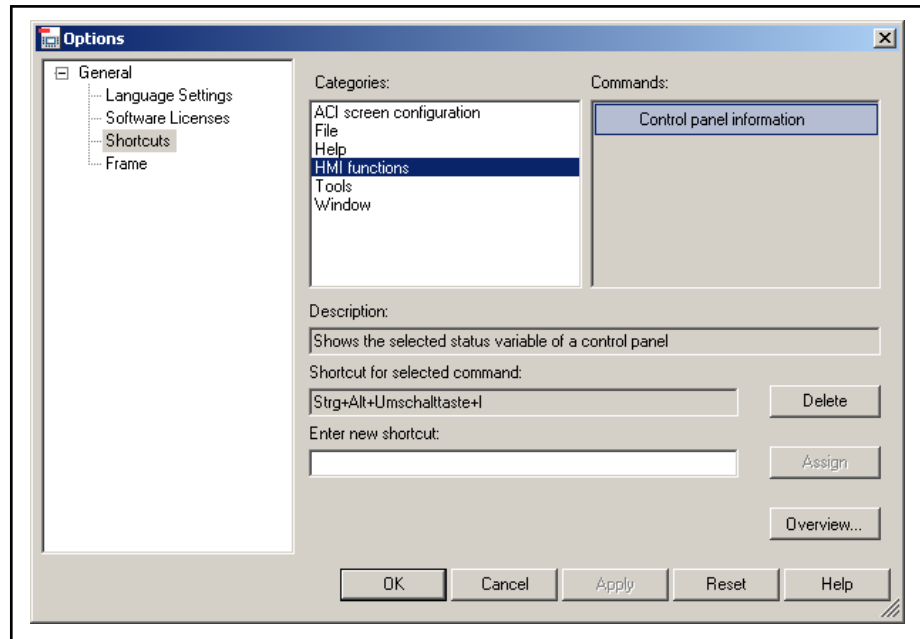


Fig. 4-30: Configuring the shortcuts to display user screen information

Operating screen for multi touch user interface

An operating screen for the multi touch user interface can only have 7 character.

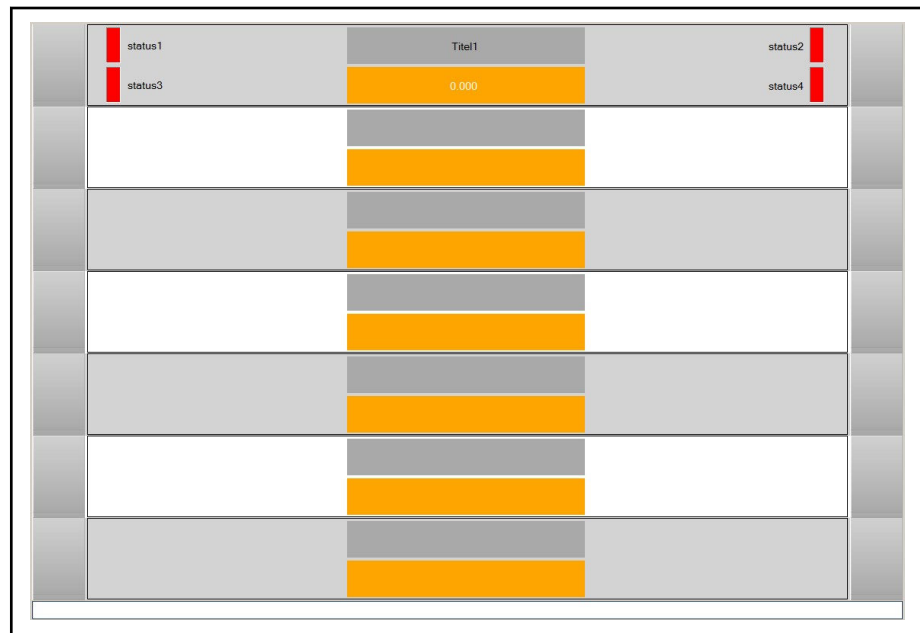


Fig. 4-31: Operating screen for multi touch user interface

The operating screen displays the variables used for the status display.

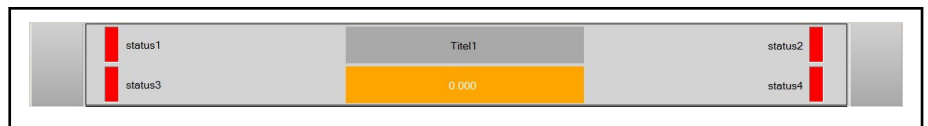


Fig. 4-32: Operating line for multi touch user interface -Variables

4.13 Criteria analysis window

The criteria analysis window is opened if the M-key assigned to a motion is pressed and the respective motion cannot be carried out.

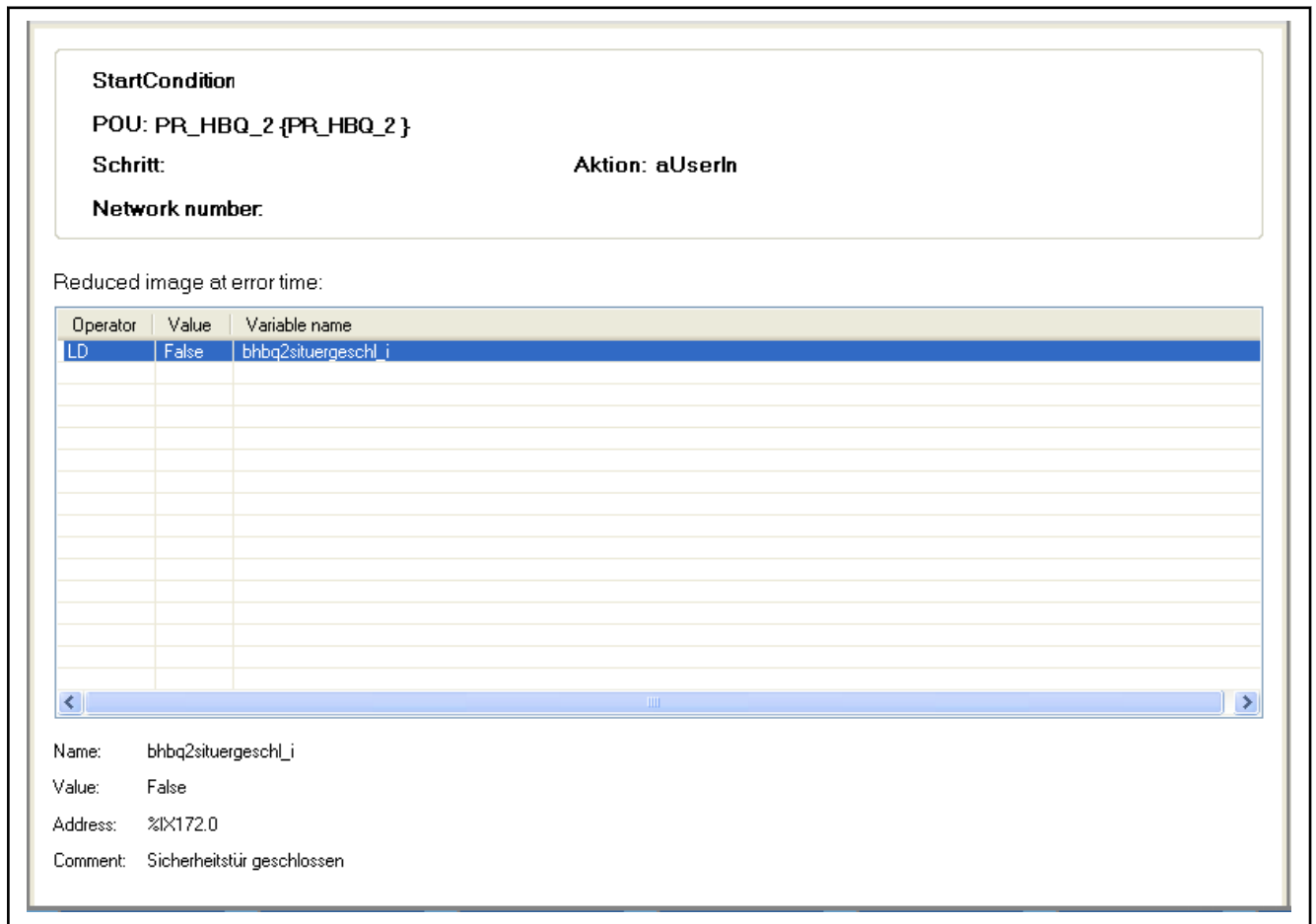


Fig. 4-33: Criteria analysis window

The criteria analysis window displays the Boolean variables preventing the motion from being carried out.

in the line "POU:", the name of the SFC is displayed.

The display is completed with <F9>.

4.14 Machine overview

The machine overview provides a list of all controls existing in a network configuration. The "Control" column shows the name of the network device. The "Status" column shows the state of the related control or of the connection.

The following values are possible:

- **Disconnected**
No connection established to this control

IndraWorks HMI user interface

- **Connected**
The connection to the control is established and there are no ProVi messages present
- **Warning**
A ProVi warning is present at the control
- **Error**
A ProVi error message is present at the control

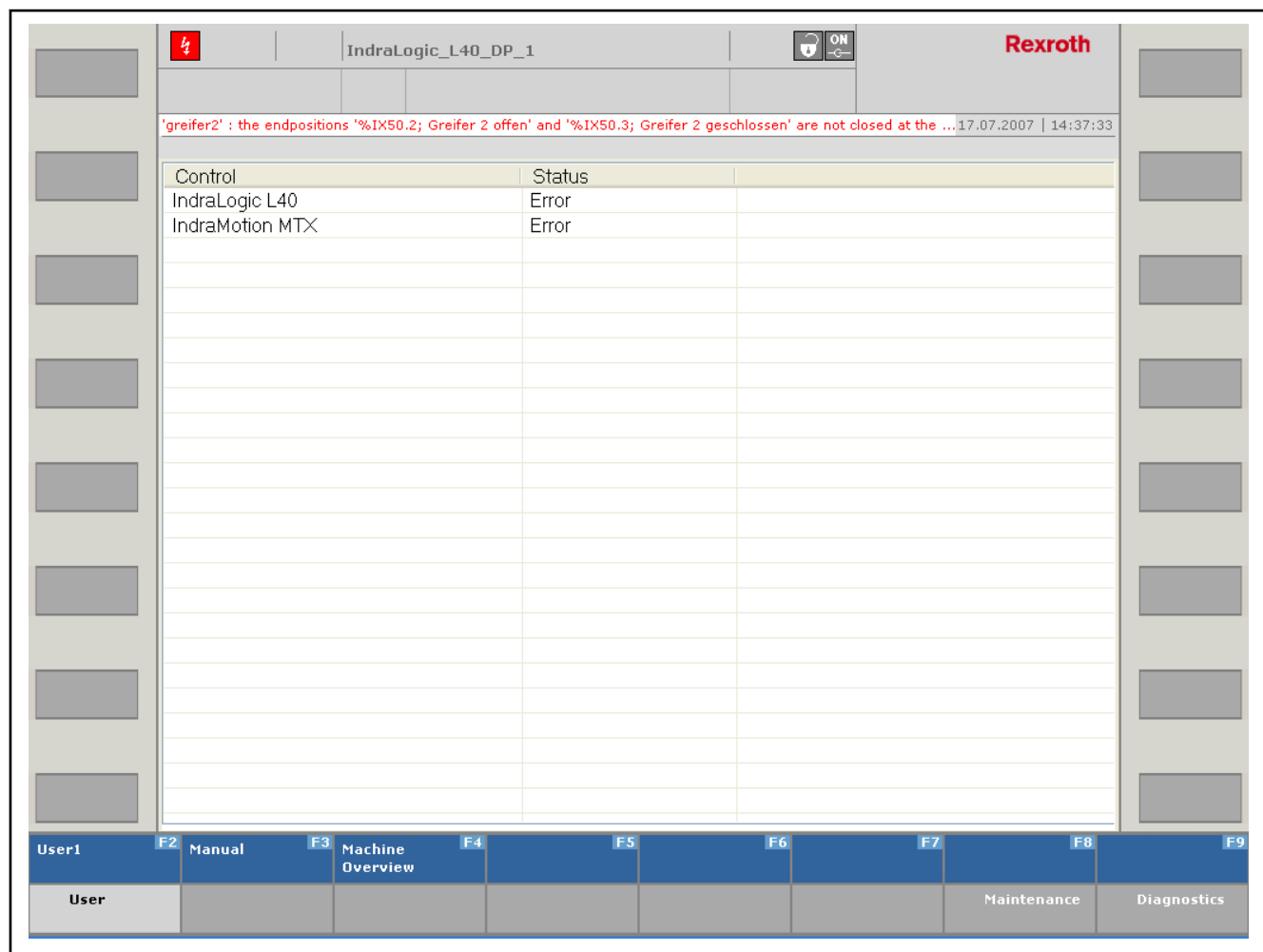


Fig. 4-34: Machine overview

Displaying diagnostic messages of a network control

If the status column displays "Error" or "Warning" as the state of the related control, call the diagnostic overview of this control to view the complete message text. Double-click on the desired control or select it by pressing the cursor keys and confirm with <ENTER>.

This opens the diagnostic overview which displays the ProVi messages of the control selected. An additional line at the top of the diagnostic window shows the name of the network control selected.

IndraWorks HMI user interface

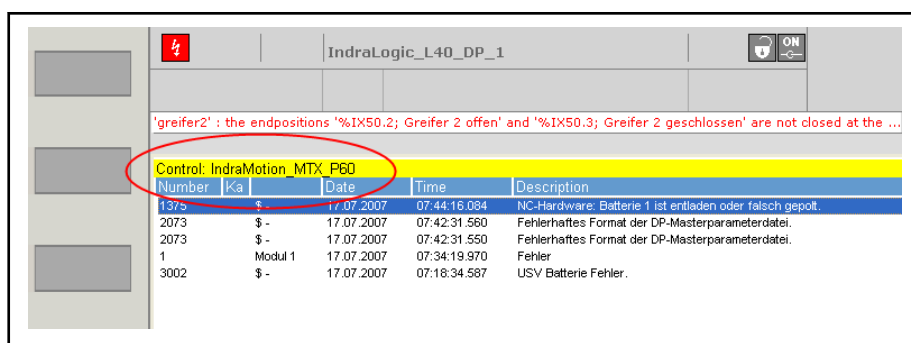


Fig. 4-35: Display of the control name on the diagnostic overview

If the diagnostic overview is exited and called again by pressing <OP9> "Diagnostics", the control originally assigned to the visualization device is displayed instead of the messages from the network control. The line displaying the control name is hidden.

4.15 ACI screens

4.15.1 Configuration in IndraWorks Operation

Brief description In IndraWorks Operation, a special configuration mode can be used to adapt the screen segments of the ACI screens and the control displays to the actual size of the ACI screens in the real user environment using the OP-, M- and F-keys and the header.



To enter the configuration mode of an ACI screen, define a shortcut to call it. Do so in the "Options" dialog that is located in IndraWorks Operation in the menu under **Extras ▶ Options...** In this dialog box, under "General/Shortcuts" of the Explorer, a shortcut has to be assigned to the command "Configuration screen layout" of the category "Configuration" (fig. 4-37 "Dialog: Options (Shortcut definition for the configuration mode)" on page 44).

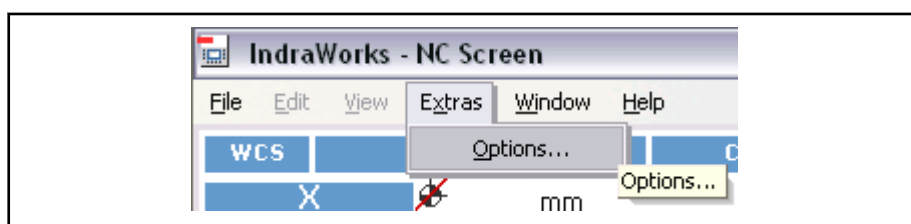


Fig. 4-36: Menu: Options (in IndraWorks Operation)

IndraWorks HMI user interface

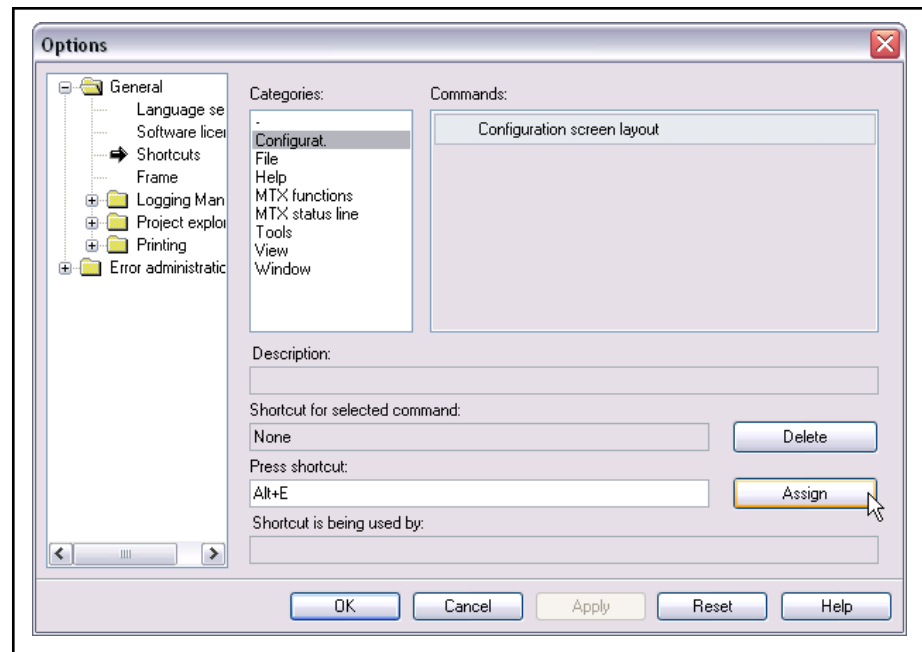


Fig. 4-37: Dialog: Options (Shortcut definition for the configuration mode)

Description

In the user interface, the controls in the ACI screens have process connection. As a result, they get a special visual appearance. To provide an overall visual appearance of the ACI screens, the size of their screen segments can be set in the configuration mode. The possibilities for setting the visual appearance of the controls, as well as further options, depend on the control's own configuration dialog.

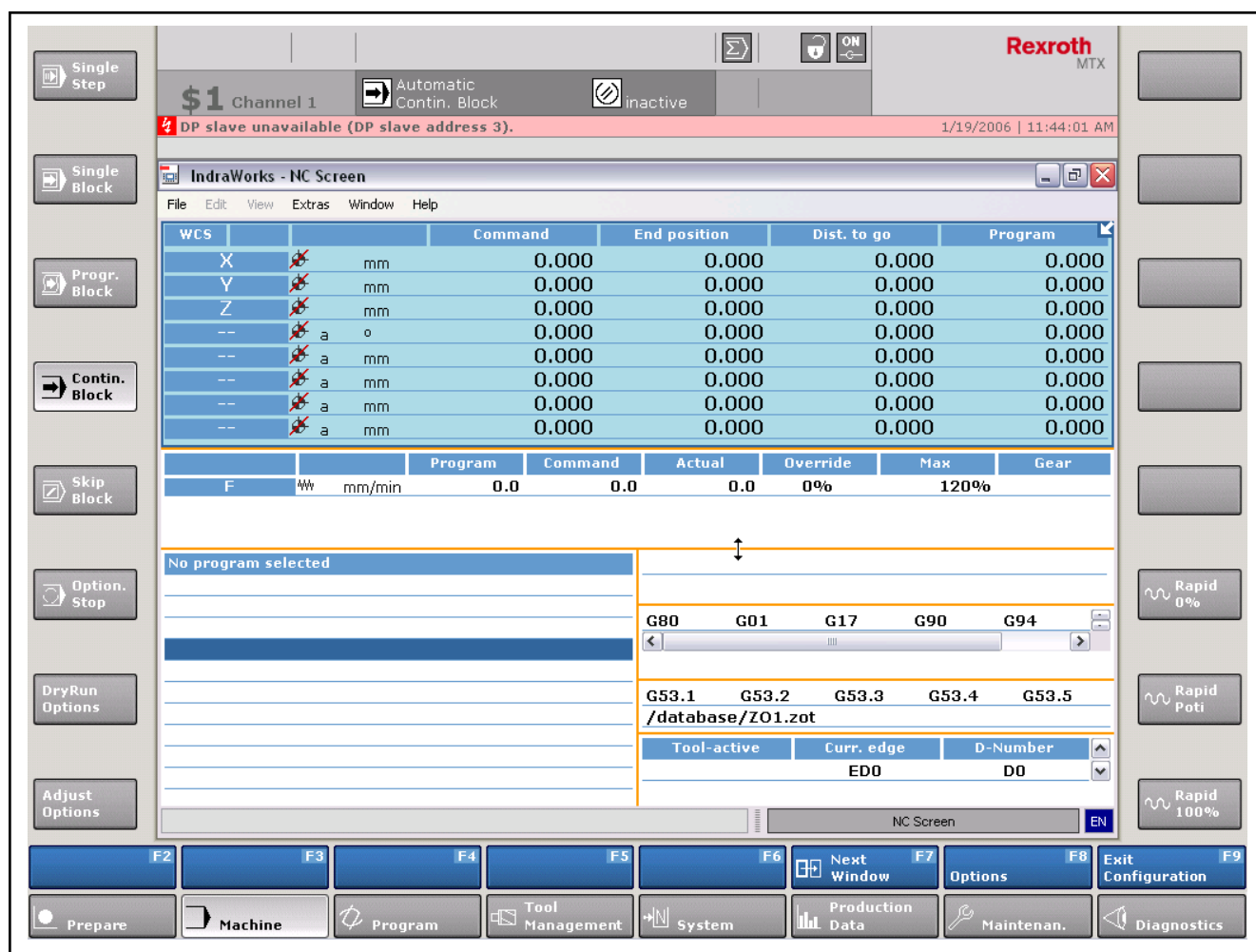


Fig. 4-38: IndraWorks Operation with ACI screen in the configuration mode

- **Dragging the screen segment dividing line:**

The size of the screen segments can be changed by moving their dividing lines. To do this, move the cursor to the dividing line and the cursor symbol changes. While holding the left mouse button down, drag the dividing line with the mouse.

The following dialogs and functions can be opened using the appropriate F-key:

- **Options:**

If the visible control of the highlighted screen segment has its own configuration dialog box, this is displayed.

IndraWorks HMI user interface

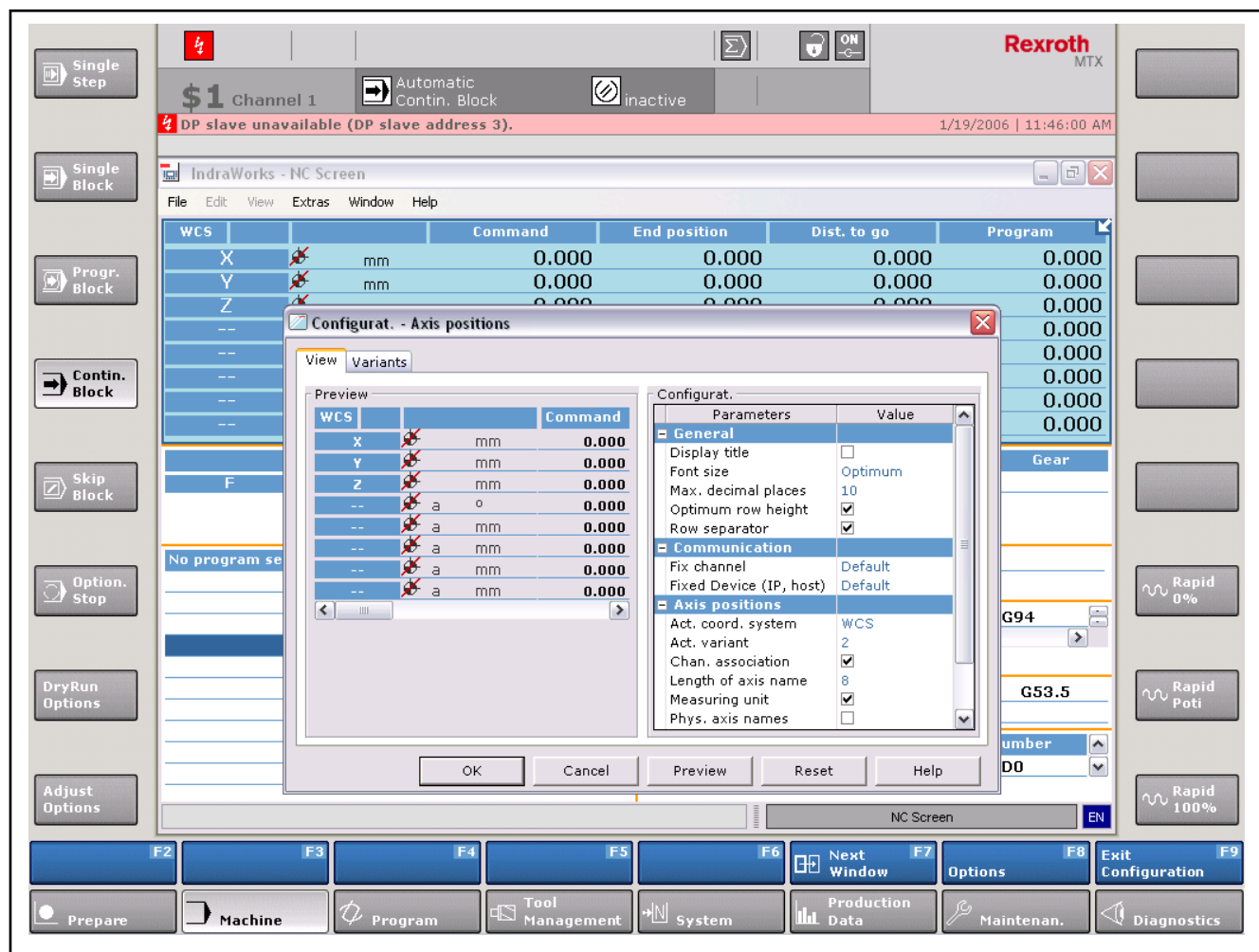


Fig. 4-39: Dialog: Example of a control-specific configuration dialog: "Configurat. - Axis positions" of the control for the axis position display

- **Next window:**

In the screen, the screen segment that follows the currently focused screen segment in the list of screen segments is focused.



The screen segments are arranged in a specific order in a list of screen segments. This order is specified due to the position of the screen segments in the screen - according to the rule "from the top to the bottom, beginning at the left-most" - and cannot be changed.

Screen segments can also be highlighted directly regardless of the order in the list of screen segments by moving the cursor to the screen segment and clicking with the left mouse button. This method of focusing does not work if the screen segment contains a control that is not correctly programmed.

- **Exit configuration:**

A security prompt appears asking whether the current layout of the screen is to be saved. Then the configuration mode is exited.



The security prompt appears even if no changes were made.

4.15.2 Integration of User Controls

Saving control settings

In the configuration mode, use the F-key <Options> to call the configuration dialog of the visible control in the focused screen segment. When the configuration dialog is closed, the identification text of the control is saved in the ACI screen via the ACI interface property "ConfigurationId" where the control stored its settings in its own configuration file.



The configuration dialog of the controls may also be called if the ACI screen is not in configuration mode. However, in this process the identification text of the control is not saved in the ACI screen. If the control changes the identification text, the changes made in the configuration dialog are only valid for the current instance of the control. When IndraWorks Operation starts again, the control has its original settings.

4.16 Version display

The version dialog that can be opened either via the **Help** menu in the toolbar and the menu item **Info** or in IndraWorks Engineering or in IndraWorks Operation display the version of the IndraWorks HMI installed. Clicking on the **Advanced** button opens a list of all components installed. Place the cursor line on the "IW-HMI ..." to display the exact version name of a component in the "Installed product + components" field.

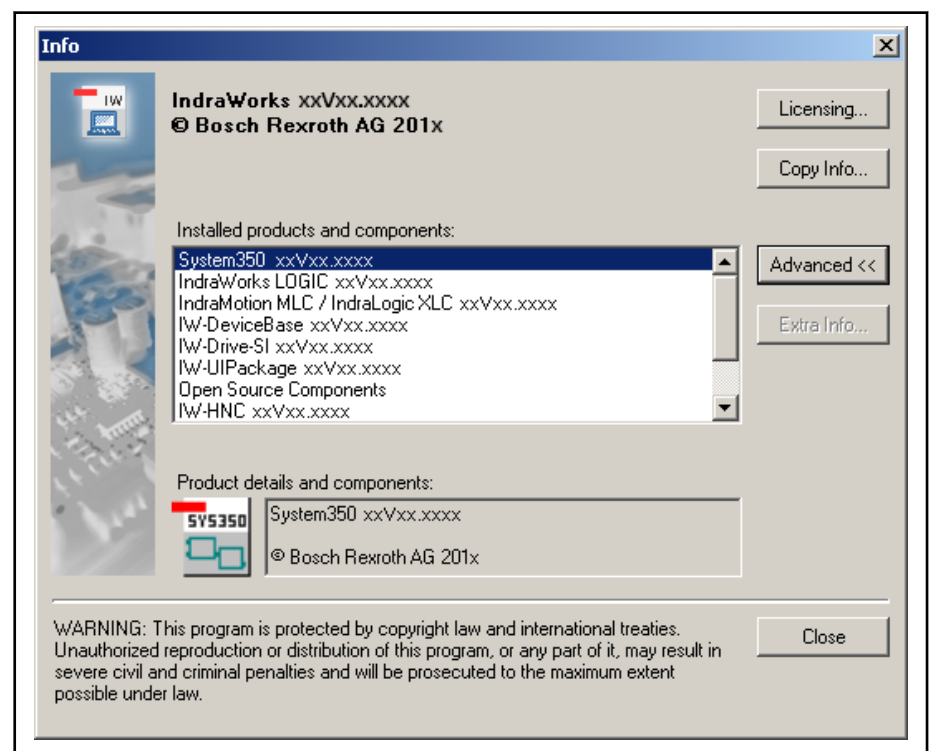


Fig. 4-40: Version display

Click on the **Copy Info** button to automatically create a list of all files installed. This opens the text editor providing information on the hardware used, the operating system and the version data of all IndraWorks components (see [fig. 4-41 "Detailed information on all files used" on page 48](#)). This process can take several minutes. The file created can be saved.

IndraWorks HMI user interface

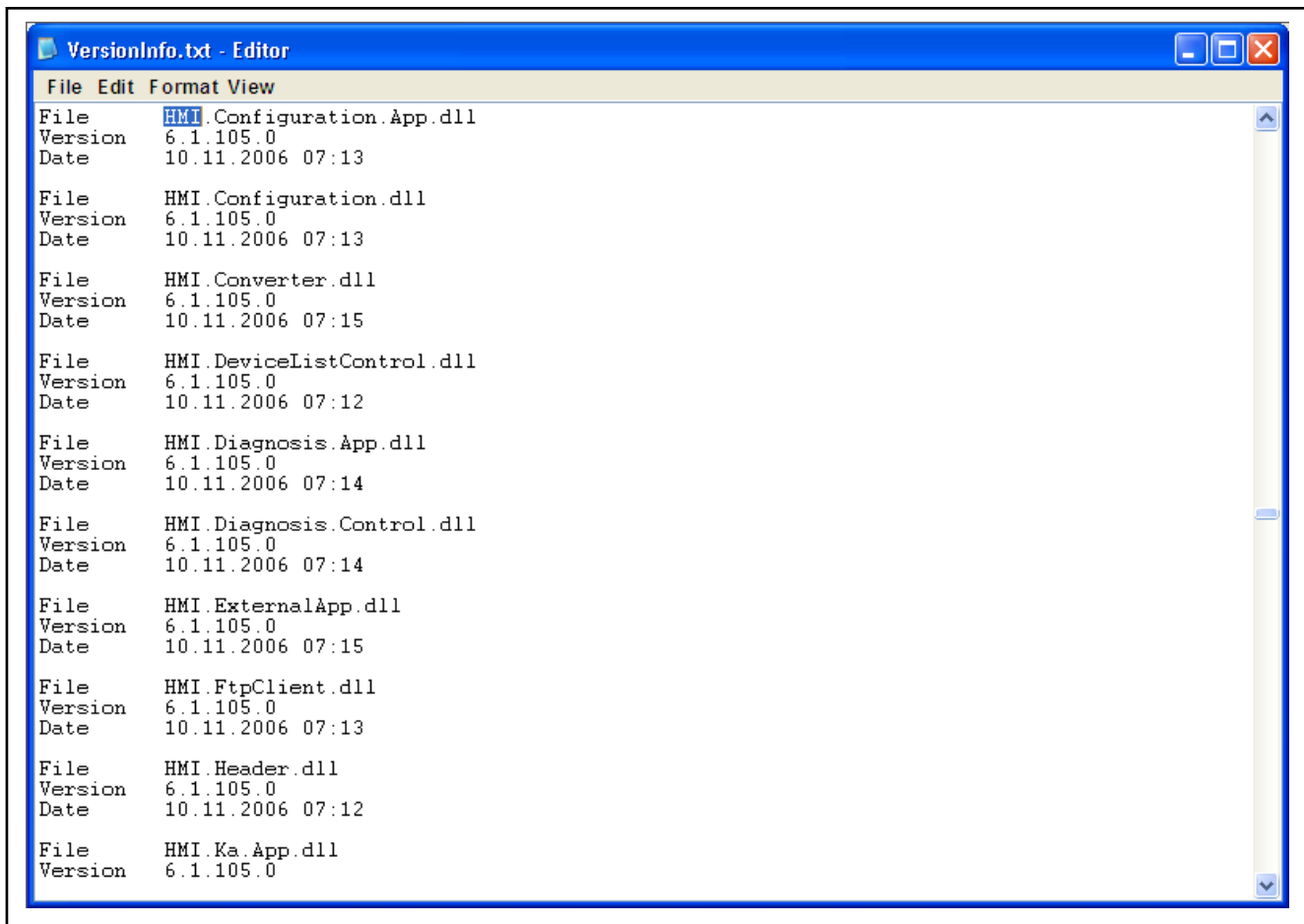


Fig. 4-41: Detailed information on all files used

5 Configuring the user interface

5.1 Overview

5.1.1 Creating the visualization device

IndraWorks HMI provides a graphical user interface to operate one or multiple controls and to visualize the machine state. The visualization is provided with preconfigured screens, so that fundamental functions are available after a few configuration steps. The scope of the preconfigured screens depends on the template selected.

In addition to the preconfigured screens, create individual screen configurations by selecting an application type and configuring the respective F-panel and M-panel assignments.

After a configuration, the created configuration files are transmitted to the target device on which the runtime system (IndraWorks Operation) is installed. The configuration files become active after a runtime system restart. The configuration can be modified and retransmitted at any time.

The screens and panels are components of a visualization device in the IndraWorks project.

Adding a visualization device

To add a new visualization device to the open project, switch to the "Library" window and open the "Visualization" tab. If the device library is hidden, it can be shown via the menu item **View ► Library**. Select the device group to which - the device to be inserted - is assigned and insert it into the project via drag-and-drop.

Alternatively, open the context menu of the project folder, go to the submenu **Add** and select the device group.

Select the device group corresponding to your real visualization device and select the respective device within this device group. The settings required with regard to display size, display resolution and operating system of the target device are made.

The "Visualization" tab contains the following device groups:

- VPx/VSx device
- VEH/VEP Device
- VCP/VCH device
- VR2x/VH2x
- VR3x/VH3x
- VR4x/VH4x

Drag the desired device group to the project in the Project Explorer (see [fig. 5-1 "Inserting a visualization device" on page 51](#)).

"VPx/VSx" device group

The "VPx/VSx" device group combines PC-based operator terminals.

The device group includes the IndraControl devices

- VPP 16
- VPP 40
- VPP 60

as well as the standard IndraControl PCs

- VSP 16
- VSP 40

Configuring the user interface

- VSP 60
- VDP 15
- VDP 16
- VDP 18
- VDP 21
- VDP 40

The overall range of function is available for this device group.

"VEH/VEP" device group

The "VEH/VEP" device group comprises the Embedded Terminals based on the Windows CE or Windows XP operating systems.

This group includes the IndraControl devices

- VEP15
- VEP 30
- VEP 40
- VEP 50
- VEH 30

These devices are configured in IndraWorks Engineering. The configuration is carried out quite similar as for the devices of the "VPx/VSx" group.

The following visualization elements are available for these devices:

- Screens
- F-panels
- M-panels
- OP-panel
- ACI screens
- WinStudio
- Operating screens

"VCP/VCH" device group

The device group "VCP device" comprises first generation compact operating devices.

The following visualization elements are available for these devices:

- VCH 05
- VCP 08
- VCP 11
- VCP 20
- VCP 25
- VCP 35

"VR2x/VH2x" device group

This device group includes the compact operating devices of the second generation.

The following visualization elements are available for these devices:

- VR21 04
- VR21 07
- VR21 09
- VH21

Configuring the user interface

- "VR3x/PR3x" device group** This device group includes the visualization device "VR3110" consisting of a display and an integrated PC box as well as different displays "DR00xx" combined with the PC box "PR31". If you have the PC box "PR31", select the respective combination from "PR31" and a display "DR00xx".
- "VR4x/PR4x" device group** This device group includes the visualization devices "VR4x15" and "VR4x21" consisting of a display and an integrated PC box as well as different displays "DR00xx" combined with the PC box "PR4x". If you have either the PC box "PR41", "PR42" or "PR43", select the respective combination from "PR4x" and a display "DR00xx".

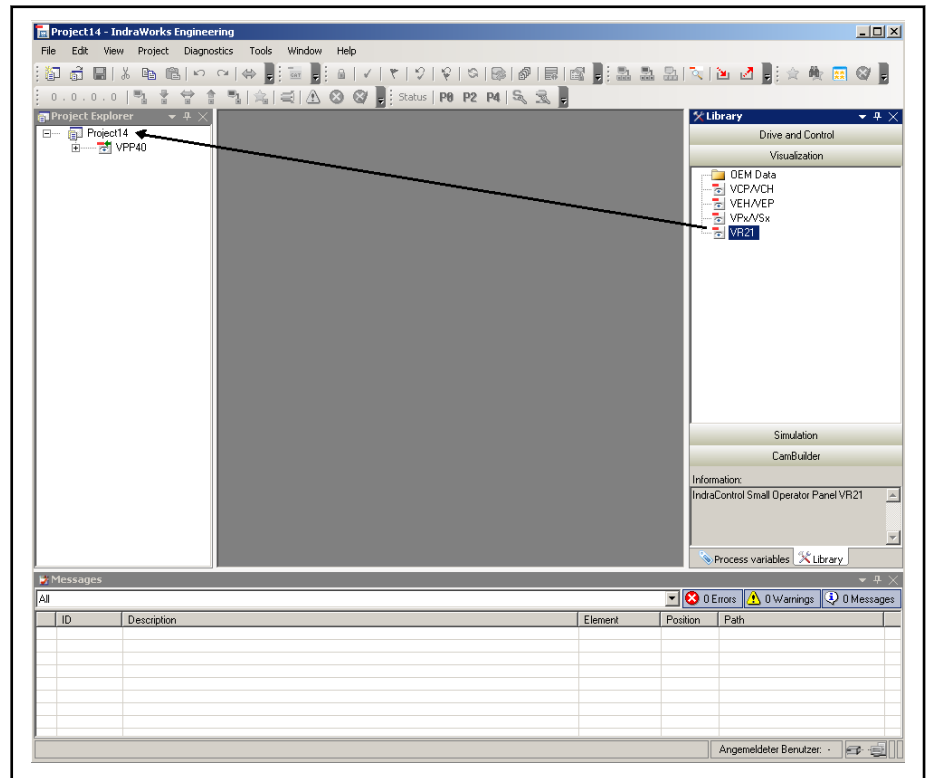


Fig. 5-1: Inserting a visualization device

5.1.2 HMI wizard

The HMI wizard opens after a device was added. It can be used to make basic settings for the visualization device.

Configuring the user interface

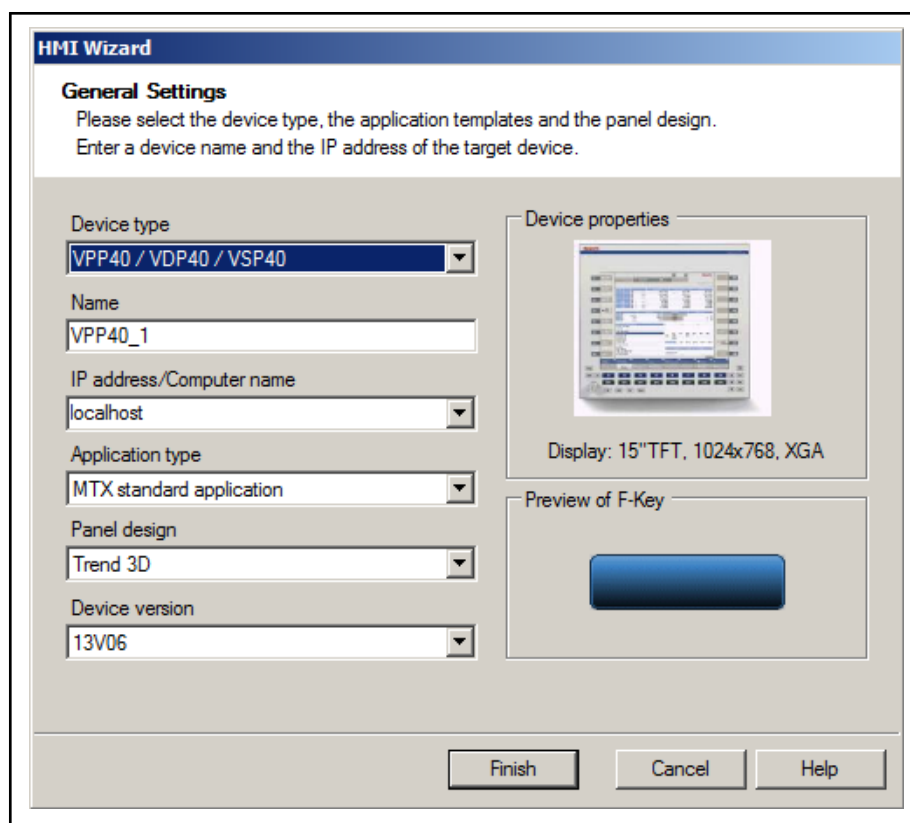


Fig. 5-2: HMI wizard

The following settings are displayed and can be edited if necessary:

- | | |
|--------------------------|--|
| Device type | Available device types in the device group selected: Select the desired device type to apply the appropriate device settings to the project. The "Device properties" field shows a preview of the device selected. It is not possible to change the device type subsequently. |
| Name | Name of the visualization device used to display the device in the Project Explorer. The name has to be unique. If a device with the same name already exists in the project, a warning is displayed. |
| IP address | IP address of the visualization device. The IP address is required to transmit visualization data to the target device. The IP address has to be unique within the project. |
| Application type | Configuration template for the visualization device. The templates of the application types differ with regard to the preconfigured screens, panels, etc. The "HMI standard application" is provided by default. This selection list provides further templates depending on the system installed. Templates can also be created for individual application types that can be used as initial configuration. It is not possible to change the application type subsequently. |
| Panel design | Graphic display of the F-keys, OP-keys and M-keys. The "Preview" field shows a single F-key with the selected design.

The panel design can only be selected for the application types HMI as well as the MTX standard applications. |
| Installation type | The device types "VDP15", "VDP18" and "VDP21", all VR21x devices as well as "VR4x21" and "DR0021 + PR31/PR41" can be used for horizontal as well as vertical installation. For these device types, a selection option is available. The installation type for the device for which a visualization is to be generated |

can be specified. The visualization data resolution is specified during the selection. The setting cannot be changed anymore subsequently.

Complete the settings with **Finish**. The wizard-guided basic settings are completed. The new visualization device appears in the Project Explorer. The settings can be changed subsequently (see [chapter 5.1.3 "Context menu of the visualization device" on page 53](#)).

If **Finish** is disabled, the device name is either not unique or the IP address is incorrect.

5.1.3 Context menu of the visualization device

Open the context menu by right-clicking on the HMI node in the Project Explorer:

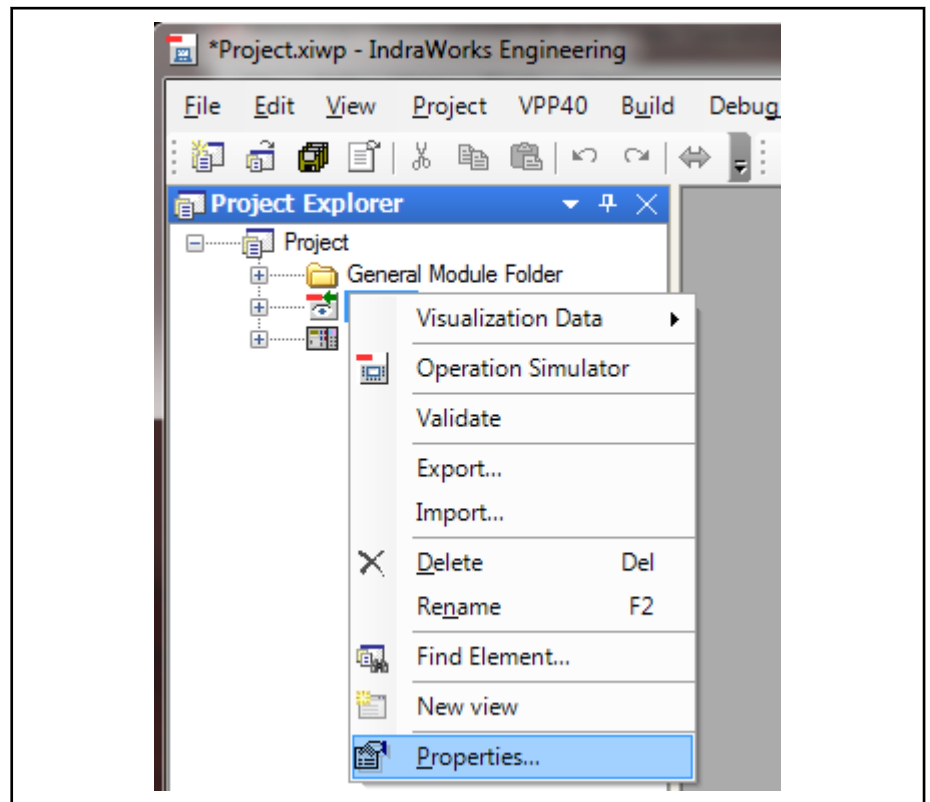


Fig. 5-3: Opening the context menu

The following functions are available:

- **Visualization data**
The current visualization data is transmitted to the operating station (see [chapter 5.1.5 "Enabling the configuration data" on page 60](#))
- **Operation Simulator**
To check the exact behavior and the design of an application in IndraWorks Operation (OP) and to be able to adjust the details, the application has to run in the required resolution in IndraWorks Operation. A device on which the OPD is running is often not available when developing an application. Thus, the OPD can run in an environment (HostedOperationDesktop) simulating the resolution of the target devices.
- **Validate**
Check for consistent visualization data (see [chapter 5.21 "Validating visualization data" on page 193](#))

Configuring the user interface

- **Update HMI Data**
The system checks whether the Project Explorer contains an element for each of the HMI configuration data
- **Export**
Individual elements (e.g. F-panels) are saved to a file
- **Import**
Elements saved to a file (e.g. F-panels) can be inserted
- **Delete**
Deletes the visualization device from the project. All related configuration data is deleted
- **Rename**
Changes the name of the visualization device in the Project Explorer
- **Find Element**
Searches a specific element in the project
- **New View**
Opens a new tab for the visualization device
- **Properties**
Opens a configuration dialog used to subsequently edit the properties of the visualization device

For IndraControl VEP and VEH devices, the context menu provides the following additional function:

- **Install Software**
All executable files required are transmitted to the visualization device (see [chapter 5.1.5 "Enabling the configuration data" on page 60](#))

5.1.4 "Properties" dialog

Change basic settings, the communication settings and the advanced settings in this dialog. Use the "Apply" button to save all properties of the visualization device.

"General" tab Device name, IP address and panel design can be changed in this dialog. The device and application type determine fundamental properties of the visualization device and cannot be changed later on.

Configuring the user interface

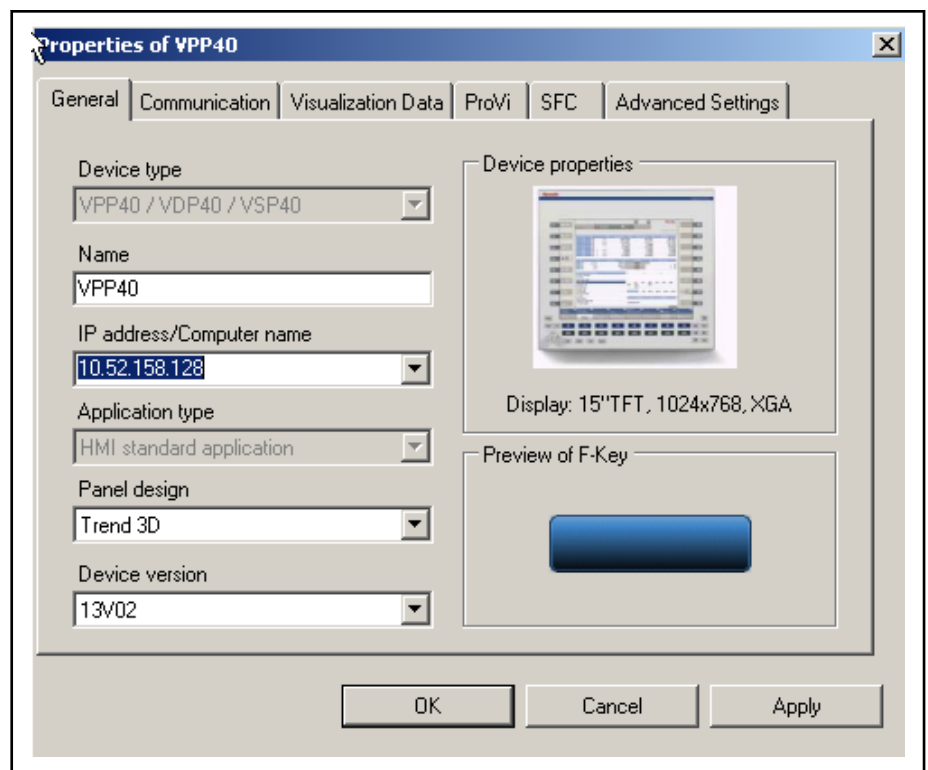


Fig. 5-4: Properties – General

"Communication" tab

Control:

Control whose diagnostic data is displayed in IndraWorks Operation. The selected control is also used as default setting in F-panels and M-panels as well as in operating screens.



If the selected control is changed under "Control" and the "Properties" dialog is closed with **OK**, all references in F-panels, M-panels and operating screens are automatically updated.

Instance of the M-key FB:

Name of the function block instance in the PLC program.

Language of the visualization device

When adding the relevant variables to the PLC program, the language of the visualization device can be synchronized with the PLC program. This function is only available if the device version (refer to tab "General") is "14V16" or higher.

Currently set language: The entered variable is updated by IndraWorks Operation upon each language switching operation at runtime. The corresponding language ID code, e.g. "de-DE" or "en-US" is transmitted as value.

Language selected in the control: The PLC program can write the ID code of a language the visualization device is to be switched to in this variable. This is activated using a trigger variable that can be defined in the *Activate preselected language* field. As soon as the trigger variable assumes the value "TRUE", the visualization device activates the preselected language and resets the trigger variable to the value "FALSE".

Configuring the user interface



If the preselected language has an invalid language ID or if no language package is installed, the visualization device displays a message upon activation.

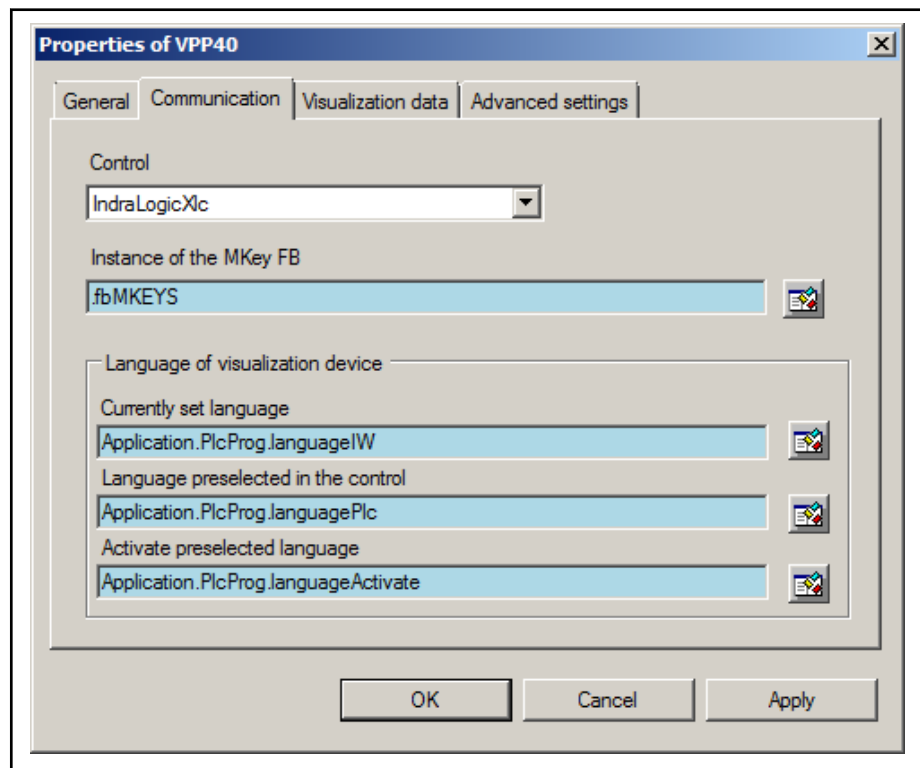


Fig. 5-5: Properties - Communication

"Download options" tab

Automatically transmit diagnostic data:

If the option "Automatically transmit diagnostic data" is selected, a program-controlled download of an HMI runtime project is executed if data (symbols, diagnostic information) has changed in the assigned PLC. Only apply this setting if the operating device is accessible and the diagnostic data can be transmitted automatically.

Configuring the user interface

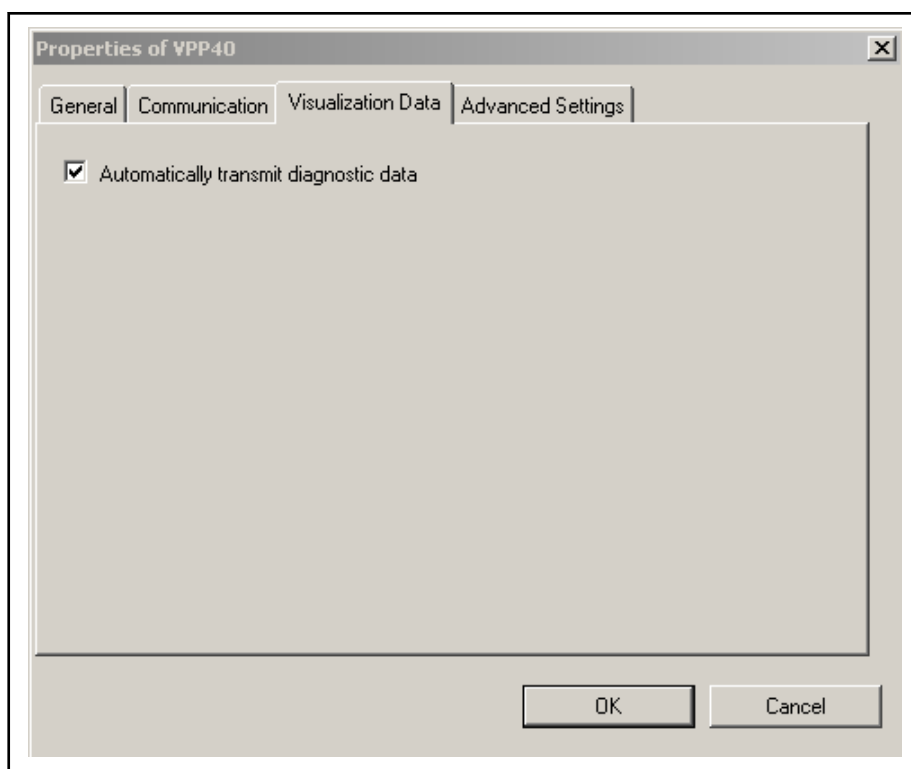


Fig. 5-6: Properties – Visualization data

"ProVi" tab In this dialog, configure which diagnostic ProVi message type of which modules are displayed at runtime.

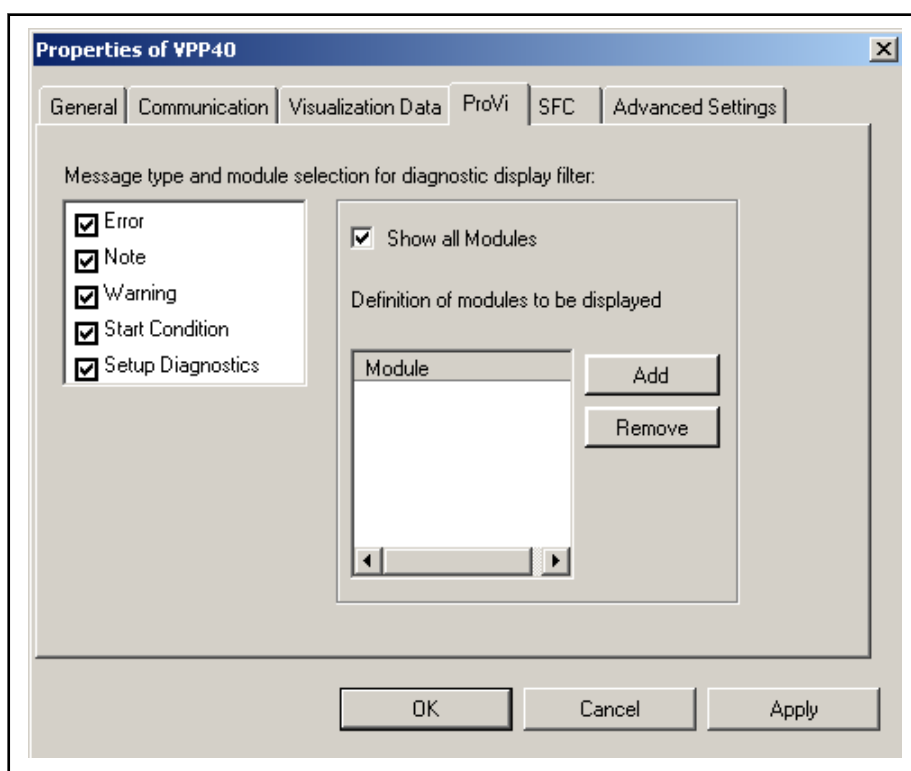


Fig. 5-7: Properties - ProVi configuration

Configuring the user interface

ProVi type:

Available message classes.

Modules:

Select module number or module number range. Enter the module number or module number range with **Add**. All modules from 0 to 99 are provided. It is also possible to enter a module number range (e.g. 40-60). If an entry is highlighted, it can be deleted with **Remove**.

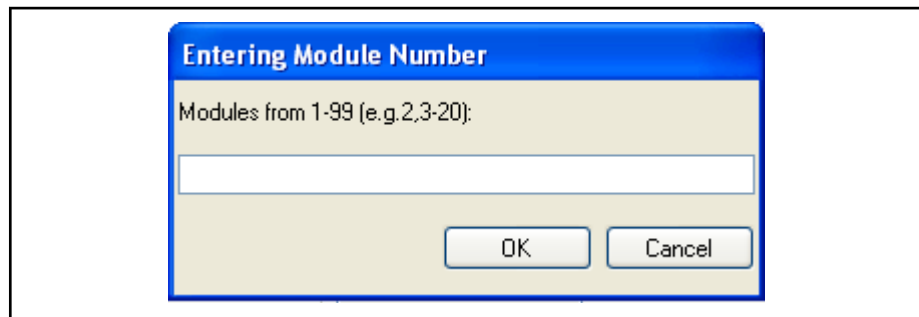


Fig. 5-8: Entering a module number

An error message is displayed if the entry is incorrect:

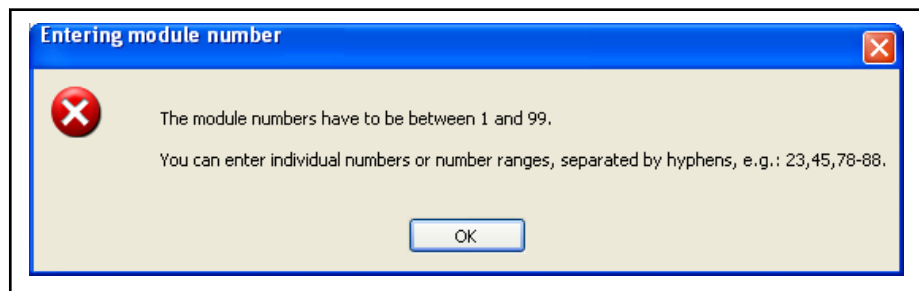


Fig. 5-9: Error message for incorrect module number entry

"SFCs" tab

In this dialog, configure which diagnostic SFC message type of which modules is displayed at runtime.

Configuring the user interface

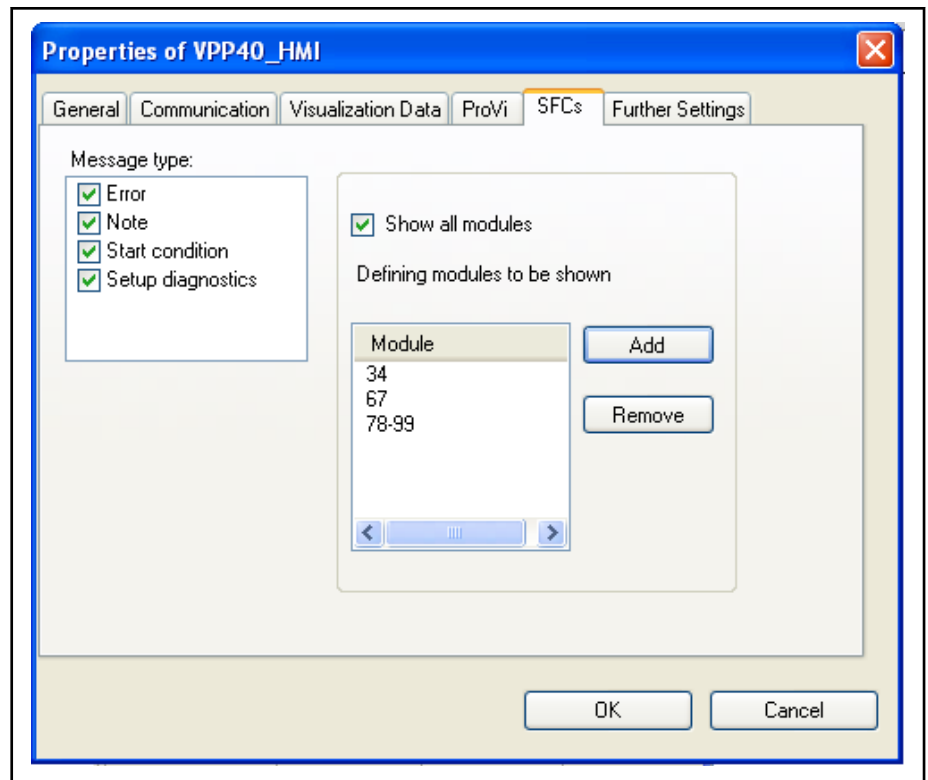


Fig. 5-10: Properties - SFC configuration

Message type:

Available message classes.

Modules:

Select module number or module number range (for more message, see "ProVi tab" on page 57)

"Advanced Settings" tab

WinStudio project:

Name of the project file used for visualization.

Device identification station name:

Name of the HMI device. If the "Station name" option is enabled, enter a name for the HMI device in the selected project language.

Is displayed in the IndraWorks Operation header. If this is not the case, the name of the assigned control is displayed.

Configuring the user interface

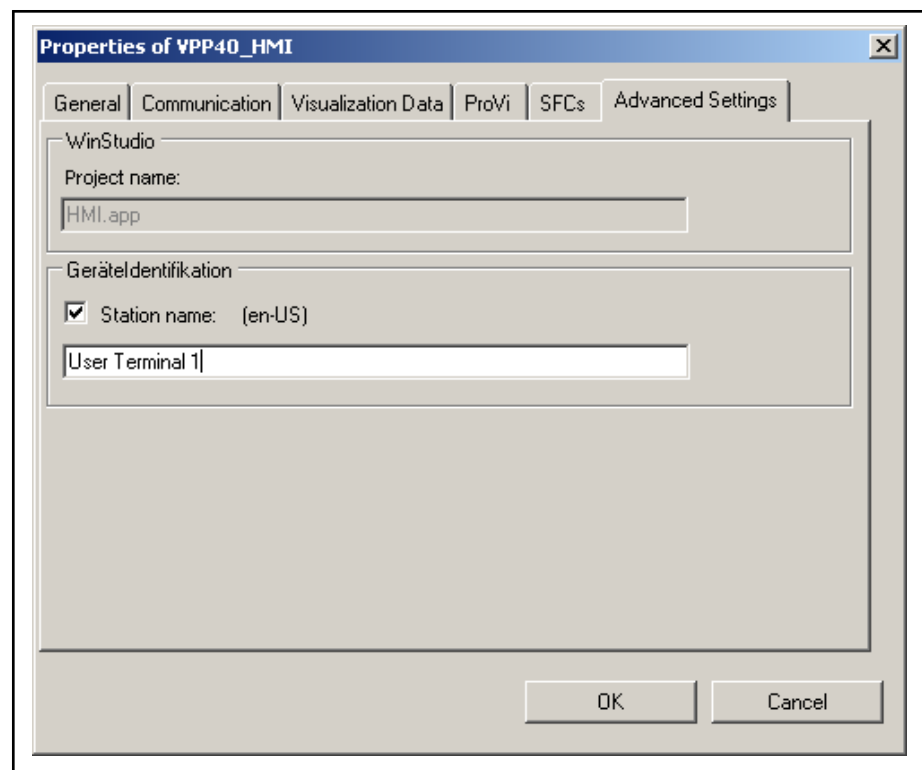


Fig. 5-11: Properties - Advanced Settings

5.1.5 Enabling the configuration data

Overview

Activation of the configuration data for IndraWorks Operation and/or WinStudio applications differs according to the selected application type.

IndraWorks Operation applications for VPx/VSx and VEH/VEP (XPe) devices

Visualization data as runtime project

The runtime project contains all visualization data relevant for the operating station. The data is extracted from the engineering project and loaded as encapsulated runtime project to the target device. Thus, it is referred to as HMI runtime data as well.

Transmission of runtime data is called via the context menu of the visualization device in the IndraWorks Engineering Project Explorer.

Configuring the user interface

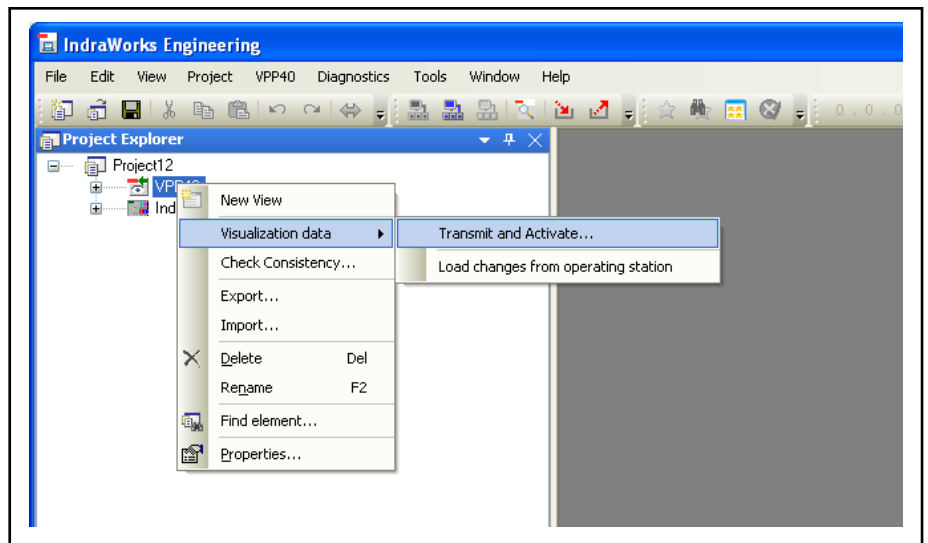


Fig. 5-12: Transmit and activate visualization data

The following dialog appears before the runtime project is transmitted:

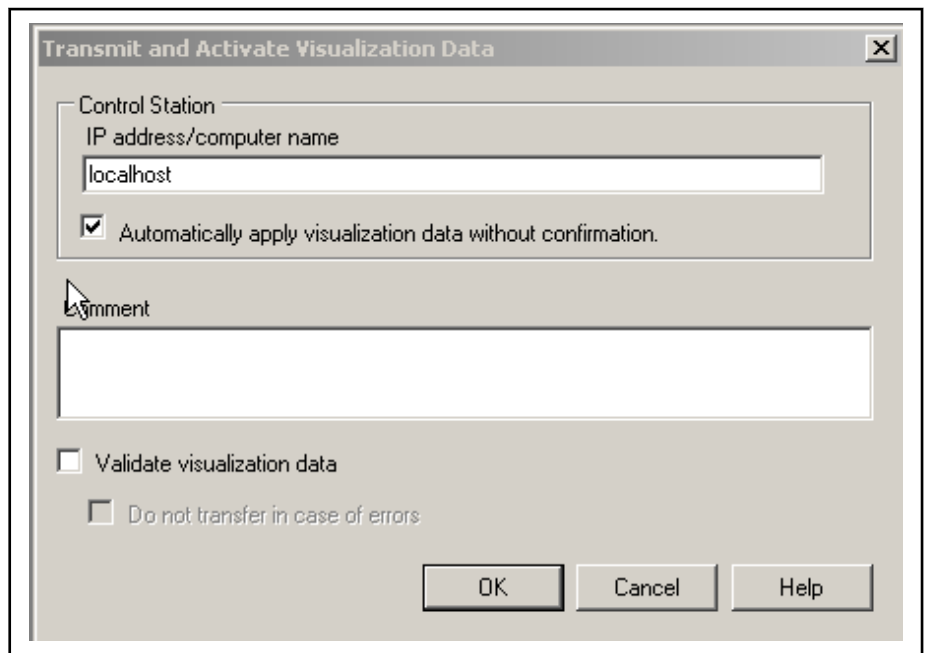


Fig. 5-13: Transmitting visualization data

The project engineer can specify the address of the target device (the address from the "HMI Device Properties" dialog is preset) and enter a comment on the runtime project. It can be specified optionally that a message appears on the operating station at IndraWorks Operation runtime as soon as data is received. Pressing **OK** completes the procedure. **Cancel** ends the procedure without a result. The **Help** button opens the respective chapter in the help. If the system detects that another HMI project configuration is active on the target operating station, the following query dialog is displayed:

Configuring the user interface

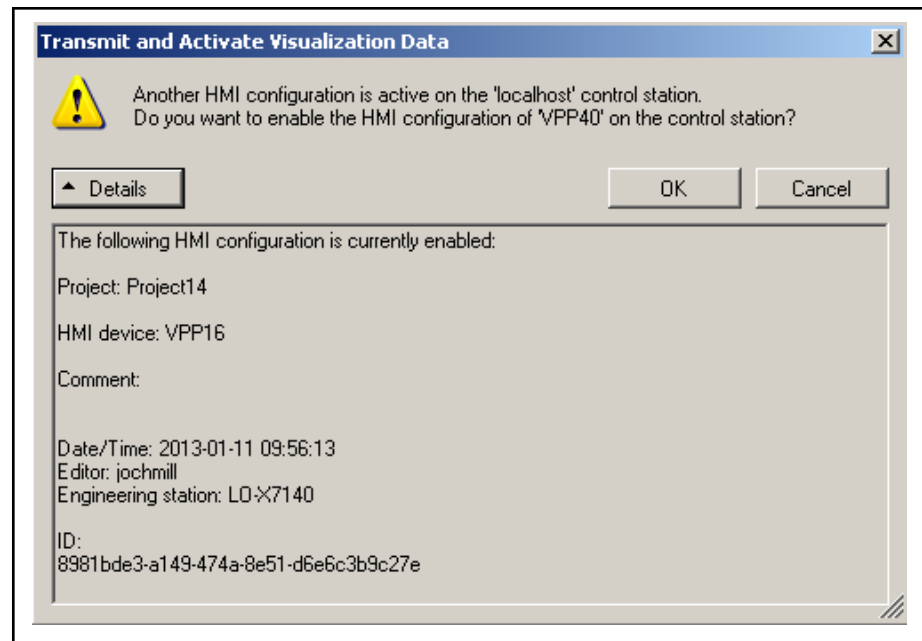


Fig. 5-14: Prompt to the project engineer when the HMI project configuration on the operating station is changed

If "Yes" is selected, the transmission starts. "No" aborts the process. The transmission starts immediately if there is no active HMI project configuration on the operating station or if it is from the same HMI device from the same IndraWorks project.

Behavior on the operating station

During a running IndraWorks Operation, type and scope of modifications are checked and it is reacted accordingly. In best case, the modifications can be applied without closing and restarting the desktop. If the machine operator rejects the application of new runtime data, an icon is displayed as note in the header. In any case, the new runtime data is applied when restarting IndraWorks Operation.

Case 1: Runtime project and archive do not match. That means that they originate from different engineering projects or contain visualization data of different HMI devices of the same engineering project. Restart to apply data.

Case 2: Runtime project and archive do match. That means that the visualization data originate from the same HMI device of an engineering project. The modifications are that major that they can only be applied after a desktop restart.

Case 3: Runtime project and archive match. The changes are minor. The Operation desktop does not have to be restarted.

The following modifications in the HMI configuration do not require an IndraWorks Operation restart and become effective immediately after transmission:

1. F-panel
 - Key representation (text and graphic)
 - Position of texts and graphics
 - Functional change
 - PLC changes (disabling/hiding an F-key and writing a variable)
2. M-panel
 - Key representation (text and graphic)

Configuring the user interface

- Position of texts and graphics
 - Functional change
 - PLC changes (disabling/hiding an F-key and writing a variable)
3. OP-panel
 - Key representation (text and graphic)
 - Position of texts and graphics
 4. Screen configuration
 - Screen number
 - Panel assignment
 - Hiding header
 5. WinStudio screens
 - Screen contents

"Visualization data -> Load changes from operating station":

This function transfers user data relevant for the operating station to the IndraWorks project. This should be carried out before transmitting the runtime project to ensure data consistency.

WinStudio applications for VPx/VSx and VEH/VEP devices

Activation of the WinStudio configuration and transmission of WinStudio Runtime is carried out via the WinStudio target platform.

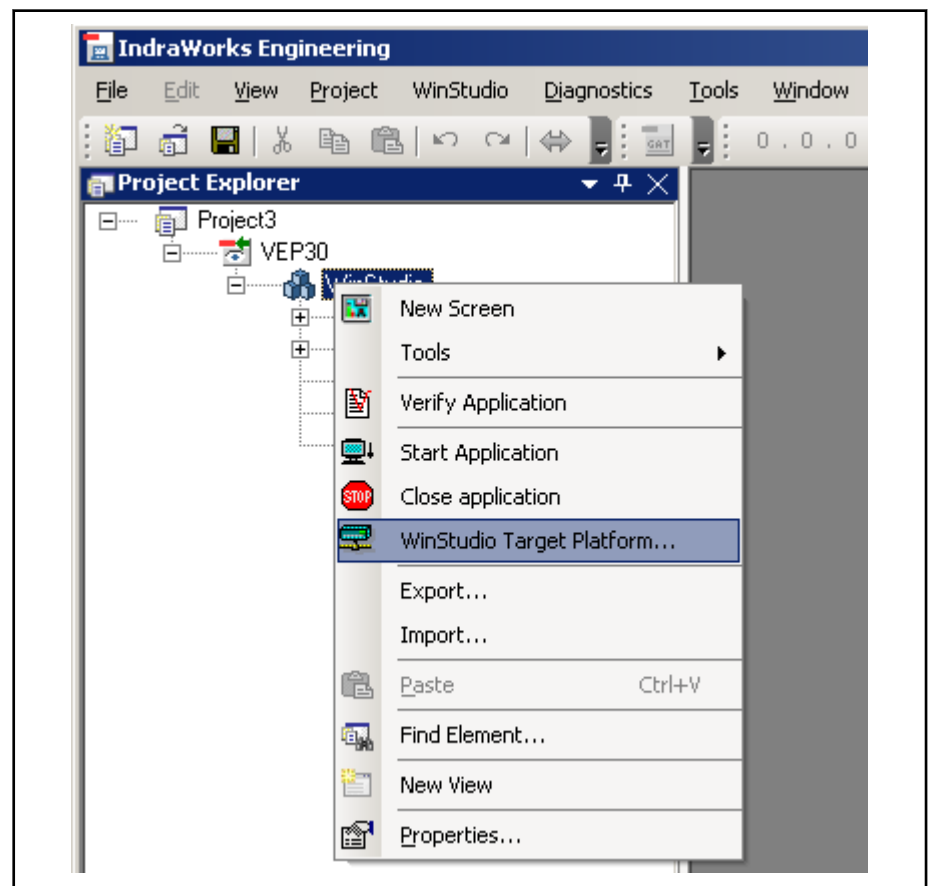


Fig. 5-15: Calling the WinStudio target platform

Install software

To install the WinStudio software for the runtime environment on VPx/VSx and VEH/VEP devices, start the WinStudio target platform and enter the IP

Configuring the user interface

address of the visualization device into the field "Network IP" in the "Target" tab. Then, establish a connection to the target device using the "Connect" button. After the connection was established successfully (the WinStudio remote server has to have started on the visualization device and has to be accessible via the network), click on the "Install System" button to install the WinStudio software.

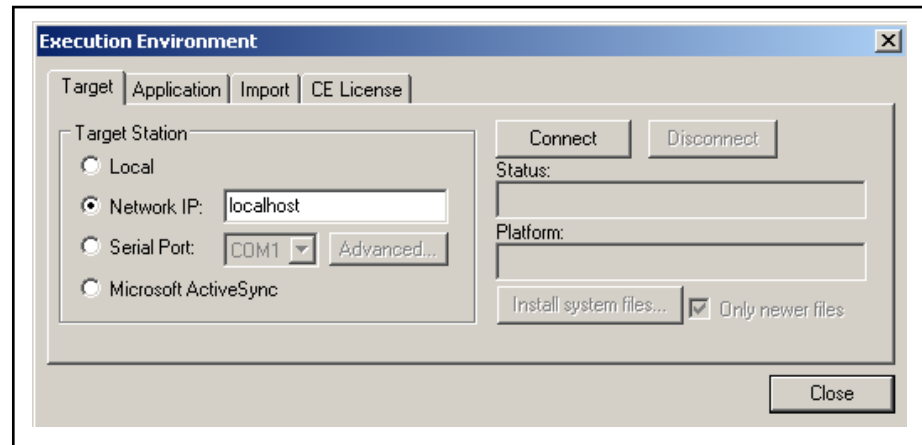


Fig. 5-16: WinStudio target platform

Transmitting and starting visualization data

After the visualization device has been configured, all respective configuration data has to be transmitted to the target device. For this purpose, call the function **Send to target** in the "Application" tab of the WinStudio target platform. Then, the application may be started both via the target platform ("Start" button) and via the remote agent on the target device.

Download status

Downloading is necessary whenever visualization device was added, deleted or edited.

The data is to be downloaded in the following cases:

- New creation of a visualization device
- Modification of screens and panels
- Modifications in language-dependent user texts
- Modification of user management data
- Modification of operating/WinStudio screens and settings
- Modification in the PLC program (symbols, diagnostic data)

A green arrow in the icon of the visualization device indicates that a download is required (see following figure).

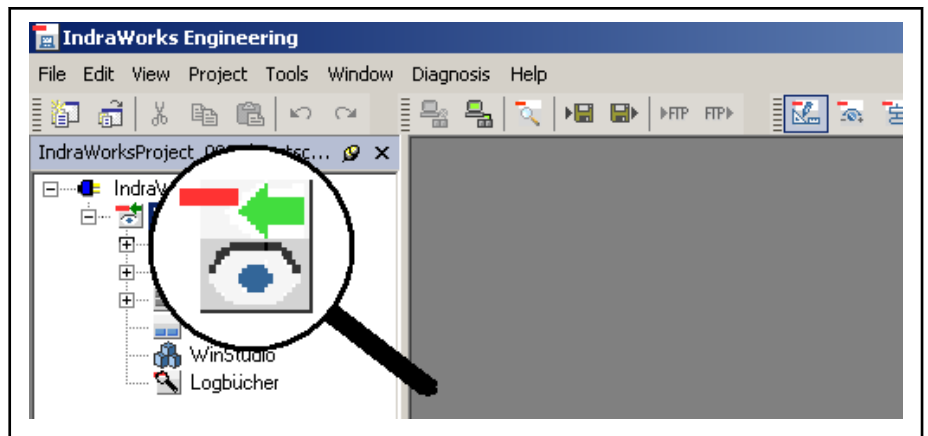


Fig. 5-17: Display of a required download

The arrow disappears immediately after the download, thus indicating that the project data matches the data on the target device.

5.1.6 Structure of the visualization device

The user interface of a visualization device is based on the configuration of so-called "screens". A "screen" consists of an application used to visualize specific machine data (e.g. "diagnostic overview") and specific F-panel and M-panel assignments, so that functions of the displayed application can be called using F-keys or M-keys.

The configuration data of a visualization device is subdivided into the following in the Project Explorer:

- **Screens**
Folder for the configuration of screens and operating areas
- **Screen Frame**
Folder for the configuration of F-panels, M-panels and OP-panels
- **Screen Application**
Folder for the configuration of applications
- **Logbooks**
Folder with all configured logbooks

The "Screens" folder contains all of the preconfigured screens and the screens additionally configured by the user. The screens are subdivided into further subfolders, so-called "operating areas".

The "Screen Frame" folder contains the configurations of the different panels:

- **F-panels**
Folder for system-specific and configured F-panels
- **M-panels**
Folder for system-specific and configured M-panels
- **OP-panel**
Configuration of OP-keys and operating areas

The "Screen Application" folder contains the editors of the applications displayed in the individual screens:

- **WinStudio**
Folder for WinStudio screens which can be freely configured by the user. Further configuration tools for WinStudio screens are located in further subfolders

Configuring the user interface

- **Operating Screens**

So-called "operating screens" can be configured in this folder. These are used to visualize and operate axes

- **Application-specific folders**

Additional folders (e.g. "Tool Management Screens") are created depending on the application type

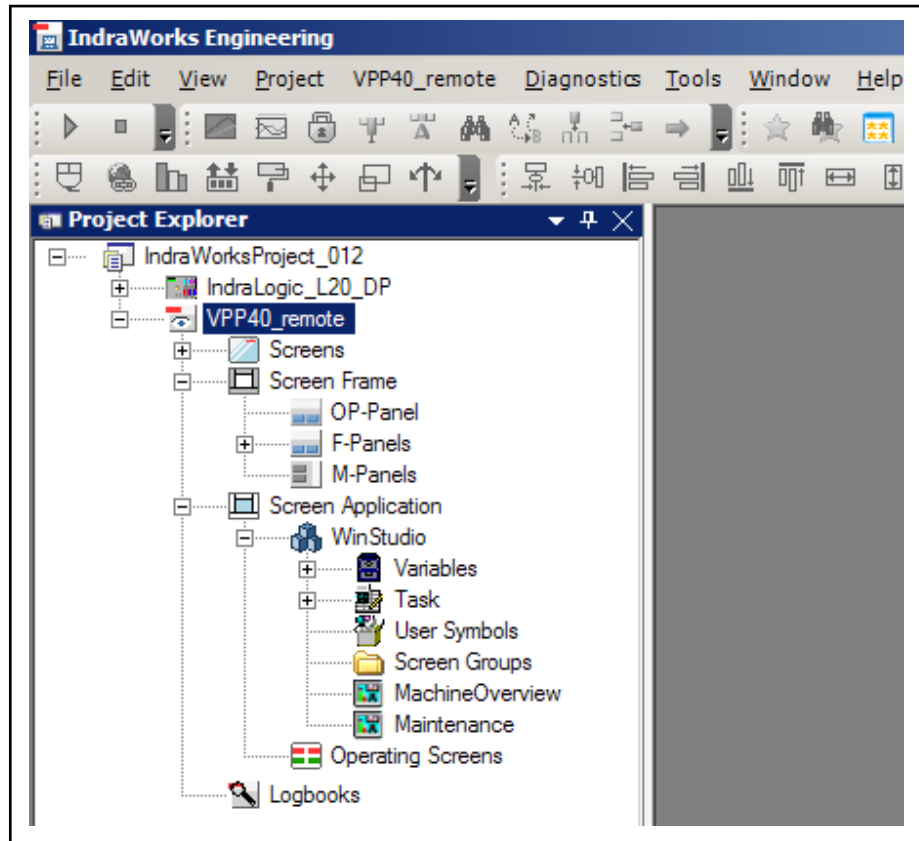


Fig. 5-18: Subfolders of an HMI project

5.2 Configuring screens

5.2.1 Overview

A screen is defined as a visible application within IndraWorks Operation (for example "Diagnostics", "WinStudio screen").

Screen creation

The configuration of a screen comprises the following steps:

- Create a screen configuration
- Application selection, e.g. operating screen, diagnostics, external Application
- Configure the screen application
- Select or create and configure F-keys
- Select or create and configure M-keys
- Enter screen number
- Assign operating areas (only for VPx)
- Configure screen management (optional)
- Adjust layout settings (optional)

It is also possible to define a screen as start screen that is displayed after the runtime environment start.

5.2.2 Screen editor

The screen editor is used to configure all parameters of a screen. It displays a preview of the screen content as well as the panels assigned.

Creating a new screen

Right-click on the "Screens" subnode or a subfolder (only VPx) to create a new screen.

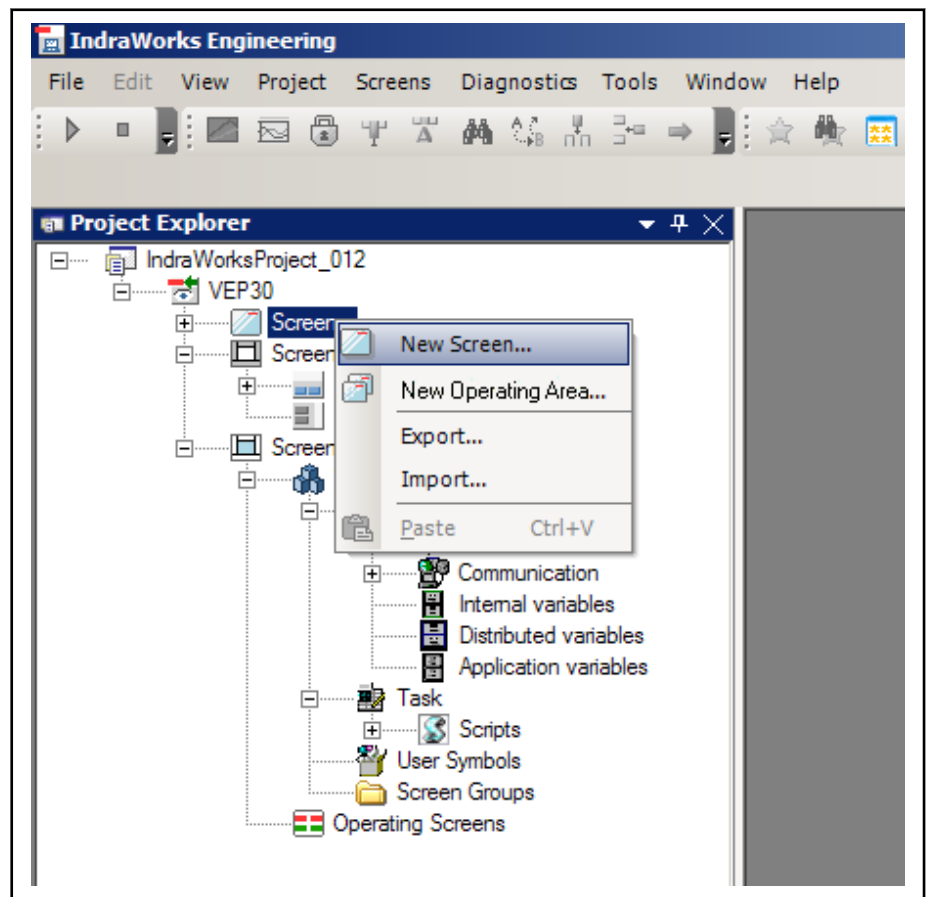


Fig. 5-19: Creating a new screen entry

Select the **New Screen...** menu item, the following input dialog then opens, in which any name can be entered. This name is used to identify the screen in case of links e.g. with an F-key. The "File" field shows the storage file of the screen information. A fixed file name is specified. After confirming with **OK**, a new entry is added to the "Screens" folder or to the selected subfolder.

Configuring the user interface

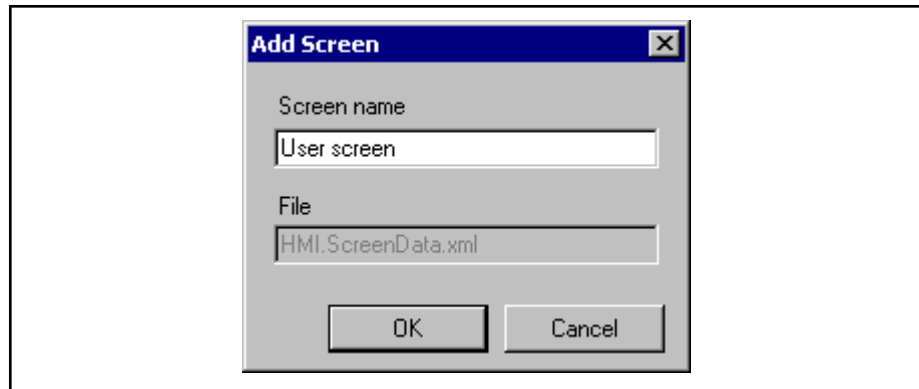


Fig. 5-20: Dialog to enter a screen name

Editing screen

The screen can be configured by double clicking on the added elements. This opens the screen editor with the current settings and shows a screen preview in the right area of the window. When a new screen is created, the input fields are initially empty and need to be filled accordingly.

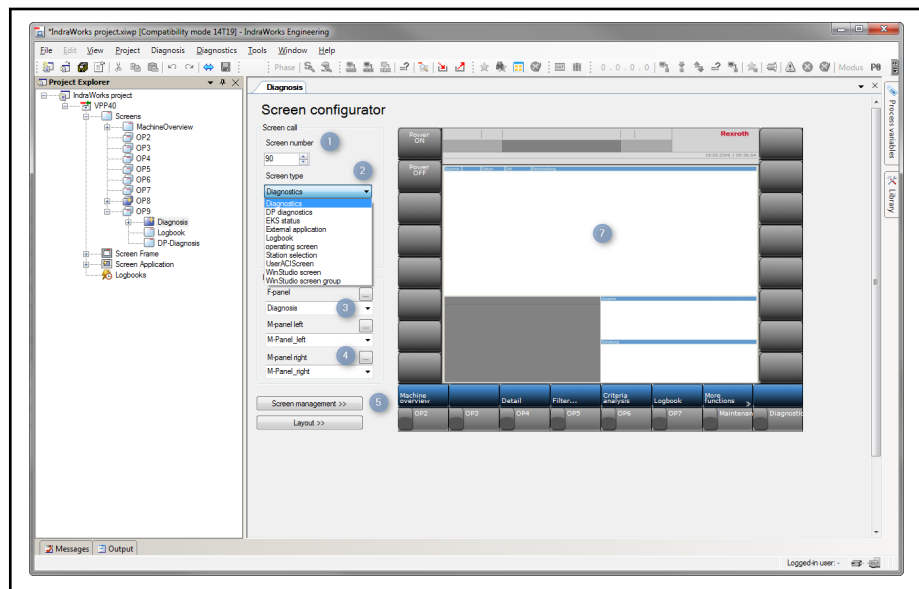


Fig. 5-21: Screen editor

Screen number ①

Unique screen number for interaction with a PLC program. The value is written to a configurable PLC variable when the screen is opened. The PLC can also trigger a screen switching if it writes the entered screen number to a configurable PLC variable (see also [chapter 5.2.4 "Screen control by the PLC" on page 74](#)). The write function of screen numbers is presently only active for user screens, the diagnostic overview and external applications.

When the editor is closed, the system checks on unique screen number. If the number is already used in a different screen, the screen is highlighted with a warning symbol in the "Screens" folder. If this is the case, change the screen number of one or more of the highlighted screens.

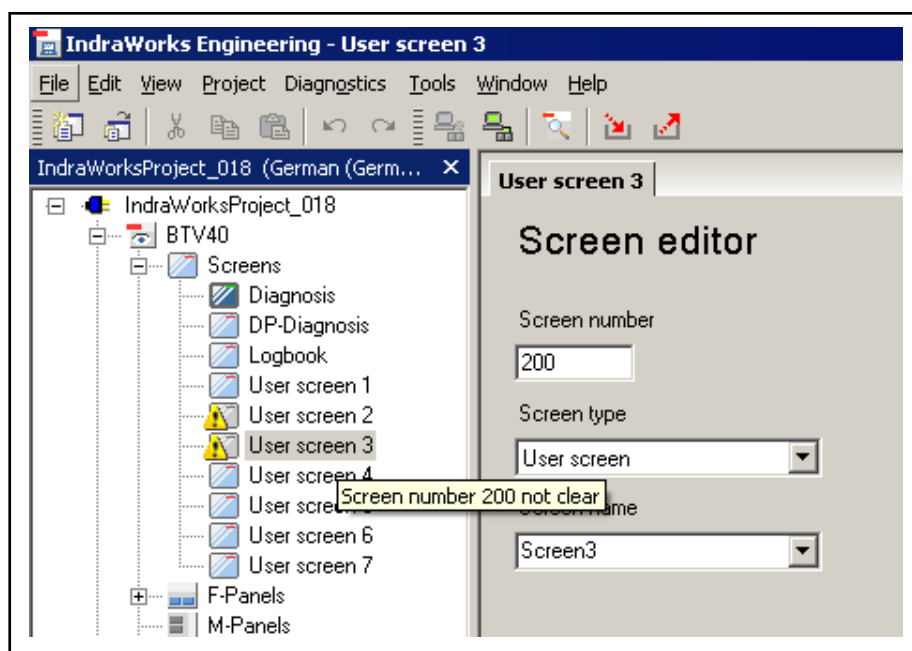


Fig. 5-22: Warning for non-unique screen numbers

Screen type ② The "Screen type" list displays all of the available applications which can be configured and called. A screen type has to be selected.

It is predefined for different screen types whether additional F-keys and M-keys can be configured. If this is not the case, the corresponding input fields are disabled. A screen type can also be provided with a parameter needed for the application start at runtime. If an application (e.g. user screen) requires such a parameter, an additional input field is displayed, where the parameter in question (e.g. screen name of a WinStudio screen) must be entered. The values to be entered for the parameter can be found in the application documentation of the respective system.

Screen parameter Some of the screen types require a parameter, which defines the screen content, to open the screen. This may, for example, be the name of a configuration file to be used. Screen types, which require a parameter, display a button used to open the "Configuration" dialog or the screen editor.

Selecting F-keys ③ F-keys can be assigned to the screen using the "F-panel" input field. These F-keys can be labeled and assigned arbitrarily with functions. These F-keys are configured by means of the F-panel editor (see [chapter 5.3 "Configuring F-keys" on page 78](#)). If no F-panel is entered, the F-key assignment of the screen previously shown remains at runtime.

The selection list provides all F-panels already created and listed in the "F-panels" folder. To create a new F-key assignment, enter the name of the new F-panel in the input field. When changing the input focus to a different field (with <TAB> or left-clicking), a dialog opens used to confirm the creation of the new F-panel. Then, the new F-panel appears in the "F-key panel" selection list and in the "F-panels" folder. Functions can be assigned to the F-panel in the F-panel editor. The panel can be used by different screens.

Selecting M-keys ④ M-keys are configured like F-keys. Existing M-panels can also be selected or new M-panels can be created for the right and left side. New entries are added to the "M-Panels" folder. Left and right M-panels can be hidden independently from each other. Proceed as described for F-keys under: "Hide F-Keys".

Screen management ⑤ Press the "Screen management" to show and hide the following options:

Configuring the user interface

- **Operating area:** Each screen can be assigned to an operating area combining several screens to form a logical unit. Press the assigned OP-key to consecutively open all screens related to an operating area at runtime. When switching to a different operating area, all screens of the operating area exited are closed. Then, the screens of the enabled operating area are opened so that only logically related screens are open.
- **Close screen automatically:** If this option is selected, the screen does not remain open in the background at runtime, but it is automatically closed with the following actions:
 1. "On screen change": If a new screen is opened or an already existing screen is added via toggling, the current screen is closed.
 2. "Upon operating area change": The screen remains open if a new screen is added. If the new screen belongs to another operating area, the current screen is closed. If the operating area is changed with the OP-key, this screen is also closed.

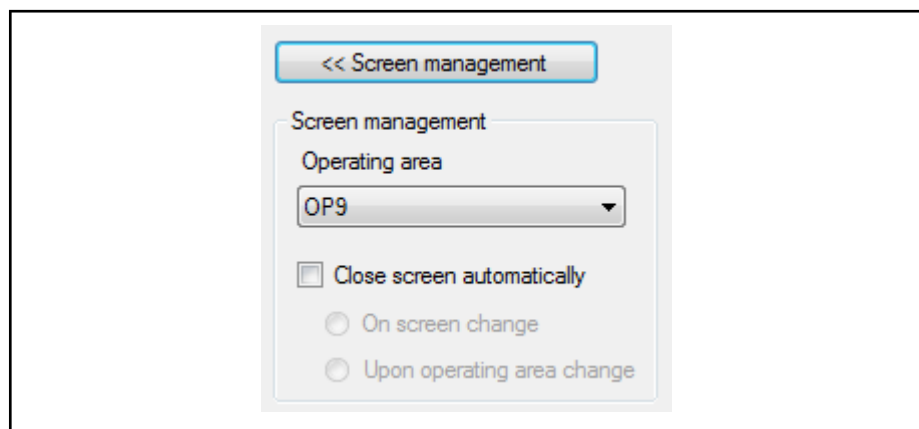


Fig. 5-23: Screen management

Layout ⑥

Press the "Layout" button to show or hide further options to configure the layout. Use these options to determine whether individual areas are to be hidden to enlarge the screen area for the application.

If areas are hidden, the changed layout can be checked immediately in the screen preview.

The configured layout is only applied at runtime if the respective screen is enabled. If another screen is subsequently opened, the configured layout is activated for the new screen.

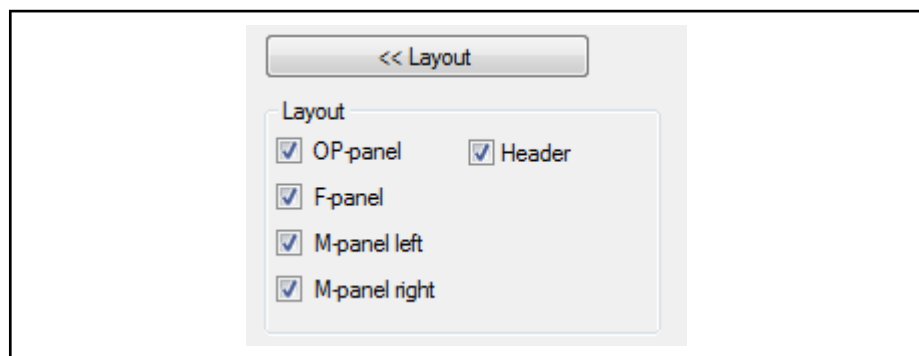


Fig. 5-24: Layout



If F- or M-panels are hidden, they are not available anymore for screen toggling at runtime. In this case, it is recommended to project the screen toggling to a remaining panel or to provide a screen toggling using a PLC variable.

- Screen preview ⑦** A preview of the respective screen type is displayed in the right window area of the screen editor to illustrate the screen content and the screen layout. This preview shows the configured layout as well as the assignment of the selected F-, M- and OP-panels added together with the screen at runtime.
- Saving screen** The settings of a screen are saved as soon as the respective editor is closed. Current changes are saved after a prompt. If the editor should remain open, the menu item **Save** in the context menu of the project node can be selected instead. This also saves the changes made in all other open editors.
- Renaming screen** The screen name can be changed later on. Select the screen entry from the "Screens" folder and click on it again. The entry changes to the input mode and can be edited.
- Deleting screen** After a screen that is not required can be deleted by pressing the key or using the context menu after it has been selected in the project folder. The entry is removed from the folder.

5.2.3 Configuring operating areas

An operating area provides a logical grouping of screens. For example, in an HMI standard application, all of the diagnostic functions (diagnostic overview, logbook etc.) are combined in the "OP 9" operating area.

When IndraWorks Operation is at runtime, the grouping of screens into operating areas means that when there is a change to another operating area, all of the open applications are firstly closed and then the application of the enabled operating area is opened.

- Adding operating area** To add a new operating area, open the context menu of the "Screens" folder and select **New Operating Area....** An input dialog appears to enter any operating area name. After confirming the entry with **OK**, a new folder is created below "Screens", in which existing screens can be added.

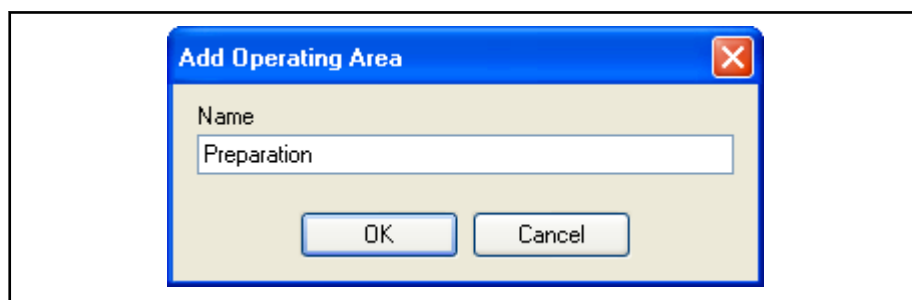


Fig. 5-25: Creating new operating

- Adding or removing screens** To assign a screen to an operating area, select a screen located in the "Screens" or operating area folder and, whilst holding down the mouse key, drag it to the required folder.

Alternatively, use the operating area editor. It is opened when double-clicking on the "Operating Area" folder.

Configuring the user interface

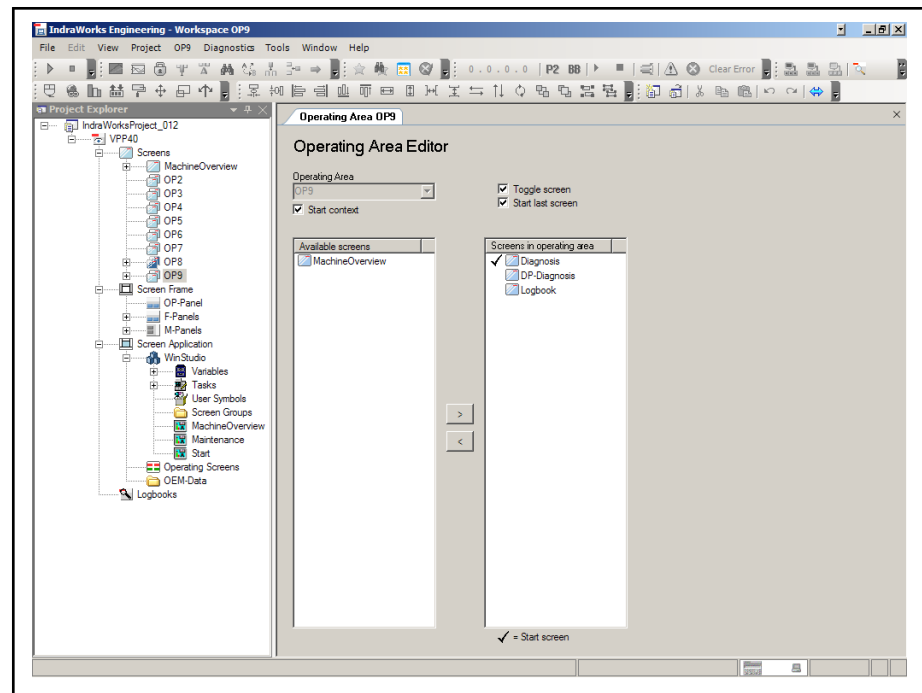


Fig. 5-26: Operating area editor

Deleting operating area

To delete an operating area, select the **Delete** function in the context menu of the folder. When an operating area is deleted, only the assignment of the screens is deleted. Its configuration data remains. The screens are not assigned to any operating area and displayed in the "Screens" folder.

Start screen of an operating area

Define a screen for every operating area. The screen is automatically opened when the operating area is enabled in IndraWorks Operation. The current start screen of the operating area is labeled by a respective icon. To change the start screen, select a screen located in the operating area folder, and select the **Set as start screen** function in its context menu.

Configuring the user interface

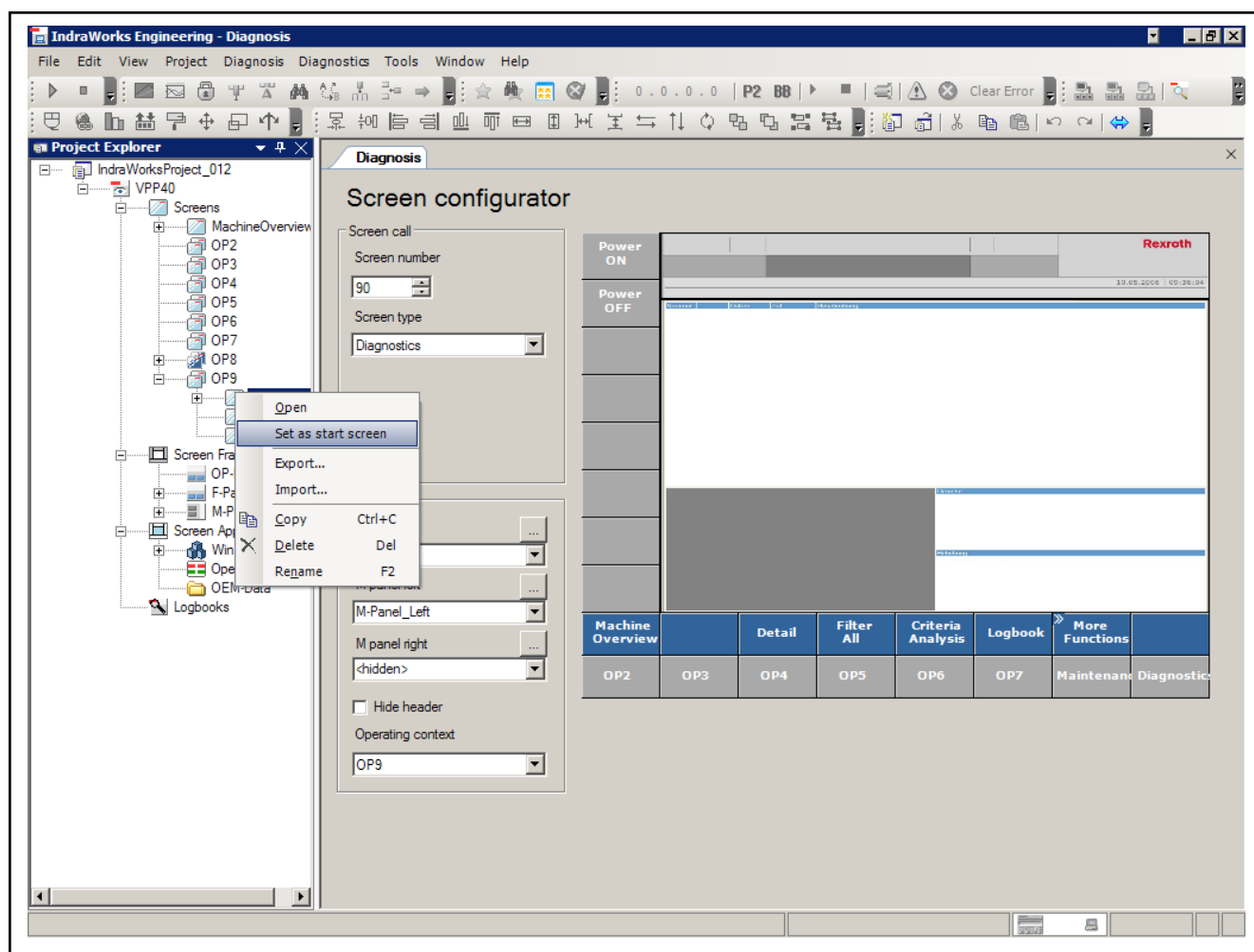


Fig. 5-27: Defining the start screen

Starting area As "start screen", a "starting area" can be defined. The starting area is the operating area enabled immediately after the start of IndraWorks Operation. The start screen of the enabled operating area is opened, that means that the start screen of the starting area is the application enabled at the start of IndraWorks Operation.

To select an operating area as starting area, perform the function **Starting area** in the context menu of the operating area folder. The current starting area is labeled by a changed icon.



The combination of "starting area" and "start screen" defines the screen enabled immediately after IndraWorks Operation has been started.

Advanced settings

As an alternative to the aforementioned editing functions, the operating area editor can also be opened by double-clicking on its folder. The operating area editor contains a list with all unassigned screens and a another list with all screens assigned to this operating area. Remove or add screens to and from this operating area using the < and > buttons.

The following settings are also available in the operating area editor:

- **Toggle screens**

Configuring the user interface

This option determines whether screens, which have already been opened, are opened after each other by repeated pressing of the assigned OP-key

- **Memorizing last screen**

This option determines whether the screen currently displayed in IndraWorks Operation will be enabled again after a change of work-space. Should the option be disabled, the selected start screen is displayed following each enabling of the operating area

Activating operating area

To enable an operating area in IndraWorks Operation, it has to be assigned to an OP-key. Open the OP-panel editor and select the required OP-key to assign an operating area to an OP-key. Then, in the group "Function", select the entry "Enable operating area" and the name of the desired operating area in the "Operating area" selection list.

5.2.4 Screen control by the PLC

Each visualization device allows to synchronize the user interface with the enabled PLC program. This can be achieved by writing a screen number entered for a screen to a configurable PLC variable as soon as the screen is displayed in the runtime environment. This mechanism can be used to trigger respective actions in the PLC program if necessary.

If a screen is called for which no screen number has been assigned, a value of "-1" is assigned to the PLC variable.



The PLC variable for the "Write screen number to PLC" function has to be declared as signed value type (e.g. INT). Otherwise, -1 is not transmitted.

It is also possible to switch a screen by setting a second PLC variable. The related screen is enabled with the operating area also being switched in certain circumstances and the specified F-panels and M-panels are displayed. This function is presently only active for user screens, the diagnostic overview and external applications.

The screen is configured via the context menu of the "Screens" folder and the menu item **Properties**.

Then, the following configuration dialog opens:

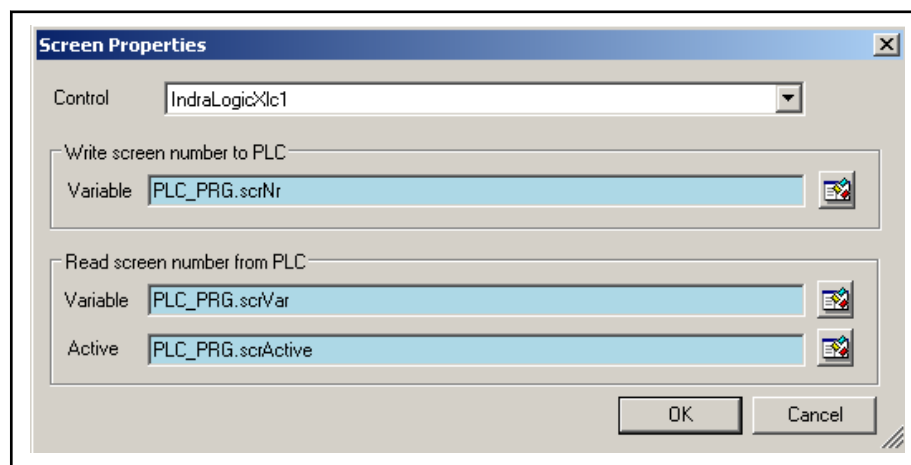


Fig. 5-28: Screen control configuration

Control

Name of the control containing the program selected for communication with the visualization device.

Configuring the user interface

Browsing PLC variables	Select PLC variables via the IndraWorks browser dialog. It is also possible to edit them manually, see chapter 6 "IndraWorks process variables" on page 199 .
Write screen number to PLC	PLC variable (including instance path) of type Integer to which the screen number is written.
Screen activation by PLC	Name of a PLC variable of type Integer from which the screen number is read from the PLC. A Boolean PLC variable read by the visualization device has to be entered in the "Active" field. Once this variable assumes the value TRUE, the screen switching (opening screen) is activated. After this variable has been evaluated, the visualization device is reset to the value FALSE.

5.2.5 External applications

IndraWorks Operation allows any Windows applications with a graphical user interface to be started using F-keys and M-keys and to be implemented into the screen management. To do this, create a screen in the "Screens" folder and select the "External Application" screen type. Next to the "Application" input field on the right, the screen editor provides a button which opens another dialog to configure the parameters of the application call.

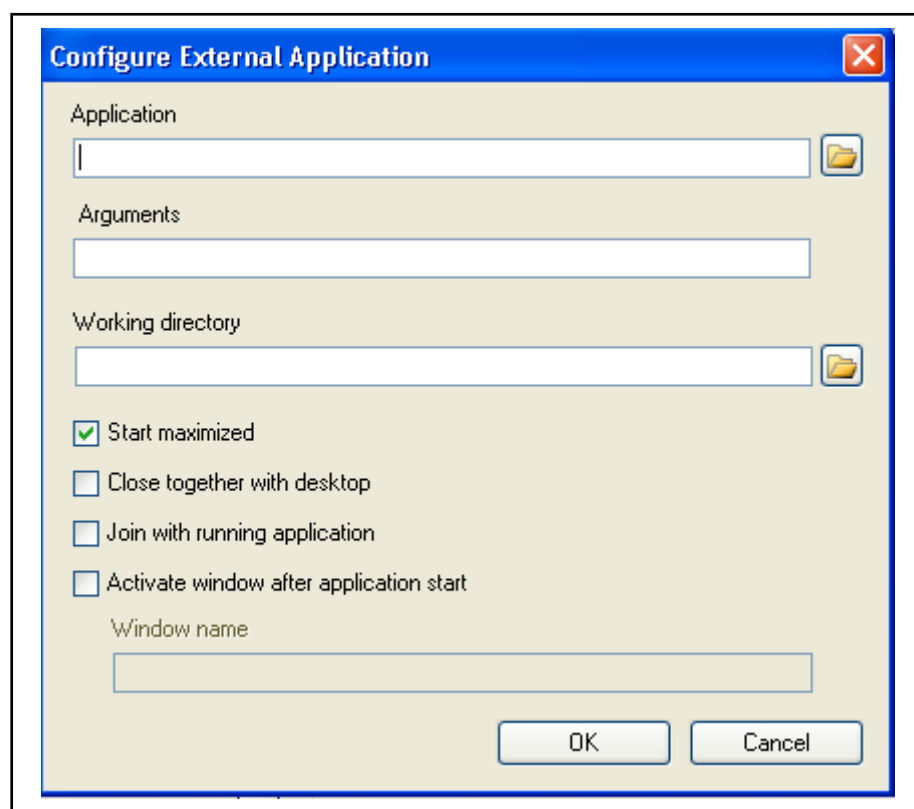


Fig. 5-29: Configuring the call of an external application

Application	Application name including path specification.
Arguments	Parameters transferred to the external application at start. It can be the name of a configuration file or a list of program parameters. The format of the argument list can be found in the documentation of the application to be started.
Working directory	To function properly, some applications require the specification of a working directory in which it is searched for additional program components.
Start maximized	Enable this option to start the application in full frame mode.
Close together with desktop	Enable this option to exit the application automatically when exiting the operating desktop.

Configuring the user interface

Join with running application

If this option is enabled, the existing instance is linked to IndraWorks Operation when the screen is opened and no new instance of the external application is started in IndraWorks Operation. This means that the application is incorporated in the operational design (e.g. when switching to a different operating area) of IndraWorks Operation. The file name of the external application acts as a linking criterion.

Activate window after application start

This option allows an already started application to be linked to IndraWorks Operation. The window name displayed in the title bar of the started has to be entered.

5.2.6 Advanced copy function

Use the "Duplicate" function of a screen to create duplicates for the referenced elements in addition to the screen duplicate. The settings for duplicate creation are defined in a separate dialog. Call the dialog via the context menu of the respective node in the Project Explorer. The existing copy function for screens remains unchanged and simply creates a copy of the screen using the referenced elements of the original.

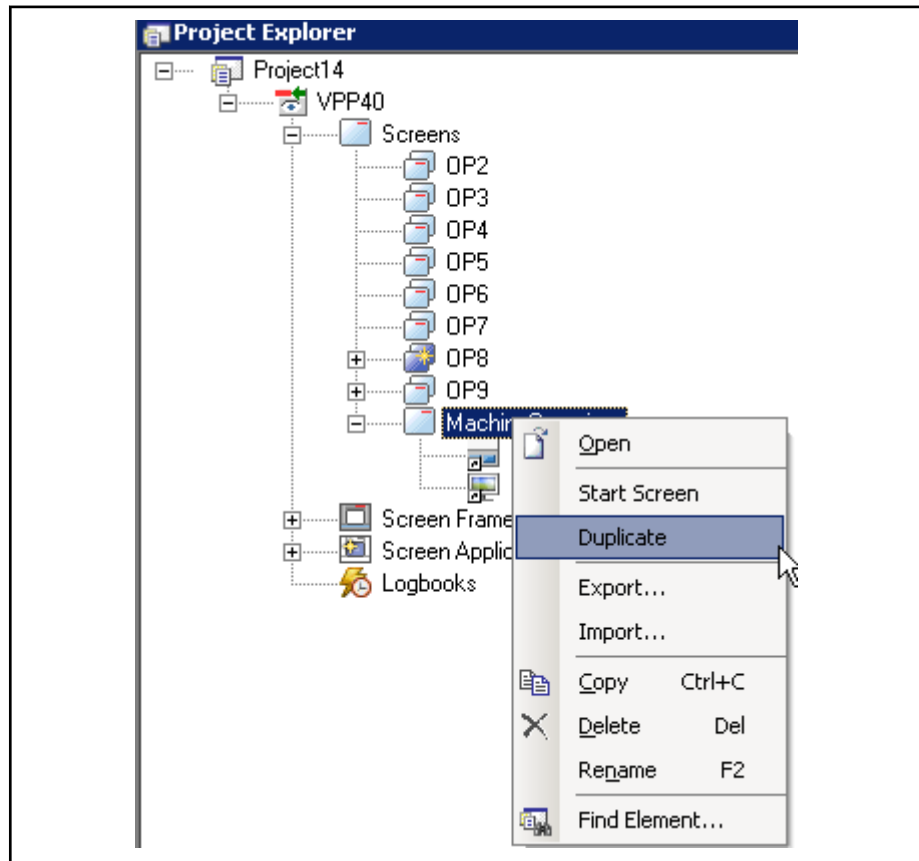


Fig. 5-30: Advanced copy function for screens

Select which referenced objects are to be copied in the "Duplicate screen" dialog. The option "Copy referenced objects" is enabled and all objects are selected in the presetting. In this case, a copy is made for all referenced objects. The names of the copies created may be edited prior to confirmation. By default, their original names with a suffixed number are entered. These names may be changed as desired. The only editing restrictions are validity (no empty string and no invalid characters) and uniqueness of the name of the new copy. If an invalid name is entered, an error provider is displayed and the **OK** button is disabled. If an element is not to be copied, deselect it. In this case, the original uneditable name of the referenced element is displayed

Configuring the user interface

and the original reference remains intact. The state of the option "Copy referenced objects" changes in this case. Unassigned references are also disabled and cannot be enabled.

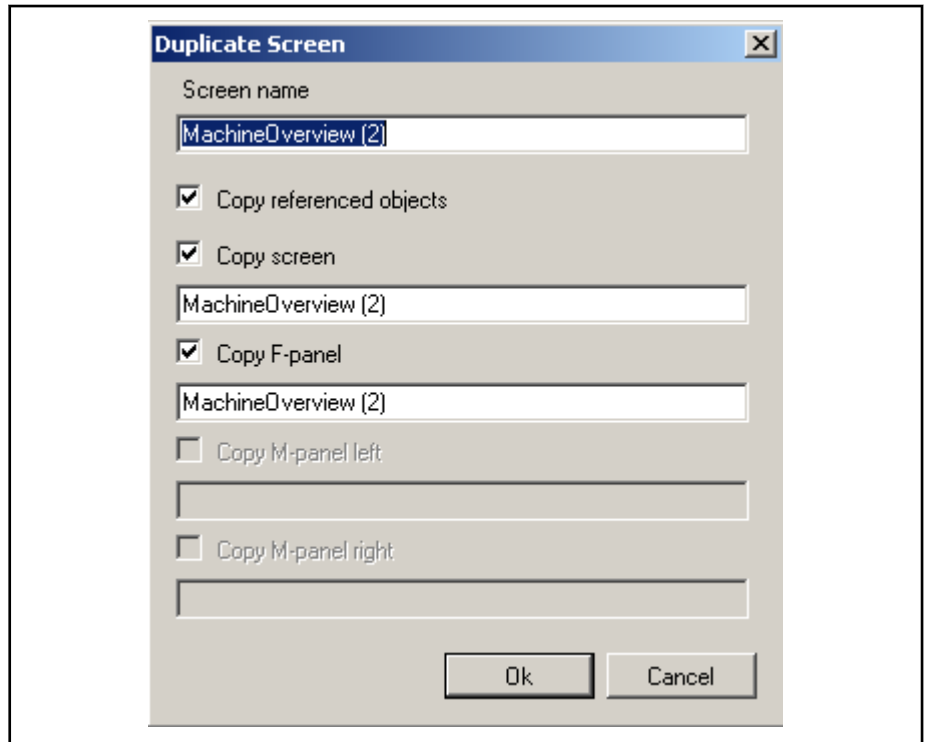


Fig. 5-31: Duplicate screen

Another special case are screen types that provide and/or require no specific editor (e.g. "External application"). Here, there is no referenced element, i.e. the copy function cannot be activated for these elements. This also applies to screens that were not assigned a screen type yet.

Configuring the user interface

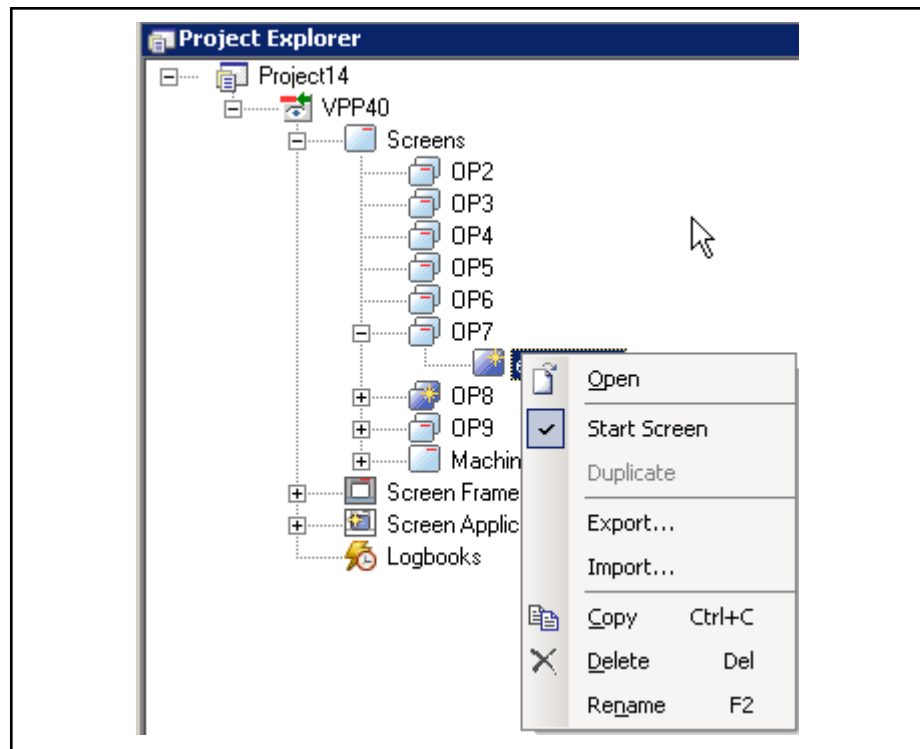


Fig. 5-32: Duplication is disabled for certain screen types

5.3 Configuring F-keys

5.3.1 General information

F-keys are used to start or enable applications, to trigger functions and to change the F-key assignment. Several F-keys are grouped to form an F-panel. The number of keys per panel depend on the device type.

5.3.2 Creating an F-panel

New F-panels are created either in the screen editor (cf. [chapter 5.2 "Configuring screens" on page 66](#)) or directly in the IndraWorks Project Explorer via the "F-panels" context menu. The entry **Add ► F-panel...** (up to 13V04 **New panel**) opens an input dialog. Enter any designation in the "Panel name" field. The new panel appears in the "F-Panels" folder with this designation. The new panel is saved to a new file whose name is displayed in the "File" field. The file name is automatically generated from the panel name entered. However, it is possible to change the default name and use any file name.

After confirming with **OK**, the new entry is added to the "F-Panels" folder. The new panel can now also be selected in the "F-keys" selection list of the screen editor.

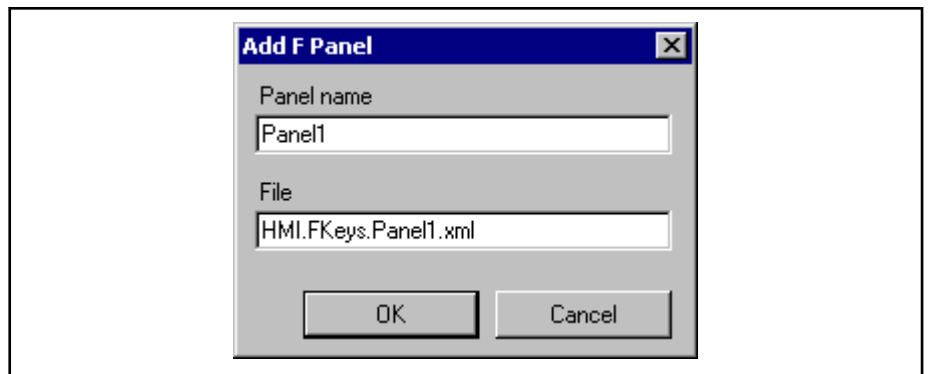


Fig. 5-33: Input dialog for a new F-panel

5.3.3 F-panel editor

Any F-panel can be edited using the F-panel editor. To open the F-panel editor, double-click on a subelement in the "F-Panels" folder. In its upper area, the F-panel editor displays a panel preview displaying the texts and images of the individual keys (see ① in the following figure). The settings of each key are displayed in the fields below the key preview (②). These fields display the settings of the preview key highlighted above.

The configuration of an F-key comprises the representation, i.e. labeling and display of an image as well as the function to be enabled when the key is pressed at runtime.

Click on **Details>>** to show advanced setting options for specific text and image alignment.

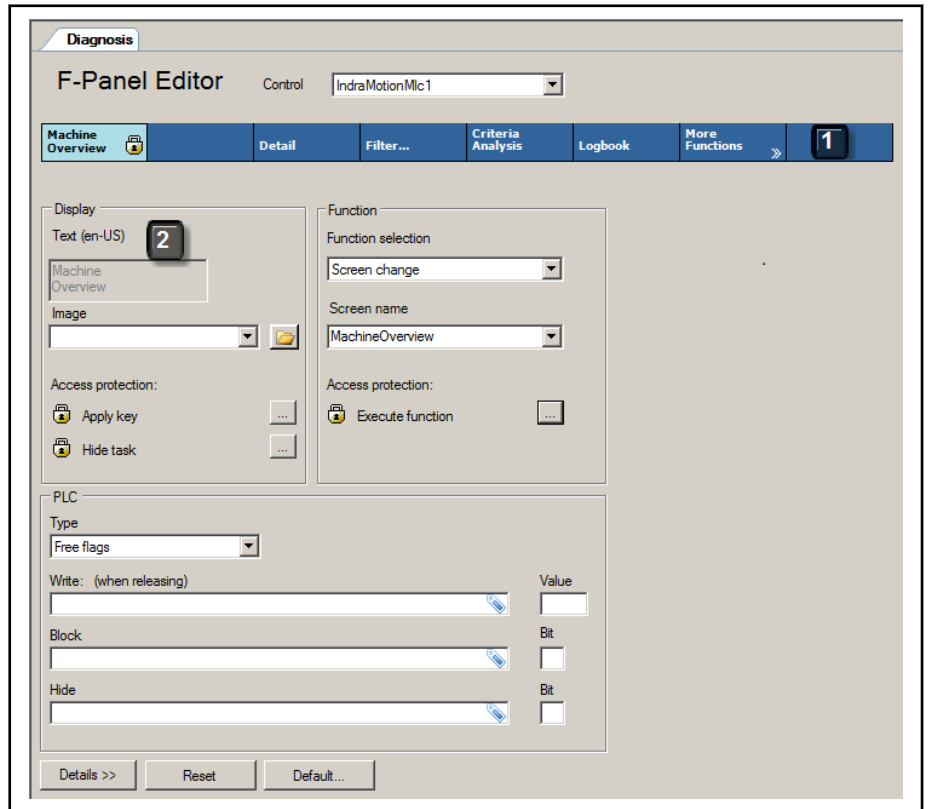


Fig. 5-34: F-panel editor

Configuring the user interface

Editing display Enter the text to be displayed on the F-key in the "Display" field. An image can also be selected.

Text The text may comprise two lines. Press <ENTER> to insert a line break.
The text is applicable to the respective language displayed. To enter the text in a different language, select the **Project ► Language ► Select project language** function.



If a text should be displayed identically in all project languages, select the project master language and enter the text. This text is used in all of the other project languages unless a different text is explicitly entered.

More details can be found in [chapter 5 "Configuring the user interface" on page 49](#).

Selecting an image Instead of or in addition to a text, it is also possible to assign an image to a key. Select an image file from the "Image" selection list. This image is immediately shown on the preview key. The selection list contains all of the image files already used in the current project. When a new project is created, this list is empty. New images can be imported to the project by clicking on the ... browser button. After the image file has been selected, it is applied to the selection list. As soon as the changes to the screen configuration are saved, the image file selected is copied to the project directory.

The following are supported to display the images:

- BMP
- ICO
- JPG
- GIF
- TIF
- PNG

It should be ensured that the size of the image selected does not exceed the size of the F-key.

The keys of the various device types differ in size, with the following default values being applicable:

- IndraControl VPP/VSP/VPD 16: 97 x 29 pixels
- IndraControl VPP/VSP/VPD 40: 126 x 40 pixels
- IndraControl VPP 21: 126 x 40 pixels

If the maximum size is used, the image is only fully visible if the image is centered in the detailed settings or if the offset values are set to '0'. It should also be ensured that the color depth of the image does not exceed the screen setting, otherwise, the image could be portrayed incorrectly.

Function Each F-key can be assigned to a function. To achieve this, the "Function" list box provides a list of predefined basic functions. Default functions are "Screen change" and "Panel change". Other functions can be provided depending on the system installed.

In the "Screen name" field, the "Screen change" function provides all screens from the "Screens" folder that have already been configured. These are user-defined screen IDs used to file specific screen information (screen type, F-panels, M-panels, screen number). If this function is executed at runtime, the screen is selected and the respective panels are displayed. Double-click on

Configuring the user interface

the appropriate screen entry in the "Screens" folder to call the filed screen information.

The "Panel change" function only causes a change in the F-key assignment. This function can be used to select a different F-panel that has been created beforehand.

Access protection

The lock icon on the F-key is displayed if one of the following three access protection function is active in the user management:

- **Execute key**
Icon is grayed out if all groups can execute the key. The icon is active as soon as a group cannot execute the key.
- **Hide key**
Icon is grayed out if all groups can see the key. The icon is active as soon as a group should not see the key.
- **Execute the function "Screen change"**
Icon is grayed out if all groups can execute the "Screen change" to the selected "screen name". The icon is active as soon as a group cannot execute the screen change.

By clicking on , the access right to each user-defined user group can be configured.

Browsing PLC variables

Drag and drop the browser elements to the respective input fields of the editor. It is also possible to edit them manually, see [chapter 6 "IndraWorks process variables" on page 199](#).

Writing a variable

Use this field to enter a PLC variable or to drag and drop it from the IndraWorks browser. A fixed value is written to the variable if the F-key is actuated at runtime. Enter the value to be written in the "Value" input field. Please ensure that a valid control is selected in the "Control" selection list.

Disabling

If a PLC variable is entered in this input field, the related F-key is disabled or enabled at runtime depending on the variable value. The key is only operable if the value is TRUE. Otherwise, key labeling is grayed out and the function cannot be executed. The "Control" selection list establishes the relation to the control. This can be set separately for F-panels.

Hiding

If a PLC variable is entered in this input field, the related F-key is hidden at runtime depending on the variable value.



In addition to entering BOOL type variables, unsigned integer data types (USINT, UINT, UDINT, ULINT) and bit sequence data types (BYTE, WORD, DWORD, LWORD) may also be used to "disable" and "hide" a key. If one of these new types is selected, the zero based bit address can be optionally added in the "Bit" input field. (the standard value is empty and/or not applicable).

Details

Click on **Details>>** to show advanced setting options for specific text and image alignment.

Resetting

The **Reset** button can be used to reset the settings of the open panels to align the texts and images to the default values.

Default...

Click on the **Default...** button to open a dialog where the current default values can be set.

Copying, deleting, pasting an F-Key

The configuration of an individual F-key can be copied and pasted to a different F-key. Right-click on the key to be copied and select "Copy". Select a dif-

Configuring the user interface

ferent key (also in a different panel). Assign the configuration by selecting the **Paste** function in the context menu. All settings, i.e. label and function of the F-key are overwritten with those of the copied key. Select the **Delete** menu item to remove all settings.

All actions can be called with the shortcuts:

- <Ctrl>+<C> (copy)
- <Ctrl>+<V> (paste)
- <Ctrl>+<X> (cut)
- (delete)

5.3.4 Renaming an F-panel

The name of an F-panel can be changed subsequently. Select the F-panel in the Project Explorer and click on it again. The entry changes to the input mode and can be edited. The file name specified when creating the panel remains.



If the panel has already been assigned to a screen, this setting has also be adjusted at the respective screens.

5.3.5 Deleting an F-panel

After selection in the Project Explorer, any F-panel that is not required can be deleted by pressing or via the context menu. The entry is removed from the folder and the related file is deleted. Any screens using the F-panel deleted has to be adjusted accordingly.

5.3.6 Structuring F-panels in folders (from 13V06)

F-panels can be structured with folders. Use "Add", "Import", "Paste" or "Move" to assign the F-panels to folders.

Create a new folder with the context menu function **Add ► Folder**. The number of folders and their nesting depth is not limited.

The context menu functions "Add", "Export"/"Import", "Copy"/"Paste", "Delete" and "Rename" of the folders is analogous to the F-panels.

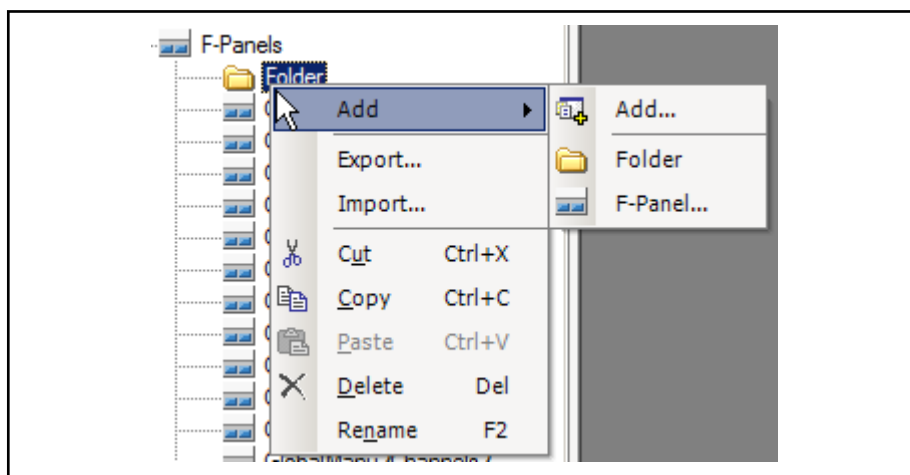


Fig. 5-35: Folder (context menu)

5.4 Configuring OP-keys

5.4.1 General information

The OP-keys at the bottom of the screen can be used to start applications (screen change) and to change what is called operating areas. In the IndraWorks HMI basic setting, the keys <OP8> and <OP9> are assigned with "Maintenance" and "Diagnostics". The function of these and the other OP-keys can be configured using the OP-panel editor.

A special function of the OP-keys is the enabling of operating areas. Exactly one operating area, which comprises several applications (e.g. screens) that are logically related to each other, can be assigned to an OP-key. When an operating area is enabled for the first time, the set start screen is opened. Start further applications with F-keys or M-keys. To display already open applications one after the other, click on the OP-keys repeatedly.

The OP key configuration can be found in the **ScreenFrame ► OP-panels** folder. This folder contains an OP-panel after adding a visualization device in the default configuration. The OP-panel is assigned with different labels and functions depending on the application type. The standard OP-panel is mandatory to operate IndraWorks Operation and can thus neither be deleted nor renamed. The preassignment, however, can be edited with the OP-panel editor.



The "OP-panels" folder is only available for visualization devices created with the current version. When wanting to update an existing project, open the HMI property dialog, change the device version to 14V08 or higher. Subsequently, the "OP-panels" folder is added. Using this folder, more OP-panels can be created.

Configuring the user interface

5.4.2 OP-panel editor

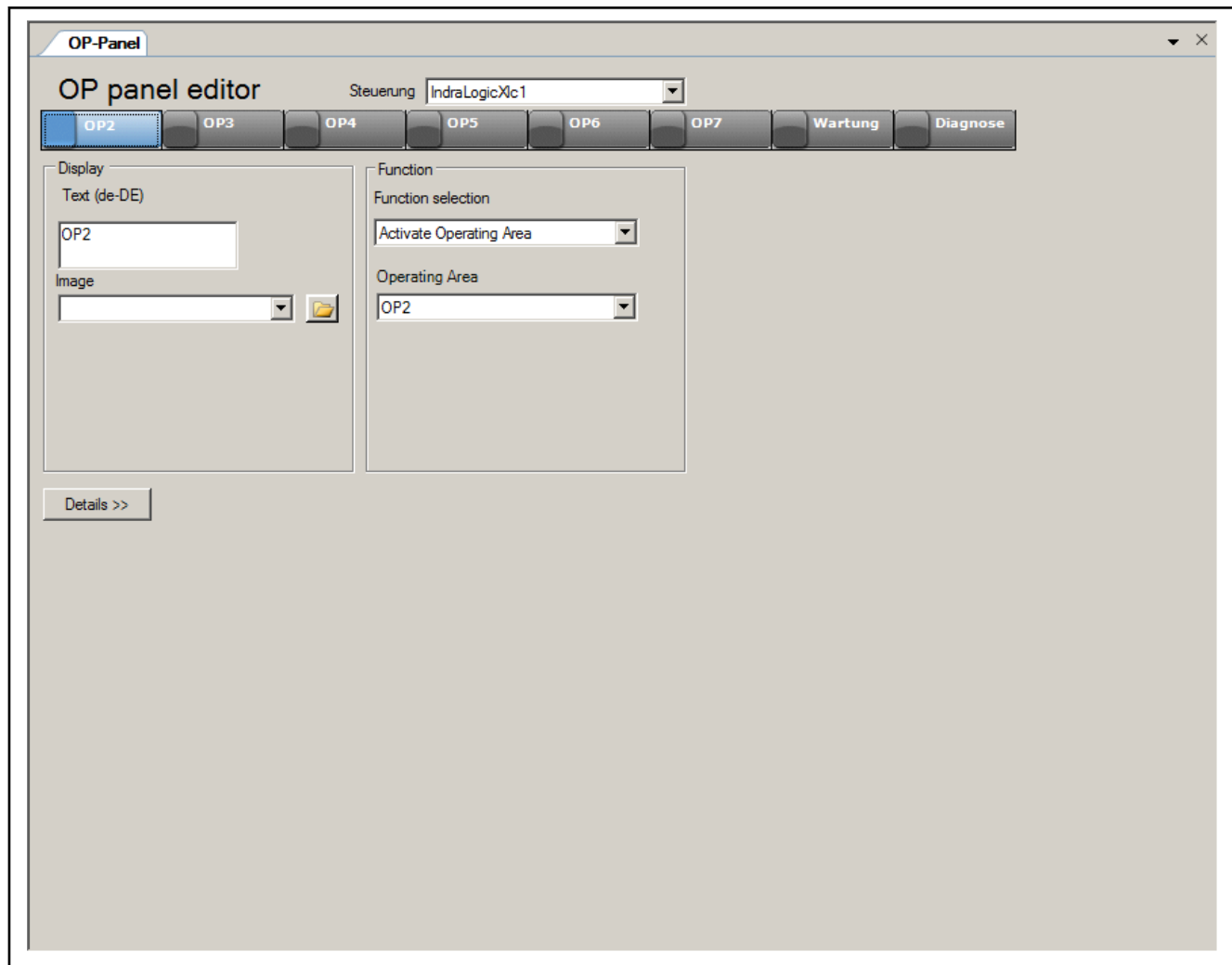


Fig. 5-36: OP-panel editor

- | | |
|----------------------------|--|
| Editing the display | Enter the text to be displayed on the OP-key in the "Display" field. The text may comprise two lines. Press <ENTER> to insert a line break. |
| Text | <p>The text may comprise two lines. Press <ENTER> to insert a line break.</p> <p>The text is applicable to the respective language displayed. To enter the text in a different language, select the Project ► Language ► Select project language function.</p> <p>For further detailed information, please refer to chapter 5 "Configuring the user interface" on page 49.</p> |
| Selecting an image | Instead of or in addition to a text, it is also possible to assign an image to a key. Select an image file from the "Image" selection list. This image is immediately shown on the preview key. The selection list contains all of the image files already used in the current project. When a new project is created, this list is empty. New images can be imported to the project by clicking on the ... browser button. After the image file has been selected, it is applied to the selection list. As soon as the changes to the screen configuration are saved, the image file selected is copied to the project directory. |
| Function | A function can be assigned to an OP-key in the same manner as to an F-key (cf. chapter 5.3 "Configuring F-keys" on page 78) although the available |

Configuring the user interface

functions differ. The "Screen change" and "Activate operating area" functions are available for OP-keys. To configure an existing operating area or create a new one, click on ... which opens the operating area editor (see [chapter 5.2 "Configuring screens" on page 66](#)).

Details Click on **Details>>** to show advanced setting options for specific text and image alignment.

For more settings to specify the position of texts and graphics, refer to "Details". Furthermore, the font color can be changed under "Layout".

Copying, pasting, deleting OP-keys

The configuration of an individual OP-key can be copied and pasted to a different OP-key. Right-click on the key to be copied and select "Copy". Select a different key (also in a different panel). Assign the configuration by selecting the **Paste** function in the context menu. All settings, i.e. label and function of the OP-key, are overwritten by the copied key. Select the **Delete** menu item to remove all settings.

All actions can be called with the shortcuts:

- <Ctrl>+<C> (copy)
- <Ctrl>+<V> (paste)
- <Ctrl>+<X> (cut)
- (delete)

The OP-panel editor is extended from 14V08.

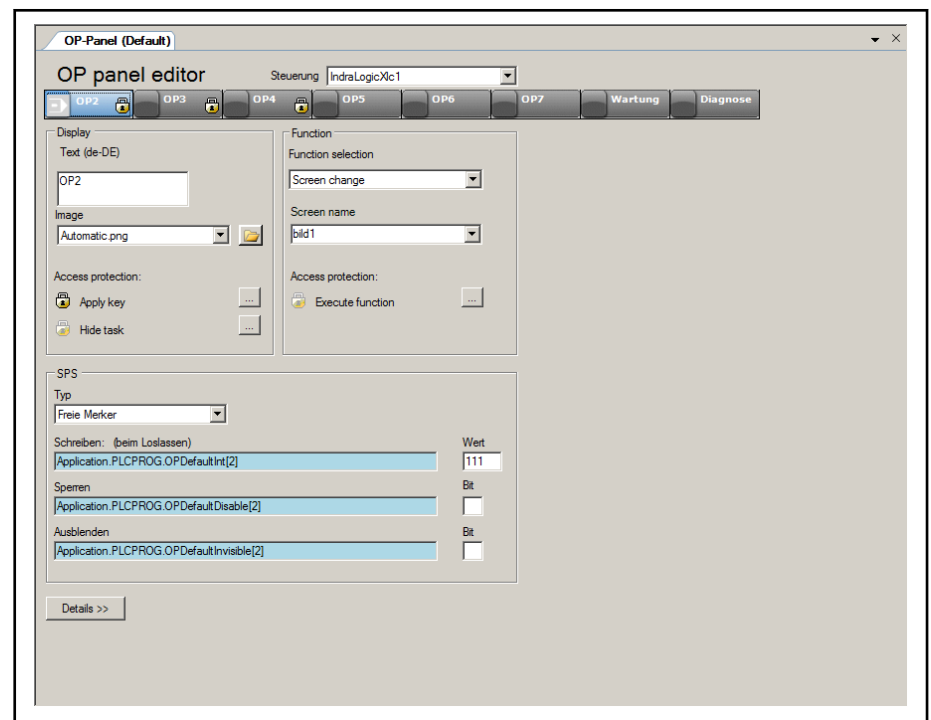


Fig. 5-37: OP-panel editor from 14V08

Access protection

The lock icon on the F-key is displayed if one of the following three access protection function is active in the user management:

- Execute key
Icon is grayed out if all groups can execute the key. The icon is active as soon as a group cannot execute the key.
- Hide key

Configuring the user interface

Icon is grayed out if all groups can see the key. The icon is active as soon as a group should not see the key.

- Execute the function "Screen change"

Icon is grayed out if all groups can execute the "Screen change" to the selected "screen name". The icon is active as soon as a group cannot execute the screen change.

By clicking on , the access rights can be configured in each user-defined user group.

Browsing PLC variables

Drag and drop the browser elements to the respective input fields of the editor. It is also possible to edit them manually, see [chapter 6 "IndraWorks process variables" on page 199](#).

Writing a variable Use this field to enter a PLC variable or to drag and drop it from the IndraWorks browser. A fixed value is written to the variable if the OP-key is actuated at runtime. Enter the value to be written in the "Value" input field. Please ensure that a valid control is selected in the "Control" selection list.

Disabling If a PLC variable is entered in this input field, the related OP-key is disabled or enabled at runtime depending on the variable value. The key is only operable if the value is TRUE. Otherwise, key labeling is grayed out and the function cannot be executed. The "Control" selection list establishes the relation to the control. This can be set separately for OP-panels.

Hiding If a PLC variable is entered in this input field, the related OP-key is hidden at runtime depending on the variable value.



In addition to entering BOOL type variables, unsigned integer data types (USINT, UINT, UDINT, ULINT) and bit sequence data types (BYTE, WORD, DWORD, LWORD) may also be used to "disable" and "hide" a key. If one of these new types is selected, the zero based bit address can be optionally added in the "Bit" input field (default value is empty or no specification).

5.4.3 Additional OP-panels

If the standard OP-panel is insufficient to operate the visualization device, more OP-panels can be added. Open the context menu of the "OP-panel" folder and select **Add... ► OP-panel**. Another element is added to the folder that can be edited analogously to the standard OP-panel.

It is automatically switched from the standard OP-panel to another panel as soon as a screen is enabled assigned to an operating area of this OP-panel. Explicit OP-panel switching is not possible. Thus, during the configuration, take into consideration that screens assigned to the standard OP-panel have to be called to be able to go back to the OP-panel.

5.5 Configuring M-keys

5.5.1 General information

M-keys can be used to enable functions on the user interface or to set PLC flags in a control. M-keys (panels) can be shown on the left or right of the screen. M-keys can be configured and labeled using the M-panel editor. Moreover, the function assignment and the link to the PLC flag can be configured by selecting the control and the instance of the "MKey" function block and entering the individual PLC flags.

Function block instance To link the M-keys to the PLC, an instance of the function block MKEYS or MKEYS_SINGLESEL has to be created in the PLC program (see [chapter 5.5.7 "Programming M-keys in the PLC" on page 91](#) or [chapter 5.5.8 "Programming M-keys in the PLC for VEx devices" on page 93](#)). Enter the name of this instance as a property of the M-panel. To do this, call up the appropriate dialog by selecting the **Properties** menu item from the "M-Panels" folder (by right-clicking).

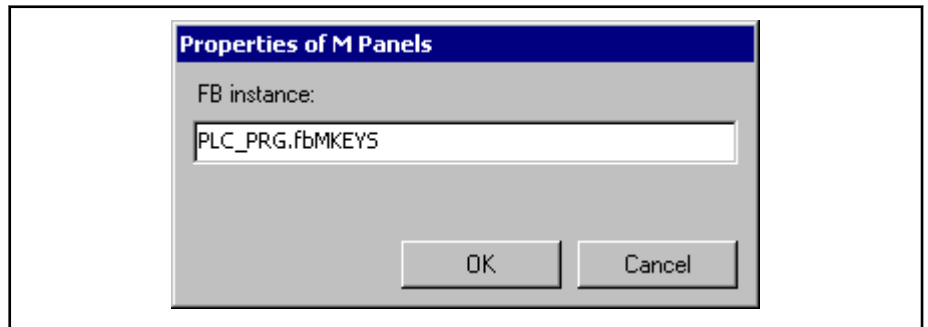


Fig. 5-38: Properties with instance name of the function block



M-keys can be parameterized so that they are active in different controls. The same instance name is used for every control.

5.5.2 Creating M-panels

Create new M-panels with the menu item **Add ► F-panel...** (up to 13V04 **New panel**) in the context menu of the folder "M-panels" (right mouse click). A dialog opens to enter a panel name which can be used as a reference when assigning this panel to a screen. A file name is generated automatically from this name. The file name is used to save the data of the M-panel for the communication with the PLC. The visual data of an M-panel, i.e. texts and images, is saved separately. Information the filing of this data is displayed in the "Reference panel" field. If necessary, change the entries by clicking on **Edit**. This is the case, for example, if a new M-panel should be inserted into an existing file and not into a new file. Select a file from the list instead of the file name specified.

Use the **Properties** function in the context menu to subsequently change the settings for an existing M-panel.

Configuring the user interface

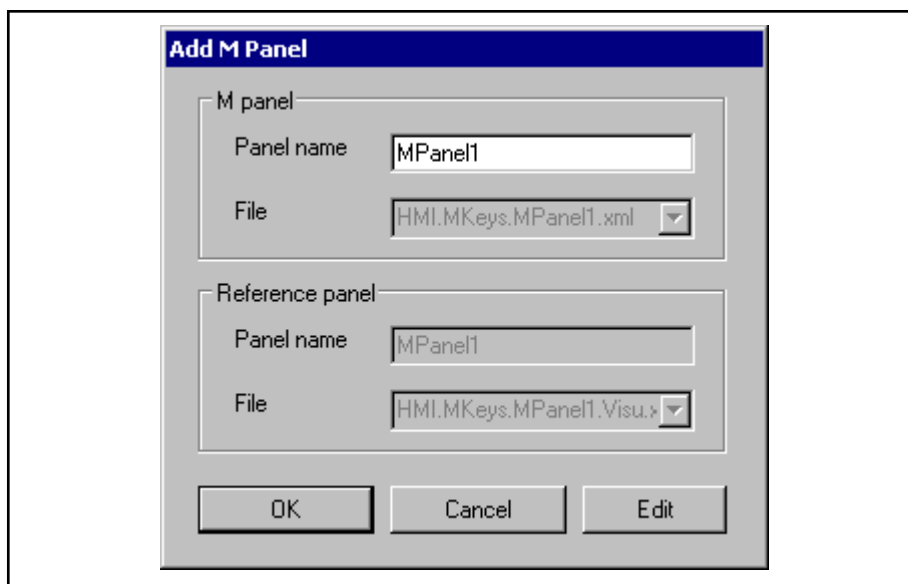


Fig. 5-39: Dialog to create an M-panel

5.5.3 M-panel editor

An M-panel can be edited using the M-panel editor. It opens via double-click on an entry in the "M-Panels" folder. On its left side, the M-panel editor provides a preview bar which displays the assignment of the current panel (see ② in the following figure). To show the settings of the selected key, click on the preview key. These settings are grouped in the "Display" (③), "Function" (④) and "PLC flag" (⑤) fields. In addition, it is also possible to set the position of texts and images (**Details** button).

Configuring the user interface

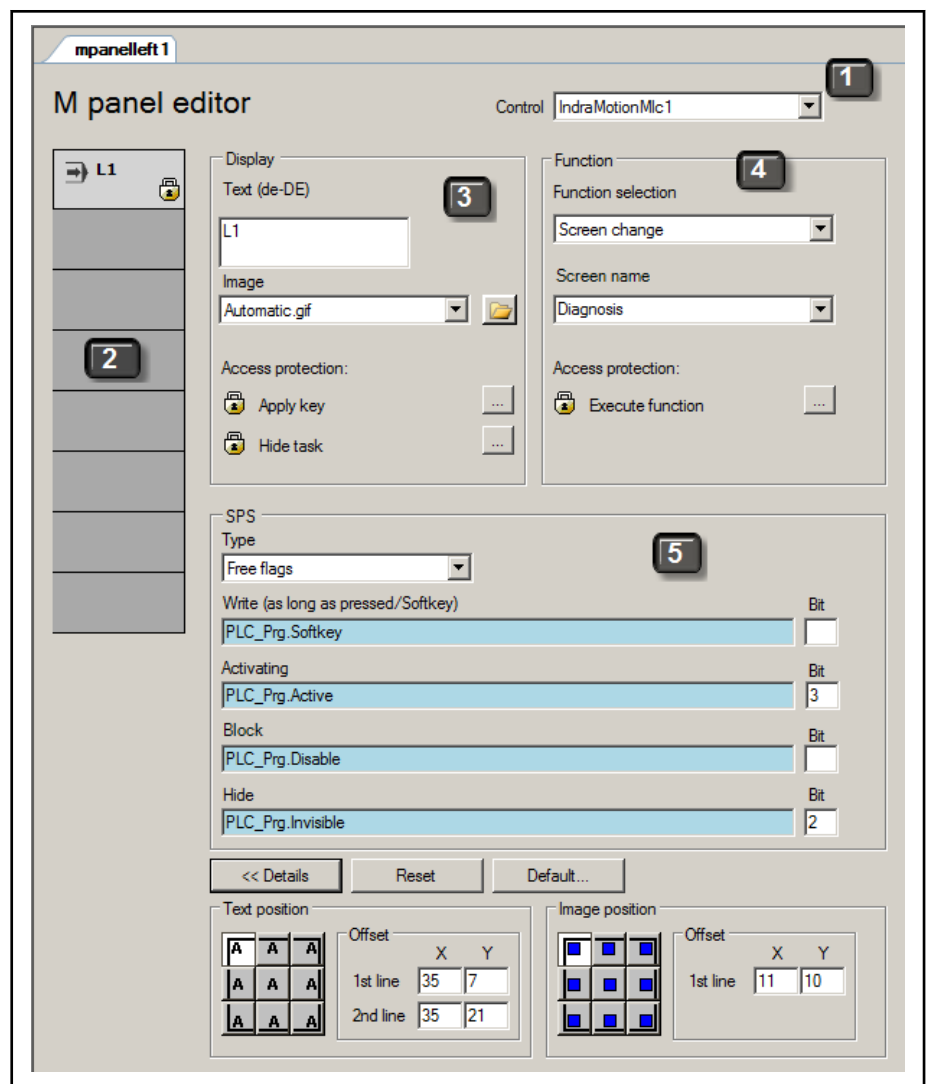


Fig. 5-40: M-panel editor

- Control ①** Enter the name of the control where the PLC program for M-keys is running in this field. If one or more configured controls already exist, then these can be selected in the input field.
- Preview bar ②** The preview bar shows the assignment of the current M-panel.
- Display ③** These settings serve to label the M-key selected with a text or an image. M-keys are configured in the same way as F-keys. For more detailed information on the input fields, please refer to [chapter 5.3 "Configuring F-keys" on page 78](#).
- Function ④** As is the case with F-keys, M-keys can also be used to call up screens and functions and to change the M-key assignment. M-keys are also configured in the same way as F-keys.
- Access protection** The lock icon on the M-key is displayed if one of the following three access protection function is active in the user management:
- Execute key
Icon is grayed out if all groups can execute the key. The icon is active as soon as a group cannot execute the key.
 - Hide key

Configuring the user interface

Icon is grayed out if all groups can see the key. The icon is active as soon as a group should not see the key.

- Execute the function "Screen change"

Icon is grayed out if all groups can execute the "Screen change" to the selected "screen name". The icon is active as soon as a group cannot execute the screen change.

By clicking on , the access right to each user-defined user group can be configured.

Browsing PLC variables

Drag and drop the browser elements to the respective input fields of the editor. It is also possible to edit them manually, see [chapter 6 "IndraWorks process variables" on page 199](#).

PLC flag ⑤

The "PLC flag" field serves to link an M-key to the appropriate PLC flags which have the following functions:

- **Write M-key variable (when pressed/softkey)**

Flag which is set to TRUE when the M-key is pressed and to FALSE when the M-key is released

- **Activating**

This flag is set by the PLC program to acknowledge that the key is pressed

- **Block**

To disable the triggering of an M-key function, enter the "Disabled" flag and set it to TRUE in the PLC program

- **Hide**

To hide an M-key, enter the "Invisible" flag and set it to TRUE in the PLC program. The M-key function is then locked.

Each flag has to be entered with its complete instance path (e.g. <Program-Name>.<FlagName>).



Only symbolic flags may be used. Inputs and outputs such as %IX1.0 may not be used. In addition to entering BOOL type variables, unsigned integer data types (USINT, UINT, UDINT, ULINT) and bit sequence data types (BYTE, WORD, DWORD, LWORD) may also be used to "disable" and "hide" a key. If one of these new types is selected, the zero based bit address can be optionally added in the "Bit" input field (default value is empty or no specification).

Details

If an alignment of texts and images that differs from the presetting is to be selected, additional configuration settings can be shown or hidden by clicking on the **Details** button. For more detailed information, please refer to [chapter 5.5 "Configuring M-keys" on page 86](#).

Copying, deleting, pasting an M-key

The configuration of an individual M-key can be copied and pasted to a different M-key. For more detailed information, please also refer to [chapter 5.5 "Configuring M-keys" on page 86](#).

5.5.4 Renaming an M-panel

The name of an M-panel can be changed subsequently. Select the panel and click on it again. The entry changes to the input mode and can be edited. The file name specified when creating the panel remains. The "Properties" dialog can also be used which can be selected from the context-sensitive menu (to be opened by right-clicking). The M-panel name and the name of the file in which the M-panel data is saved can be changed there.

5.5.5 Deleting an M-panel

After selection in the Project Explorer, any M-panel that is not required can be deleted by pressing the key or via the context menu. The entry is removed from the folder and the related file is deleted. Any screens using the deleted M-panel have to be adjusted accordingly.

5.5.6 Structuring M-panels in folders (from 13V06)

M-panels can be structured with folders. Use "Add", "Import", "Paste" or "Move" to assign the M-panels to folders.

Create a new folder with the context menu function **Add ► Folder**. The number of folders and their nesting depth is not limited.

The mode of operation of the context menu functions "Add", "Export"/"Import", "Copy"/"Paste", "Delete" and "Rename" of the folders is analogous to the M-panels.

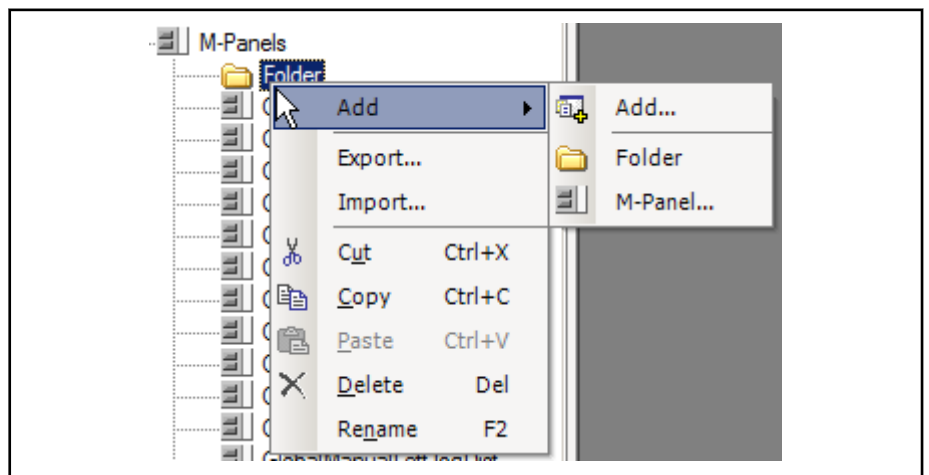


Fig. 5-41: Folder (context menu)

5.5.7 Programming M-keys in the PLC

Library and function blocks

By installing the respective IndraLogic target (target system), the HMI_MKeys.lib library becomes available. This library contains two function blocks (FB) to implement the M-key functionality.

Its use depends on the type of visualization device:

- [chapter "MKEYS function block" on page 91](#): Device with foil keyboard
- [chapter "MKEYS_SINGLESEL function block" on page 92](#): IndraControl VPP 21
- [chapter "MKEYS_VPP21 function block" on page 93](#): IndraControl VPP 21 without hardware key assignment

One instance is needed per visualization device in the PLC program.

Adding a library

The HMI_MKeys.lib library has to be added to the PLC user program. This is done using the Library Manager (see IndraLogic documentation and/or Help).

MKEYS function block

The MKEYS function block supports control and visualization devices with foil keys, e.g. VPP 16 or VPP 40. The inputs R1-R8 and L1-L8 have to be assigned with the key inputs. The "Enable" input can be used to enable (= TRUE) or disable (= FALSE) the function block.

Configuring the user interface

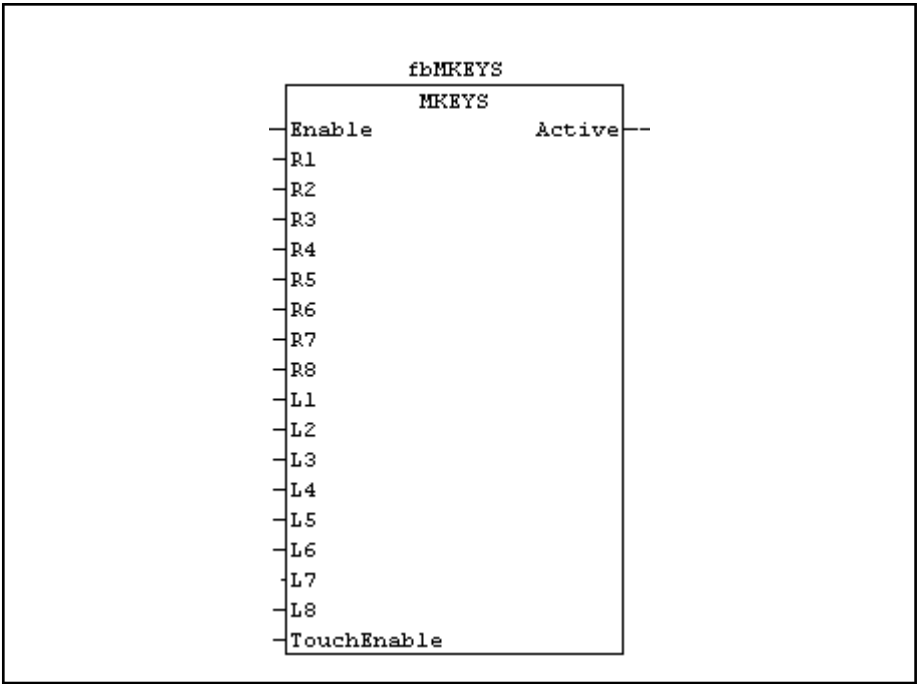


Fig. 5-42: MKEYS function block

Input/output	Description
Enable	Enables (= TRUE) or disables (= FALSE) the block
R1-R8	Assigned with the inputs of the hardware keys if any
L1-L8	Assigned with the inputs of the hardware keys if any
TouchEnable	To enable touch control, the hardware enabling key has to be assigned to this input
Active	It is TRUE if an M-key assignment is enabled

Tab. 5-1: Inputs and outputs of the MKEYS function block

MKEYS_SINGLESEL function block

The function block MKEYS_SINGLESEL supports the IndraControl VPP 21 visualization device.

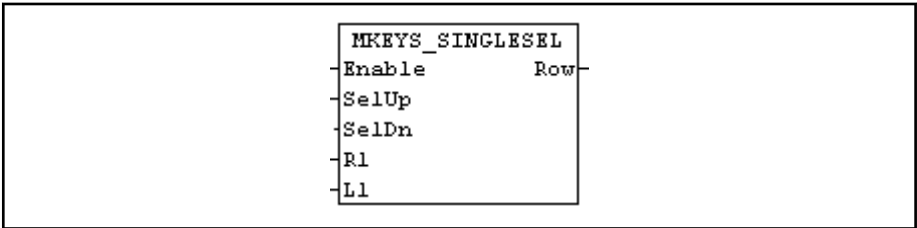


Fig. 5-43: MKEYS_SINGLESEL function block

Input/output	Description
Enable	Enables (= TRUE) or disables (= FALSE) the block
SelUp	Assigned with the inputs of the upper hardware keys if any
SelDn	Assigned with the inputs of the lower hardware keys if any
R1	It is assigned with the input of the right central hardware key

Configuring the user interface

L1	Assigned with the input of the middle hardware key on the left-hand side
Row	Provides the number (1-8) of the selected "key line"

Tab. 5-2: Inputs and outputs of the MKEYS_SINGLESEL function block

MKEYS_VPP21 function block

If the IndraControl VPP 21 visualization device does not provide the hardware keys as PLC inputs, the MKEYS_VPP21 function block can be connected in series with the MKEYS_SINGLESEL function block.

The instance name has to be the same as the instance name of the MKEYS_SINGLESEL function block with the extension "In". If the name of the instance of the MKEYS_SINGLESEL function block is, for example, **MK01**, the name of the instance of the MKEYS_VPP21 function block has to be **MK01In**.

The states of the hardware keys are transferred to the MKEYS_VPP21 function block via the user interface.

⚠ CAUTION

Danger with safety-critical machine functions!

⇒ Since the PLC cannot reliably detect whether the hardware keys are released, this function block may not be used for safety-critical machine functions.

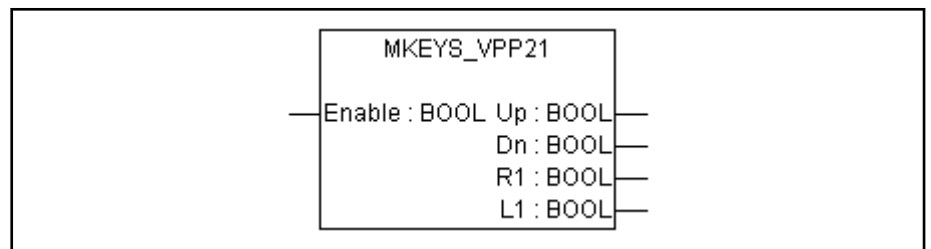


Fig. 5-44: MKEYS_VPP21 function block

Input/output	Description
Enable	Enables (= TRUE) or disables (= FALSE) the block
Up	It is linked to SelUp of the MKEYS_SINGLESEL function block
Dn	It is linked to SelDn of the MKEYS_SINGLESEL function block
L1	It is linked to L1 of the MKEYS_SINGLESEL function block

Tab. 5-3: Inputs and outputs of the MKEYS_VPP21 function block

5.5.8 Programming M-keys in the PLC for VEx devices

Library and function blocks

On installing the respective IndraLogic target (target system), the HMICE_Mkeys.lib, SysLibSymbols.lib and RIL_VExUtil.lib libraries become available.

The PLC program requires one instance of HMICE_MKEYS and one instance of IL_VEXKEYS for each visualization device.

Adding a library

At first, the HMICE_Mkeys.lib, SysLibSymbols.lib and RIL_VExUtil.lib libraries have to be added to the PLC user program. This is done using the Library Manager (see IndraLogic documentation and/or Help).

Configuring the user interface

HMICE_MKEYS function block

The HMICE_MKEYS function block supports visualization devices with foil keys (e.g. IndraControl VEH 30) and visualization devices with touch screens.

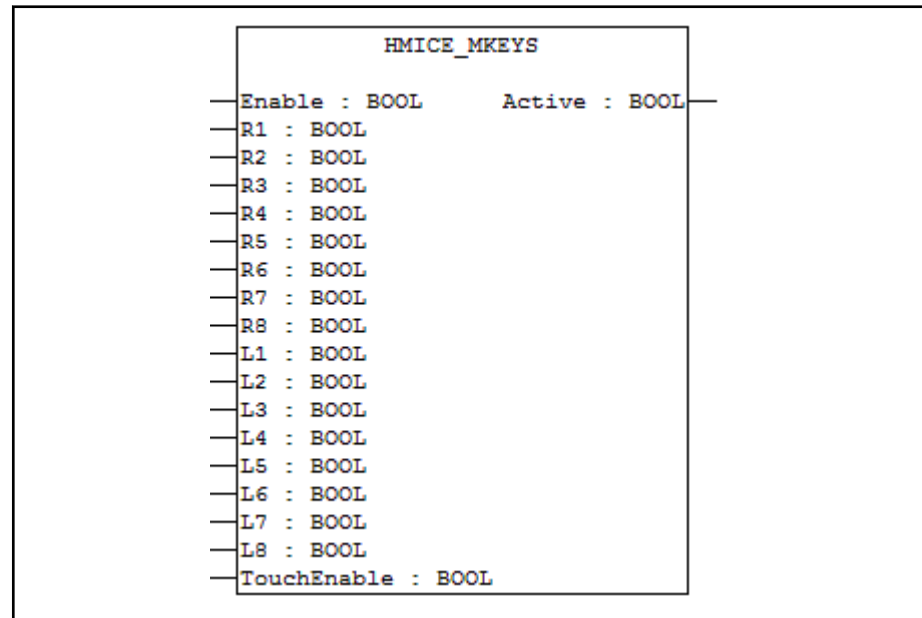


Fig. 5-45: HMICE_MKEYS function block

Input/output	Description
Enable	Enables (= TRUE) or disables (= FALSE) the block
R1-R8	Assigned with the outputs of the hardware key signals if any (e.g. VEH devices)
L1-L8	Assigned with the outputs of the hardware key signals if any (e.g. VEH devices)
TouchEnable	For the touch control, the hardware enabling key has to be assigned to this input
Active	It is TRUE if an M-key assignment is enabled

Tab. 5-4: Inputs and outputs of the HMICE_MKEYS function block

SYSLIBGETSYMBOLADDRESS function block

The SYSLIBGETSYMBOLADDRESS function block is used to resolve the address of the variable.

The "SYSLIBGETSYMBOLADDRESS" function is called by HMICE_MKEYS.

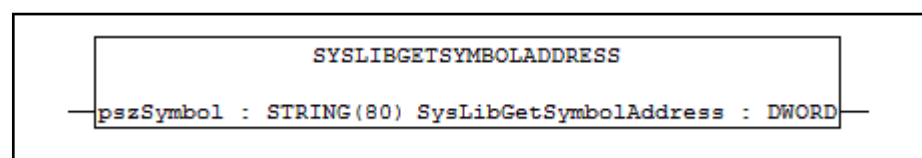


Fig. 5-46: SYSLIBGETSYMBOLADDRESS function block

Configuring the user interface

Input/output	Description
pszSymbol	Variable name
SysLibGetSymbolAddress	Returns the physical address of the variable

Tab. 5-5: Inputs and outputs of the SYSLIBGETSYMBOLADDRESS function block

IL_VEXKEYS function block

The IL_VEXKEYS function block is used to transmit the UDP signals of the foil keys to the control.

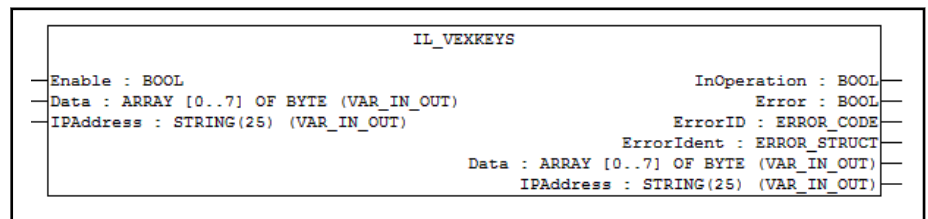


Fig. 5-47: IL_VEXKEYS function block

Input/output	Description
Enable	Enables (rising edge) or disables (= FALSE) the function block
Data	Inputs and outputs of the hardware key signal
IPAddress	Ethernet address of the VEx device
InOperation	State of the function block at runtime; OK (= TRUE), not OK (=FALSE)
Error	Error is present (= TRUE), no error (=FALSE)
ErrorID	ID for error description
ErrorIdent	Structure further describing the error

Tab. 5-6: Inputs and outputs of the IL_VEXKEYS function block

5.6 Configuring default structures

General information Use default structures to simplify the configuration of F-key flags, M-key flags and operating screens.

Configuration for F-panels To activate the default structures, the configuration dialog **Default Structures...** can be opened via the context menu of the "F-Panels" folder via the menu item (see [fig. 5-48 "Default Structures... menu item for F-panels" on page 96](#)).

Configuring the user interface

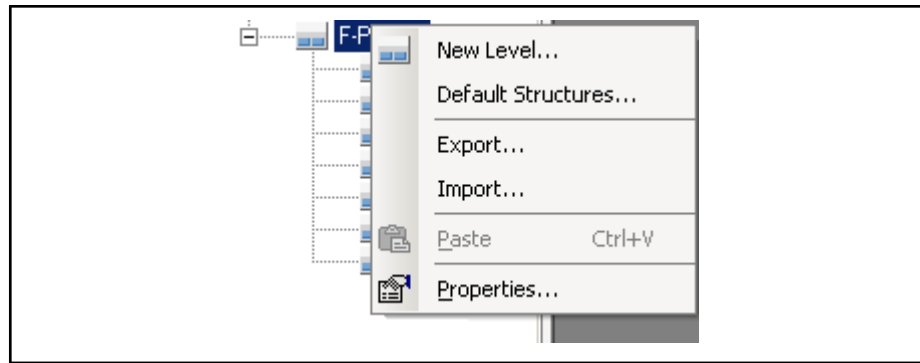


Fig. 5-48: "Default Structures..." menu item for F-panels

A list of all of the existing F-panels is provided in the configuration dialog (cf. [fig. 5-49 "Default structures dialog for F-panels" on page 96](#)). For every F-panel, the use of an individual default structure can be enabled or disabled using the checkbox in front of it. When the dialog is closed using **OK**, the configuration files of the F-panels are updated for the use of the default structures.

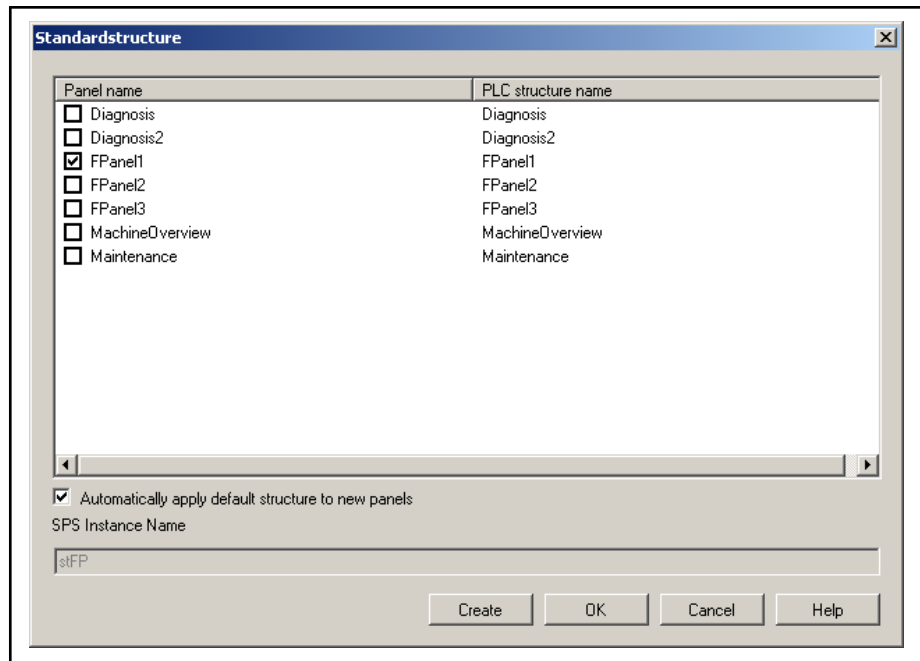


Fig. 5-49: Default structures dialog for F-panels

The "PLC structure name" column displays the PLC structure used for each F-panel. This is identical to the panel name for every default setting. The structure name can be edited by double-clicking on it, provided that there are no ambiguities with instances of other F-panels (cf. [fig. 5-50 "Input of the structure name for F-panels" on page 97](#)).

Configuring the user interface

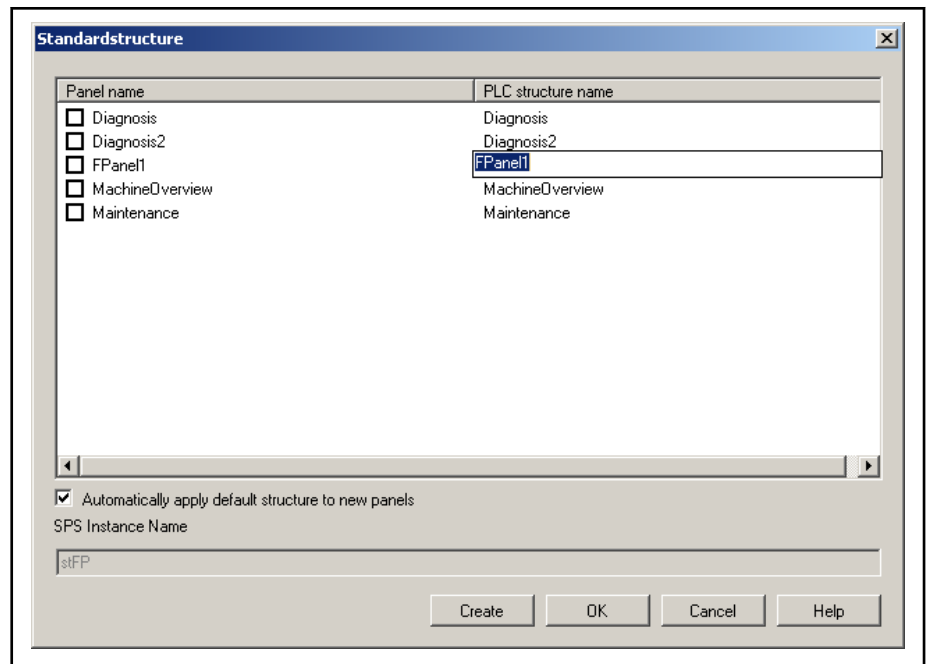


Fig. 5-50: Input of the structure name for F-panels

If the new structure name already exists, the lines of the duplicate structure names are highlighted in red (cf. [fig. 5-51 "Duplicate structure name for F-panels" on page 97](#)). In this case, **OK** is grayed out and only the lines highlighted in red can be edited.

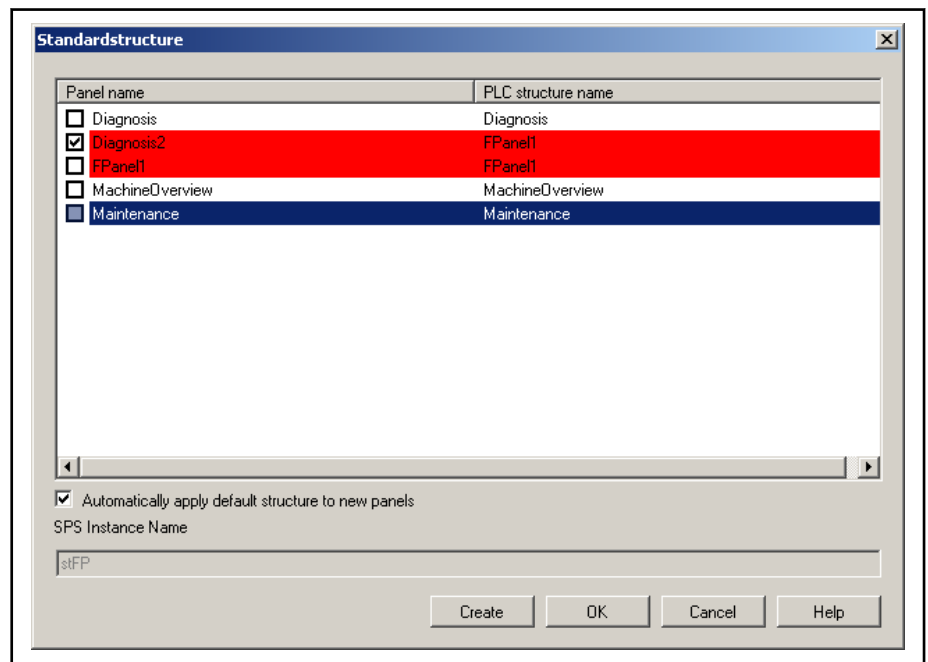


Fig. 5-51: Duplicate structure name for F-panels

The option "Automatically apply default structure to new panels" can be used to define whether default structures should be used with a new F-panel, so that the dialog opens automatically if a new F-panel is added (cf. [fig. 5-52 ""Standard Structures" dialog for a new F-panel" on page 98](#)).

Configuring the user interface

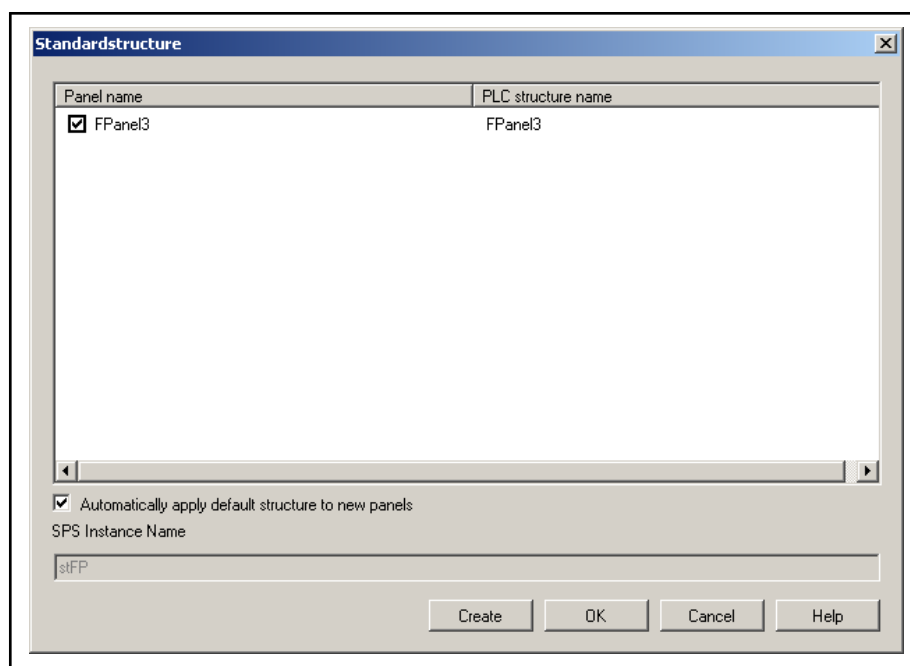


Fig. 5-52: "Standard Structures" dialog for a new F-panel

The "PLC Instance Name" text field displays an instance name of a superordinated structure, which contains all of the required instances of the structure. Clicking on **OK** saves the configurations and closes the dialog. Clicking on **Create** shows a "Save As" dialog in order to define the filing location of the file (see fig. 5-53 "Export of default structures in IndraLogic 1.xx for F-panels" on page 98 and fig. 5-54 "Export of the default structures in IndraLogic 2G for F-panels" on page 99). For IndraLogic 1.xx, the file has the extension "exp"; for the IndraLogic 2G "iwx". Finally, Import the file into the assigned IndraLogic project. If the file does not yet exist, then the name "FPanelDefaultStruct" is provided.

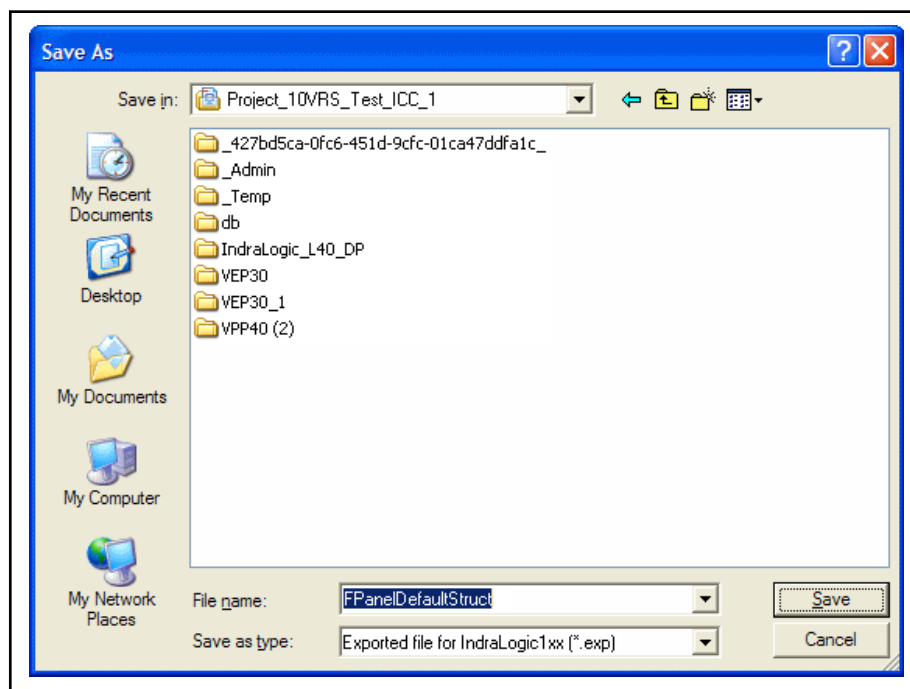


Fig. 5-53: Export of default structures in IndraLogic 1.xx for F-panels

Configuring the user interface

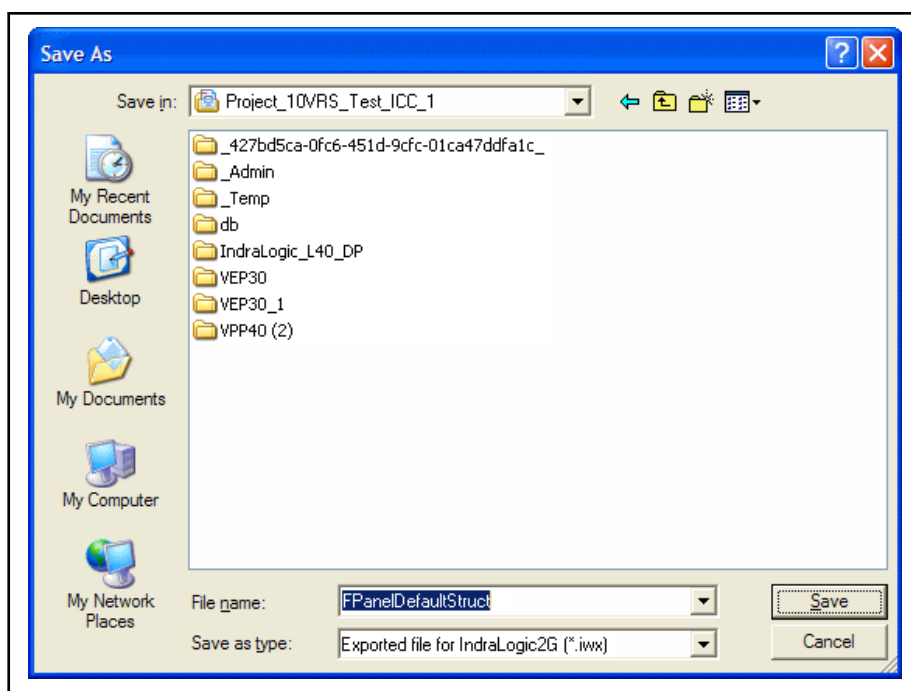


Fig. 5-54: Export of the default structures in IndraLogic 2G for F-panels

If the use of default structures for an F-panel has been enabled, the PLC flags are entered on the F-keys according to the defined structure. Should PLC flags have already been configured on an F-key, these are overwritten by the flags of the default structure. The input fields for the PLC flags are disabled in this case (cf. fig. 5-55 "Default structures in the F-panel editor" on page 99).

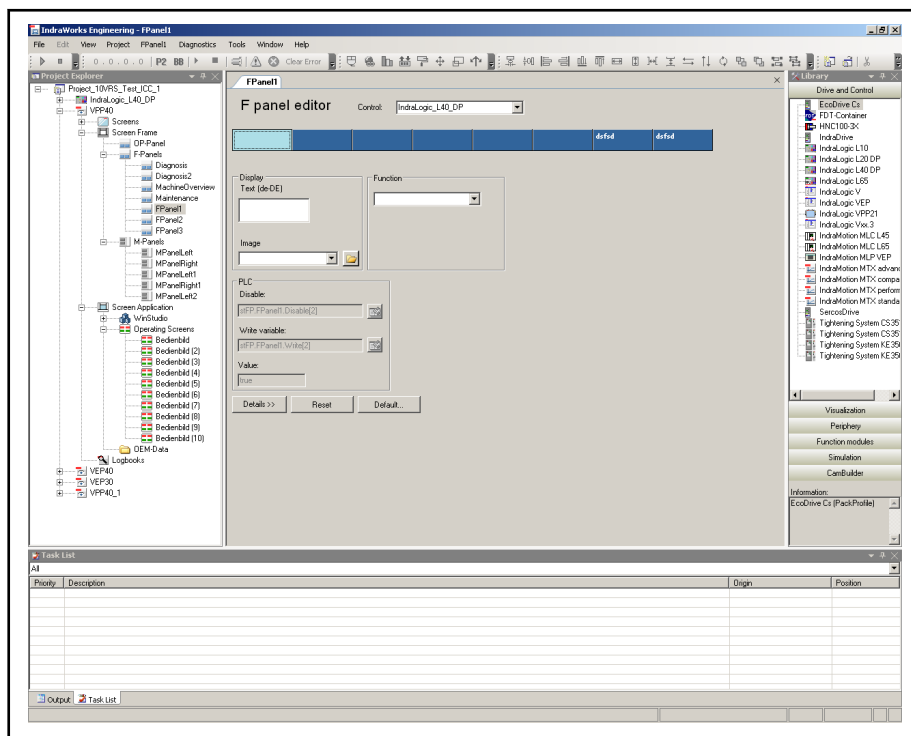


Fig. 5-55: Default structures in the F-panel editor

Configuring the user interface

Configuration for M-panels

It is projected via the **Default Structures...** dialog that is opened via the context menu of the folder "M-Panels" (see [fig. 5-56 "Default Structures... menu item for M-panels" on page 100](#)).

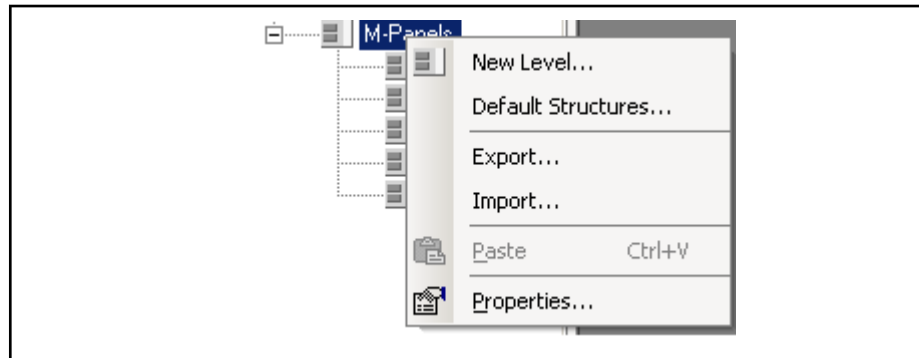


Fig. 5-56: "Default Structures..." menu item for M-panels

A list of all existing M-panels is provided in the configuration dialog (cf. [fig. 5-57 "Standard Structures" dialog for M-panels" on page 100](#)). For every M-panel, the use of an individual default structure can be enabled and disabled using the checkbox in front of it. When the dialog is closed using **OK**, the configuration files of the M-panels are updated for the use of the default structures.

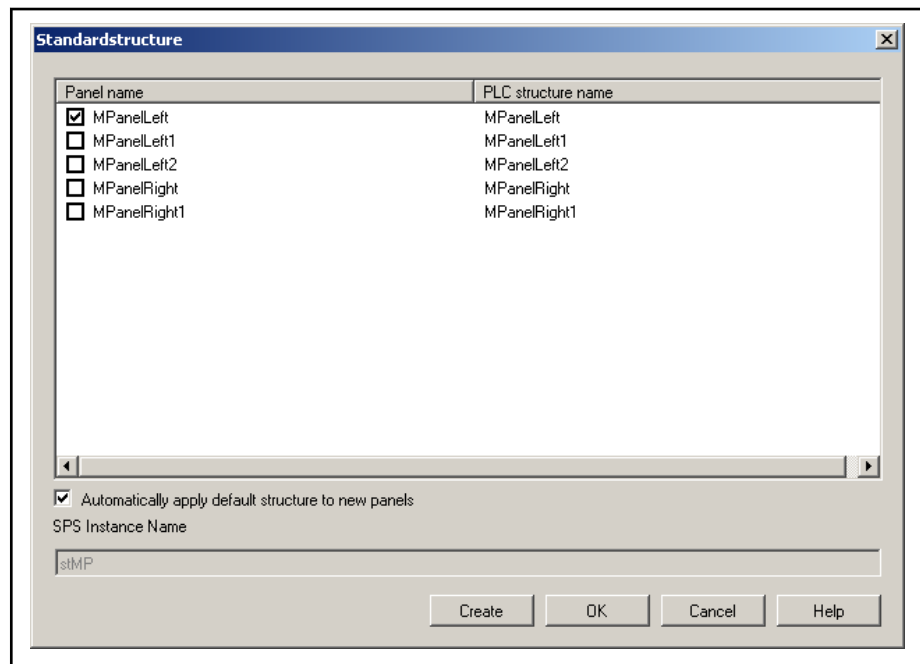


Fig. 5-57: "Standard Structures" dialog for M-panels

The "PLC structure name" column displays the PLC structure used for each M-panel. This is identical to the panel name for every default setting. The structure name can be edited by double clicking on it, provided that there are no ambiguities with instances of other M-panels (cf. [fig. 5-58 "Input of the structure name for the M-panel" on page 101](#)).

Configuring the user interface

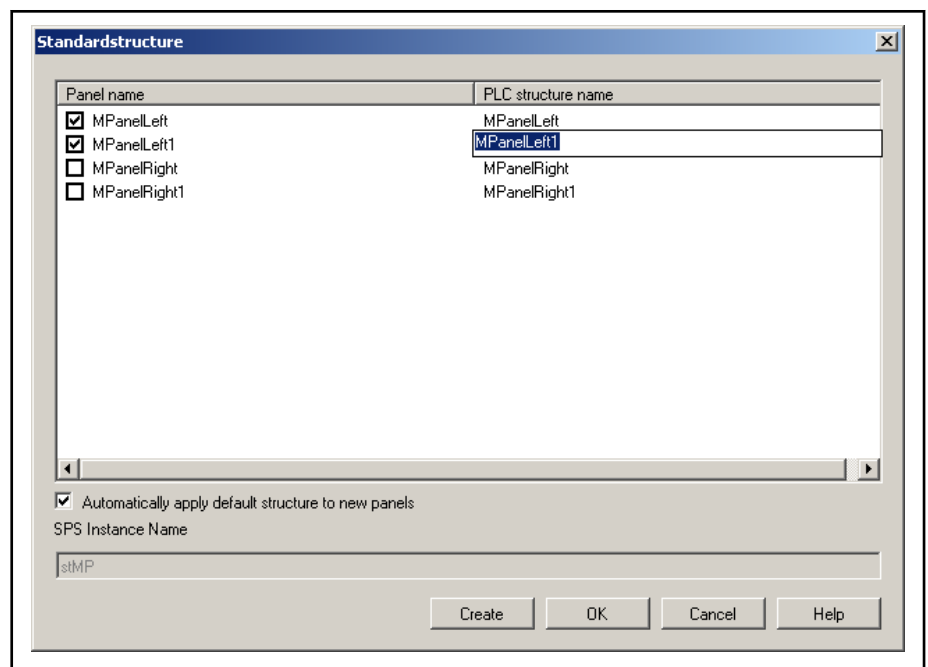


Fig. 5-58: Input of the structure name for the M-panel

If the new structure name already exists, the lines of the duplicate structure names are highlighted in red (cf. [fig. 5-59 "Duplicate structure name for M-panels" on page 101](#)). In this case, **OK** is grayed out and only the lines highlighted in red can be edited.

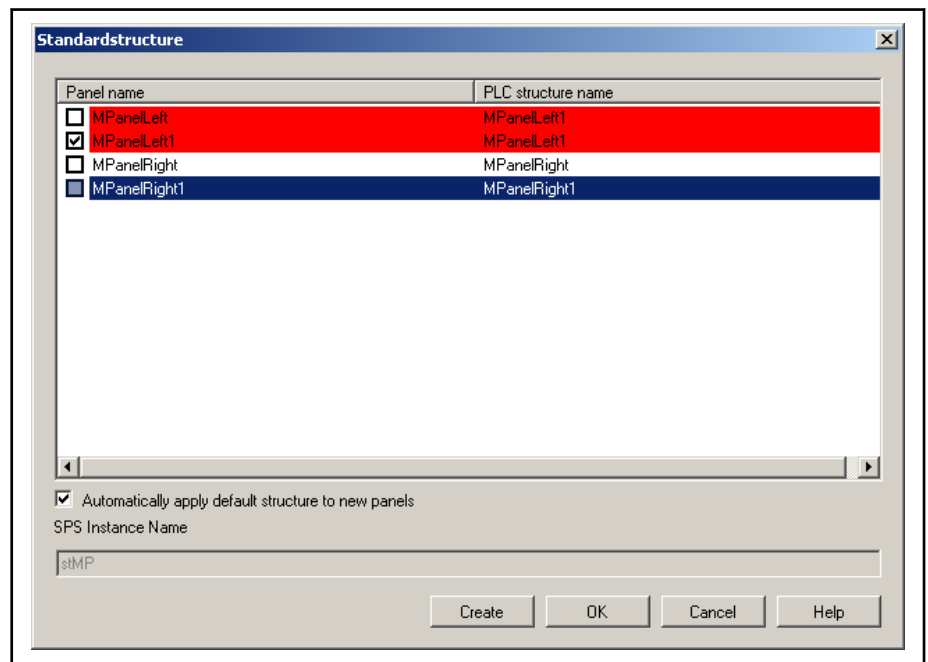


Fig. 5-59: Duplicate structure name for M-panels

The option "Automatically apply default structure to new panels" can be used to define whether default structures should be used with a new M-panel, so that the dialog opens automatically if a new M-panel is added (cf. [fig. 5-60 ""Standard Structures" dialog for a new M-panel" on page 102](#)).

Configuring the user interface

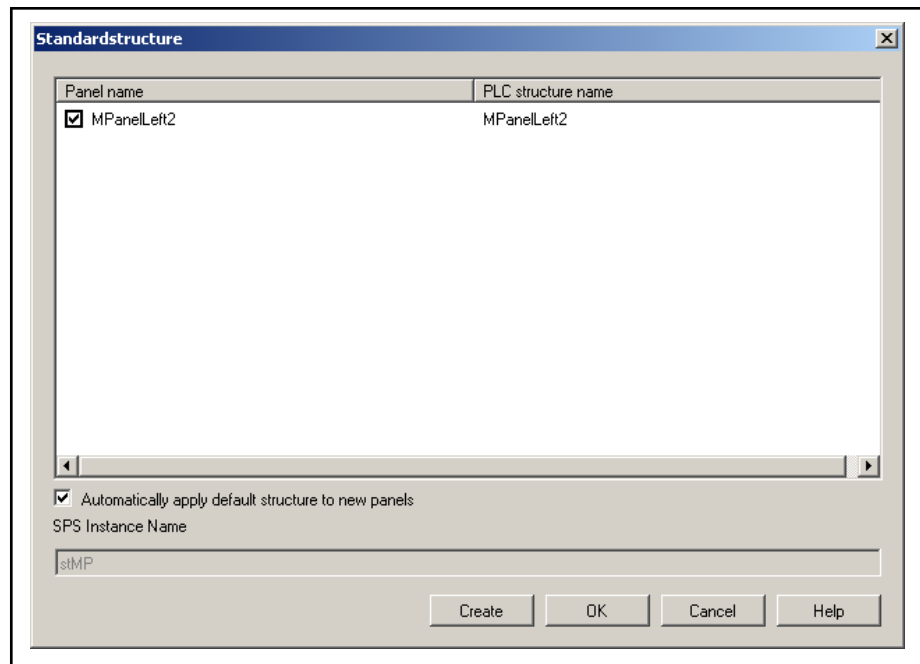


Fig. 5-60: "Standard Structures" dialog for a new M-panel

The "PLC Instance Name" text field displays an instance name of a superordinated structure, which contains all of the required instances of the structure. Clicking on **OK** saves the configurations and closes the dialog. Clicking on **Create** shows a "Save As" dialog in order to define the filing location of the file (see [fig. 5-61 "Export of default structures in IndraLogic 1.xx for F-panels" on page 103](#) and [fig. 5-62 "Export of the default structures in IndraLogic 2G for M-panels" on page 103](#)). For IndraLogic 1.xx the file has the extension exp; the link from the PLC instance name (stMP) to the symbol file is also produced. The extension for IndraLogic 2G is iwx. This file can be imported into the assigned IndraLogic project. If the file does not yet exist, then the name "MPanelDefaultStruct" is provided.

Configuring the user interface

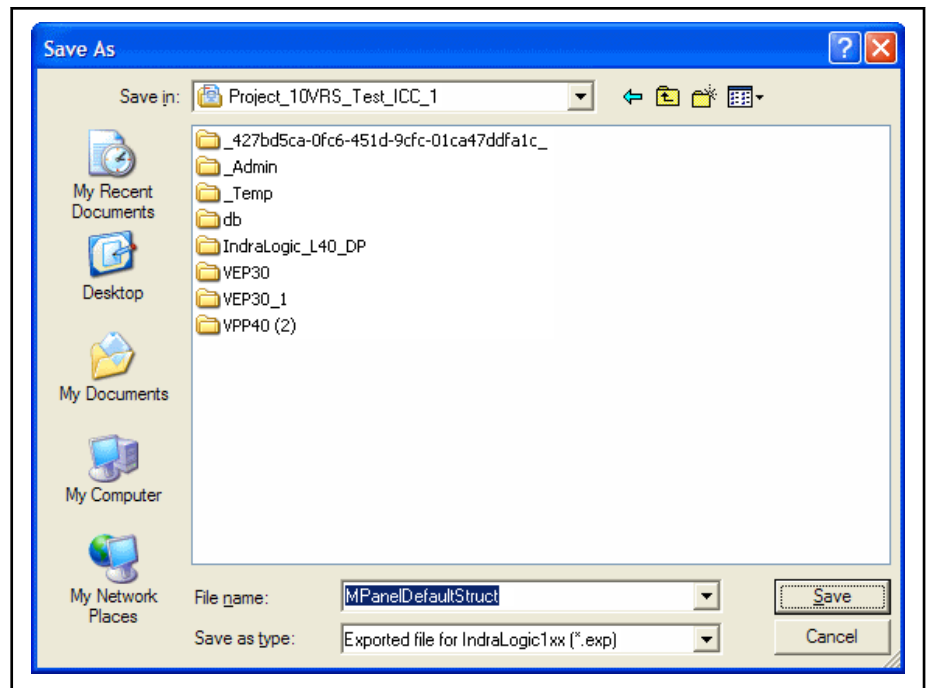


Fig. 5-61: Export of default structures in IndraLogic 1.xx for F-panels

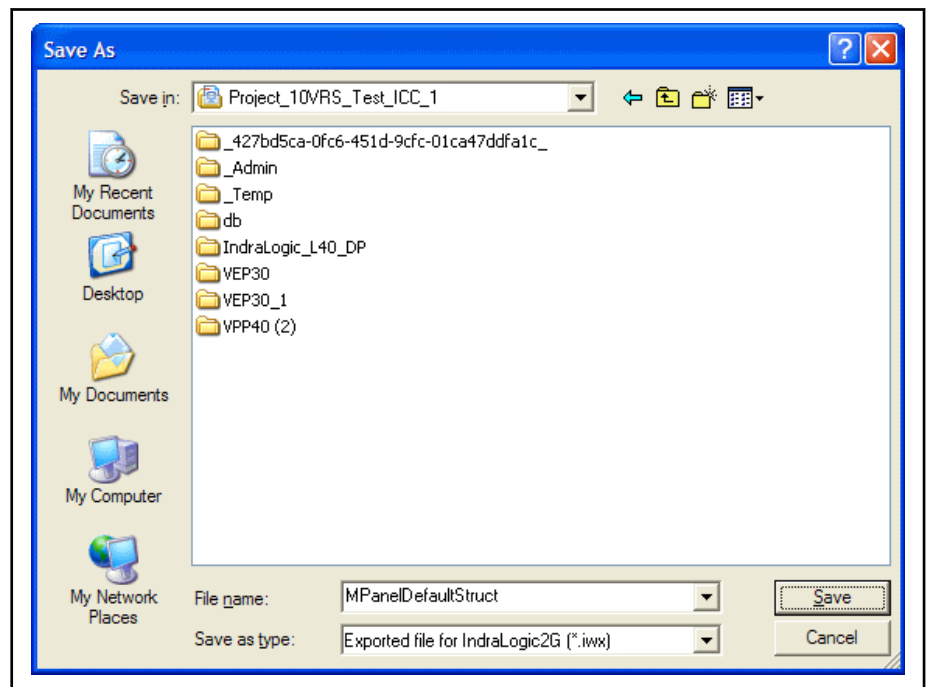


Fig. 5-62: Export of the default structures in IndraLogic 2G for M-panels

If the use of default structures for an M-panel has been enabled, the PLC flags are entered on the M-keys according to the defined structure. If SFC flags have already been configured on an M-key, the configuration remains unchanged. If PLC flags have already been configured on an M-key, they are overwritten by the flags of the default structure. The input fields for the PLC flags have been disabled in this case. If a single key is changed from "SFC flags" to "Free flags" in the M-panel editor, the default variables are automatically inserted and the input is grayed out (cf. [fig. 5-63 "Standard structures in the M-panel editor" on page 104](#)).

Configuring the user interface

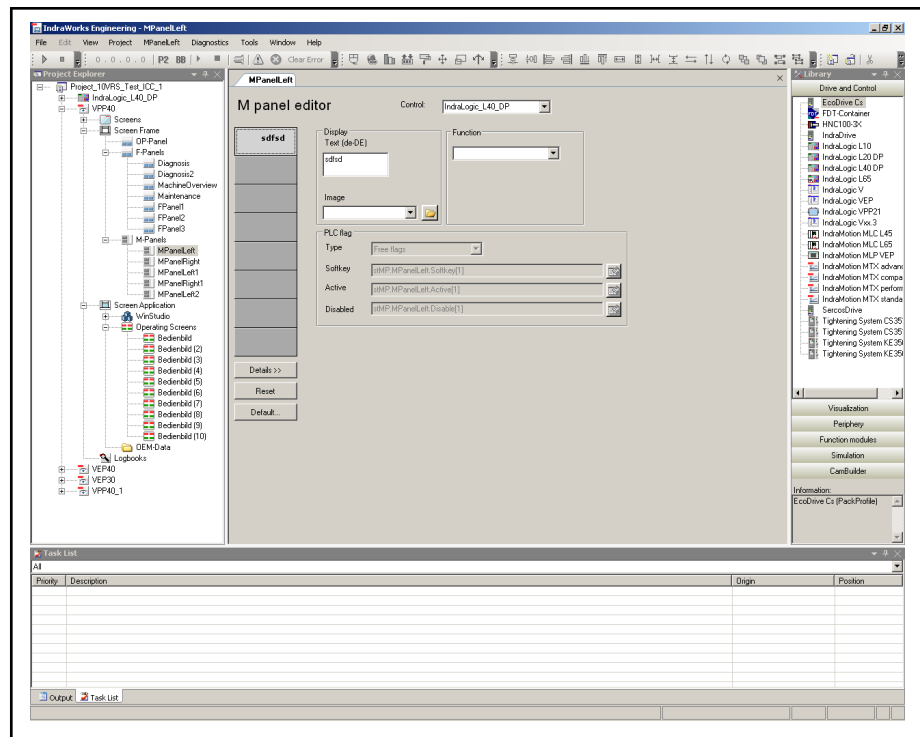


Fig. 5-63: Standard structures in the M-panel editor

Configuration for operating screens

It is projected via the **Standard Structures...** dialog that is opened via the context menu of the folder "Operating Screens" (see [fig. 5-64 "Default Structures... menu item for operating screens" on page 104](#)).

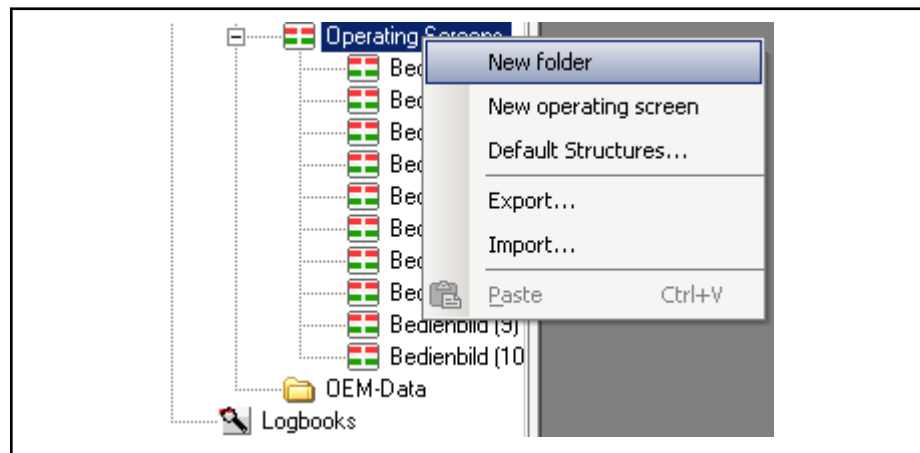


Fig. 5-64: "Default Structures..." menu item for operating screens

A list of all existing operating screens is provided in the configuration dialog (cf. [fig. 5-65 ""Default Structures" dialog for operating screens" on page 105](#)). The use of a default structure can be enabled and disabled for every operating screen using the checkbox in front of it. When the dialog is closed via **OK**, the configuration files of the operating screens are updated for the use of the default structures.

Configuring the user interface

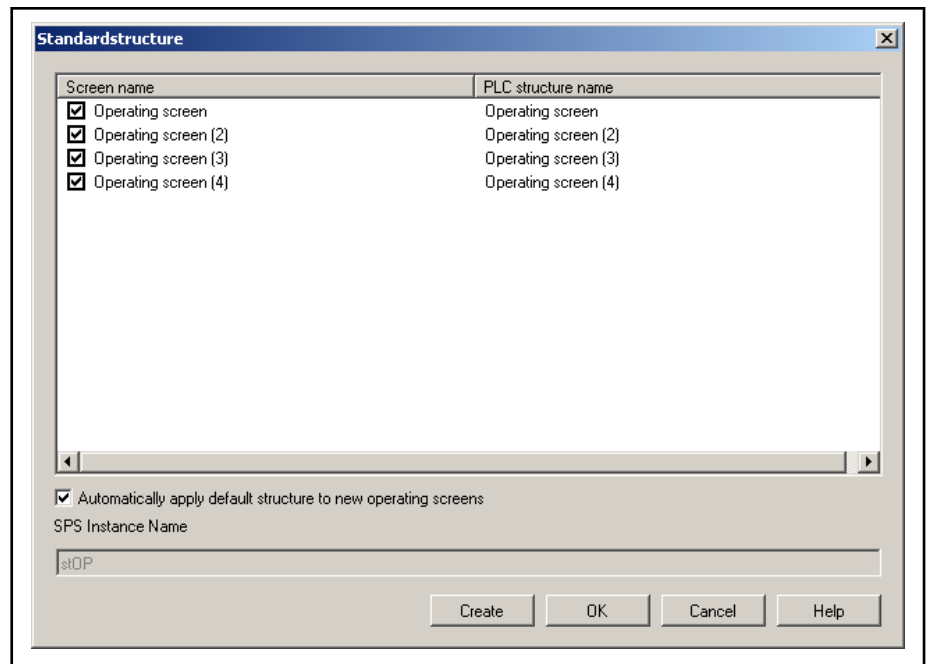


Fig. 5-65: "Default Structures" dialog for operating screens

The "PLC structure name" column displays the PLC structure used for each operating screen. This is identical to the panel name for every default setting. This structure name can be edited by double clicking on it, provided that there are no ambiguities with instances of other operating screens (cf. [fig. 5-66 "Input of the structure name for the operating screens" on page 105](#)).

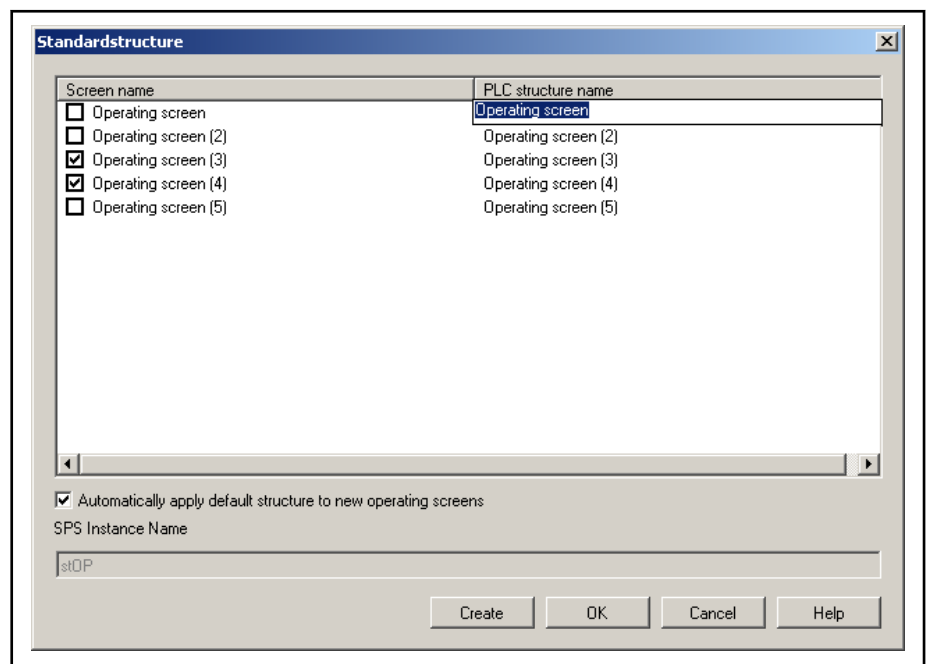


Fig. 5-66: Input of the structure name for the operating screens

If the new structure name already exists, the lines of the duplicate structure names are highlighted in red (cf. [fig. 5-67 "Duplicate structure names for an operating screens" on page 106](#)). In this case, **OK** is grayed out and only the lines highlighted in red can be edited.

Configuring the user interface

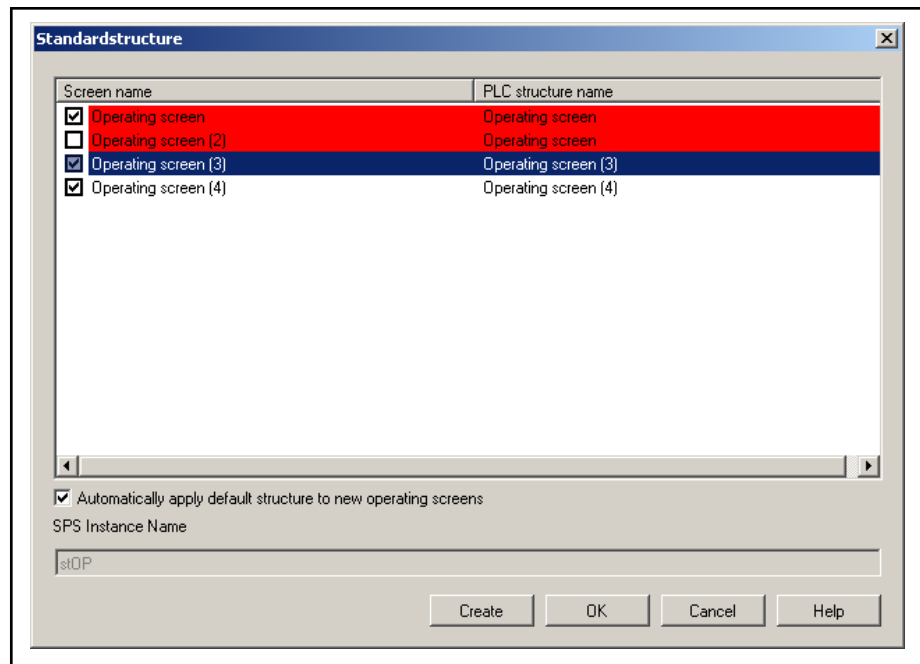


Fig. 5-67: Duplicate structure names for an operating screens

The option "Automatically apply default structures to new screens" can be used to define whether default structures should be used with a new operating screen, so that the dialog opens automatically if a new operating screen is added (cf. fig. 5-68 "Default structures dialog for new operating screen" on page 106).

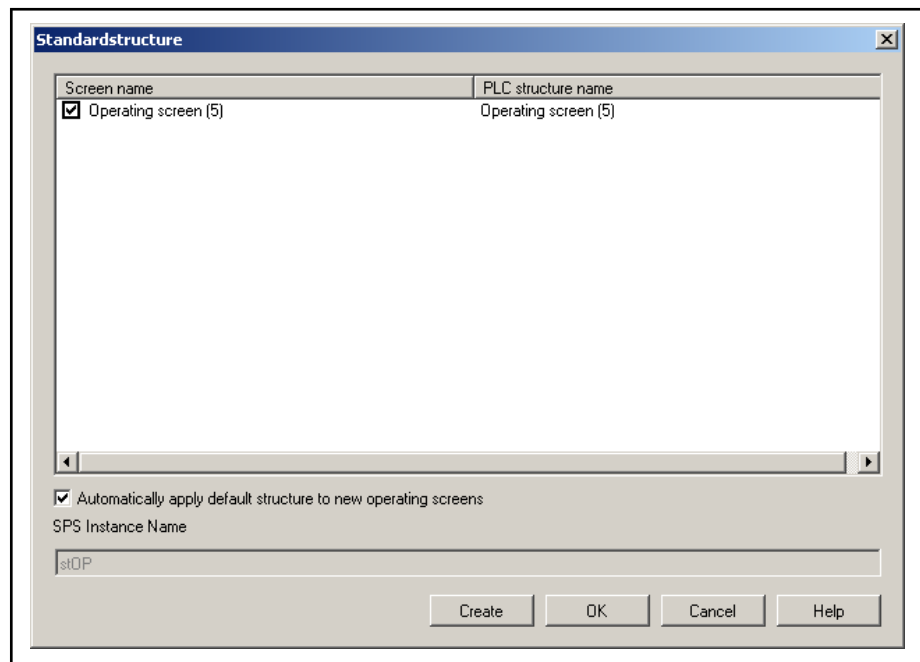


Fig. 5-68: Default structures dialog for new operating screen

The "PLC Instance Name" text field displays an instance name of a superordinated structure, which contains all of the required instances of the structure. Clicking on **OK** saves the configurations and closes the dialog. Clicking on **Create** shows a "Save As" dialog in order to define the filing location of the file (see fig. 5-69 "Export of the default structures in IndraLogic 1.xx for an operating screen" on page 107 and fig. 5-70 "Export of the default structures

Configuring the user interface

in IndraLogic 2G for an operating screen" on page 107). For IndraLogic 1.xx, the file has the extension "exp"; for the IndraLogic 2G "iwx". This file can be imported into the assigned IndraLogic project. If the file does not yet exist, then the name "OPScreenDefaultStruct" is provided.

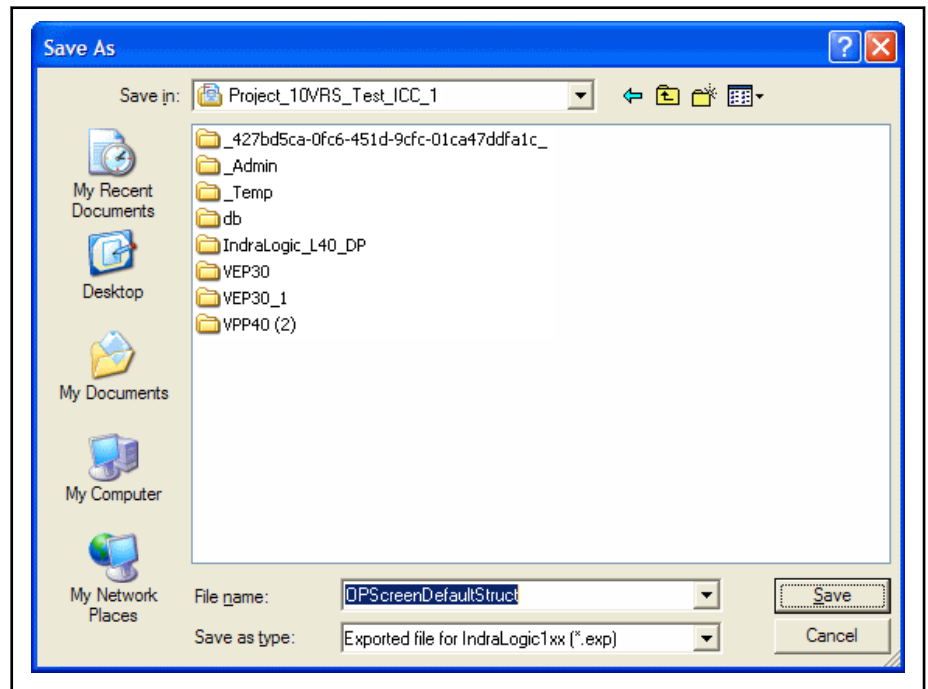


Fig. 5-69: Export of the default structures in IndraLogic 1.xx for an operating screen

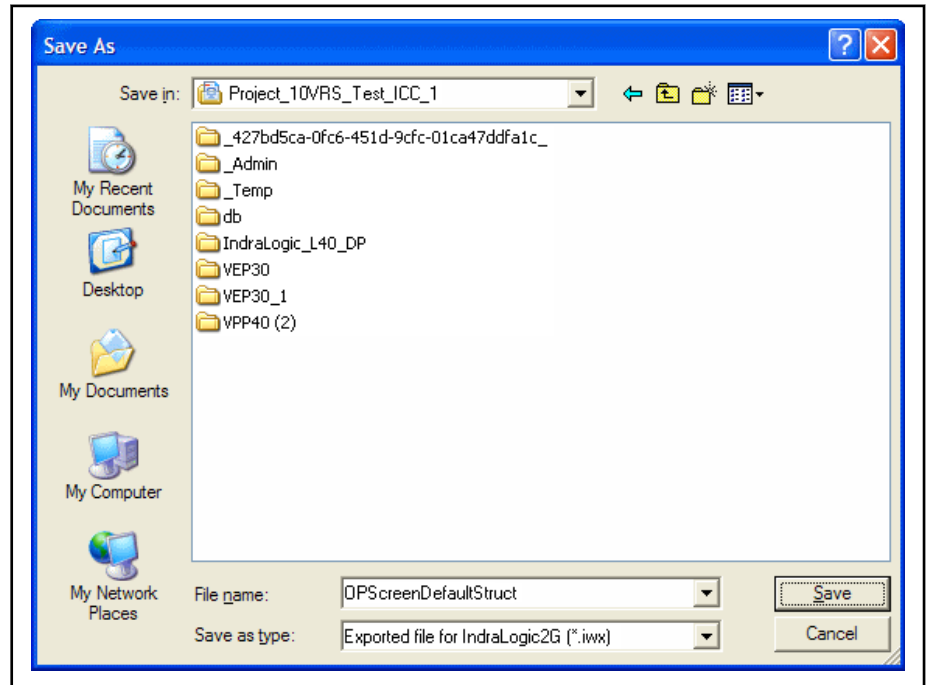


Fig. 5-70: Export of the default structures in IndraLogic 2G for an operating screen

If the use of default structures for the operating screen has been enabled, the PLC flags are entered into the fields of the 5- or 7-line operating screen according to the defined structure. The input fields for the PLC flags in the "Manual" dialog are disabled in this case. If SFC flags were configured on the

Configuring the user interface

M-key assigned to the operating line, the feasibility display is not taken into consideration by the default structure. If a single key is changed from "SFC flags" to "Free flags" in the M-panel editor or in the operating screen, the default variables are automatically inserted and the input given is grayed out. The corresponding operating screen element "Feasibility" in the operating screens is also automatically changed again to a default variable (cf. [fig. 5-71](#) "Default structures in the operating screen editor" on page 108).

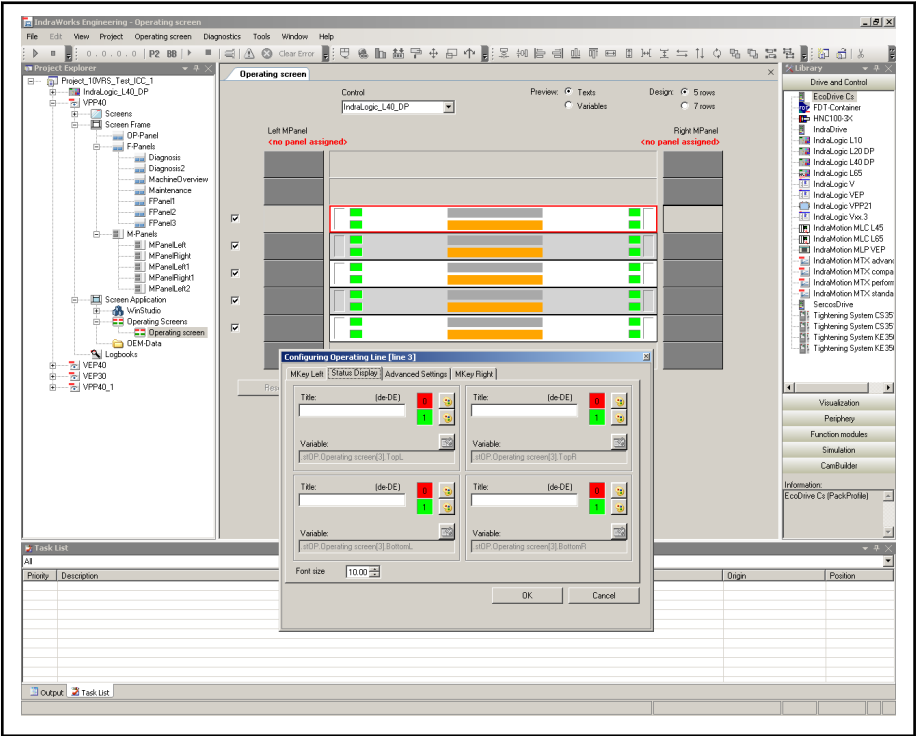


Fig. 5-71: Default structures in the operating screen editor

5.7 OEM Data

Intended Use

Folders for a structured management of any files with the option to create installation packages from them that can be transmitted and installed on the operating station.

Adding "OEM Data" to the project

The "OEM Data" element is located in the library under "Visualization". Drag the element from the library and drop it on the topmost node (project node). The element can only be created once on project level.



The functionality for the installation packages described in this chapter is not provided in "old" OEM data elements located below an HMI node ("Visualization Devices" node).

It is strongly recommended to move the data located in these OEM data elements and to delete this element here so that only one single, global OEM data element exists in the project in which all project-comprehensive OEM data is located.

To move data, drag&drop, export&import and "Save As..." are available.

Installation packages

This is a collection of installation packages (*.msi) that can be transmitted to the operating station and installed there.

Configuring the user interface

To add a file to the "Installation Packages" folder, select the following menu item **Add...** in the context menu of the folder. A dialog opens to select an installation package that should be copied to the project and managed there.

Select **Transmit...** to transfer all installation packages to the operating station.



Each change in the installation package generally requires a new transmission to the operating station. The respective symbol on the "Installation Packages" folder indicates that a new transmission is required. The installation packages are transmitted via the context menu item **Transmit...** In the transmission dialog, enter the IP address of the operating station to which the installation packages should be sent. The name or the IP address of the visualization device is entered as default.

Installation package configurations

OEM data is managed in specified IndraWorks categories (executable files, help files, configuration file) within the project. An installation package (*.msi) can be created from this installation package configuration transferred to an operating station and installed there.

Advantages

- Windows Standard (*.msi)
- Optional removal of the installation package from the operating station when uninstalling IndraWorks
- Manual removal of the operating station via Control Panel -> Software

Installation package configuration

Select **New Configuration** to create a new installation package configuration. A specified folder structure is created for each configuration.

Add the files to be installed according to their usage:

- **Executable files:**
.exe;.dll;*.ocx;*.tlb;*.pdb;*.cmd;*.bat
- **Configuration files:**
.xml;.ini;*.xmi;*.ini; all files
.resources.dll;.xml;*.txt
- **Help files:**
.chm;.doc;*.pdf;*.html;*.txt; all files

The files are restored on the respective correct location of the IndraWorks runtime environment on the operating station:

- **Executable files:**
%startup%
Example: C:\Program Files\Rexroth\IndraWorks
- **Configuration files:**
%config%
Example: C:\Program Files\Rexroth\IndraWorks\config
%startup%\{Localization}
Example: C:\Program Files\Rexroth\IndraWorks\de
Example: C:\Program Files\Rexroth\IndraWorks\en
- **Help files:**
%config%\Help\{Localization}
Example: C:\Program Files\Rexroth\IndraWorks\Help\de

Configuring the user interface

Example: C:\Program Files\Rexroth\IndraWorks\Help\en



.NET UserControls implementing the ACI interface have to be provided with the "COMVisible=true" property and the respective type library (*.tlb) has to be added as well.



Files being an integral part of IndraWorks cannot be added to the configuration.



To add new localization folders (e.g. "de", "en") to the configuration, simply add the desired project language. A localization folder can only be deleted again if the language from the project languages used has been removed.

Creating installation packages

Select **Create Installation Package** to create a new installation package.

Specify the following installation package properties:

1. Name
2. Published by
3. Comment
4. Version
5. Remove from the operating station when uninstalling IndraWorks



Enter a new name for the installation package or select an existing installation package which should be completely replaced "Major Upgrade". In case of replacement, a higher version has to be entered.

Observe the IndraWorks output window regarding progress, notes, warnings and errors during the creation. If it has been created successfully, the installation package has been added below the "Installation Packages" node. Information on the installation package can be displayed via **Properties...**, **Details...** or via the tooltip.

Installing installation packages

After one or several installation packages have been transferred to the operating station, the user is notified that new installation packages are available.



It is installed on the operating station if IndraWorks Operation is active. The installation requires administrative user rights.

If the user is not provided with these Windows administrator rights, a respective login dialog appears. If this is canceled or the user is not able to enter the right name/password combination, the installation is not carried out and IndraWorks Operation is started again.

If there are several installation packages, these are called and processed in sequence. When denying the note on the installation, the user can start the installation at a later start of IndraWorks Operation or delete the installation packages.



Fig. 5-72: Deleting OEM data

If the OEM data is transferred at runtime, a message that there is new OEM data appears.

⚠ WARNING

Uninstalling IndraWorks

If at least one installation package was installed on the operating station, IndraWorks can only be uninstalled via **Start ► All Programs ► Rexroth ► Tools ► IndraWorks Manager**

If at least one installation package has been installed on the operating station for which the option "Remove from the operating station when uninstalling IndraWorks" was **not** selected, the installation directory has not been completely moved to the recycle bin, but remains with the OEM installation packages and residual IndraWorks files!

5.8 HMI functions

The following functions are available for the application type "HMI Standard". Depending on the device version and the function type, they may be assigned to F/M/OP-keys.

HMI functions

Function	Applicable	Description	Device Version
Close active screen	F/M-panel	Closes the active screen	
Device overview/details	F/M-panel	Opens the detailed view of the device overview	
Device overview/filter	F/M-panel	Filter for the device overview	
Save user data	F/M-panel	Saves user-specific settings in IndraWorks Operation	
Enabling the workspace	OP-panel	Displays the transferred workspace and closes all screens of the active workspace	
Operating screen information	F/M-panel	The state display in the operating screen toggles between value, text and I/O address	
Exit IndraWorks Operation	F/M/OP-panel	Exits IndraWorks Operation	
Command to ACI screen	F/M-panel	Use this function to send various commands to the ACI screen.	
Command to ACI Control	F/M-panel	Use this function to send various commands to the ACI Control.	

Configuring the user interface

Function	Applicable	Description	Device Version
Logout User	F/M/OP-panel	Opens the login dialog to enter the user name and password	
Login User	F/M/OP-panel	Logs the currently logged in user out	
Screen change	F/M/OP-panel	This function switches from system to user screens	
Diagnostics/details	F/M-panel	Opens the detailed view of the diagnostic view	
Diagnostics/filter	F/M-panel	Opens a filter dialog to filter the diagnostic view	
Diagnostics/Delete	F/M-panel	Deletes the diagnostic message from the diagnostic overview	
Level change	F/M-panel	This function switches between F/M-panels	
IndraWorks Engineering	F/M/OP-panel	Starts IndraWorks Engineering	
Criteria analysis	F/M-panel	Opens the criteria analysis	
Language selection	F/M-panel	Changes the interface/project language of IndraWorks Operation.	from 13V06

Tab. 5-7: HMI function selection

5.9 Language-dependent user texts

Texts within a visualization device are always language-dependent user texts.

The editors for F-keys, M-keys and OP-keys contain a "Display" input field in which the key labels can be displayed and edited.

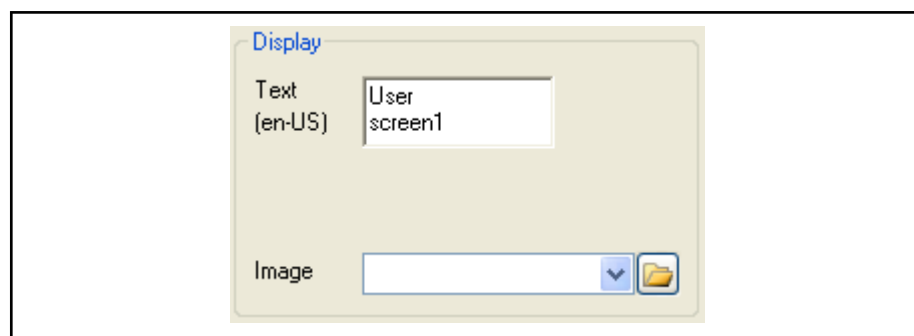


Fig. 5-73: Configuring a language-dependent key label

Language-dependent texts

The country code of the currently selected language, e.g. "en-US", is displayed below "Text". The text shown is assigned to this language and can be overwritten if necessary. To enter texts in other languages, select the **Project ► Language ► Select project language** menu item from the IndraWorks menu bar. After the language has been changed, all of the texts are immediately displayed in this language. If no text has yet been entered in the language selected, the text previously displayed remains, but can be overwritten later. The text entered, is assigned to the language currently set.

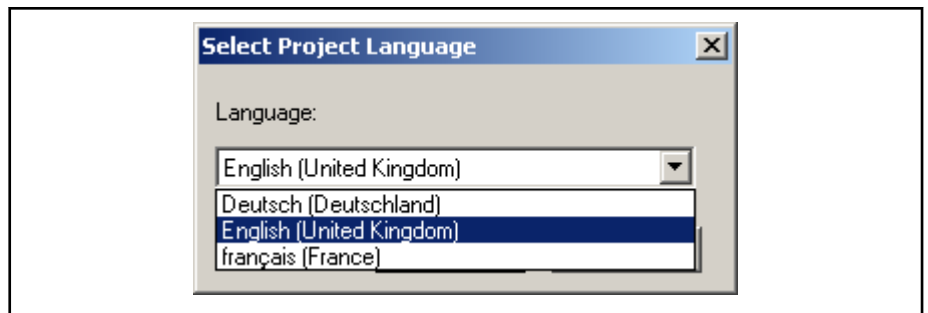


Fig. 5-74: Select project language

- Texts from resource files** Predefined F-panel, M-panel and OP-panel assignments vary depending on the control system installed. If available in several languages, the related panel labels are saved to so-called Resource files. The texts cannot be edited. The input field is disabled.
- Adding a language** To add further languages, select the **Project ► Language ► Manage Project Languages...** IndraWorks menu item.
- It is also possible to export, translate externally and then to import language-dependent texts. For a description of these functions, please refer to the IndraWorks Engineering documentation (see [chapter 1.3 "Required and supplementing documentation IndraWorks" on page 6](#)).

5.10 Configuring logbooks

5.10.1 General information

The IndraWorks project allows users to configure several logbooks. Different messages can be recorded in each of these logbooks.

5.10.2 Creating a new logbook

To create a new log book, select the **New log book** menu item from the context-sensitive menu of the "Log books" folder (to be opened by right-clicking on this folder).

Configuring the user interface

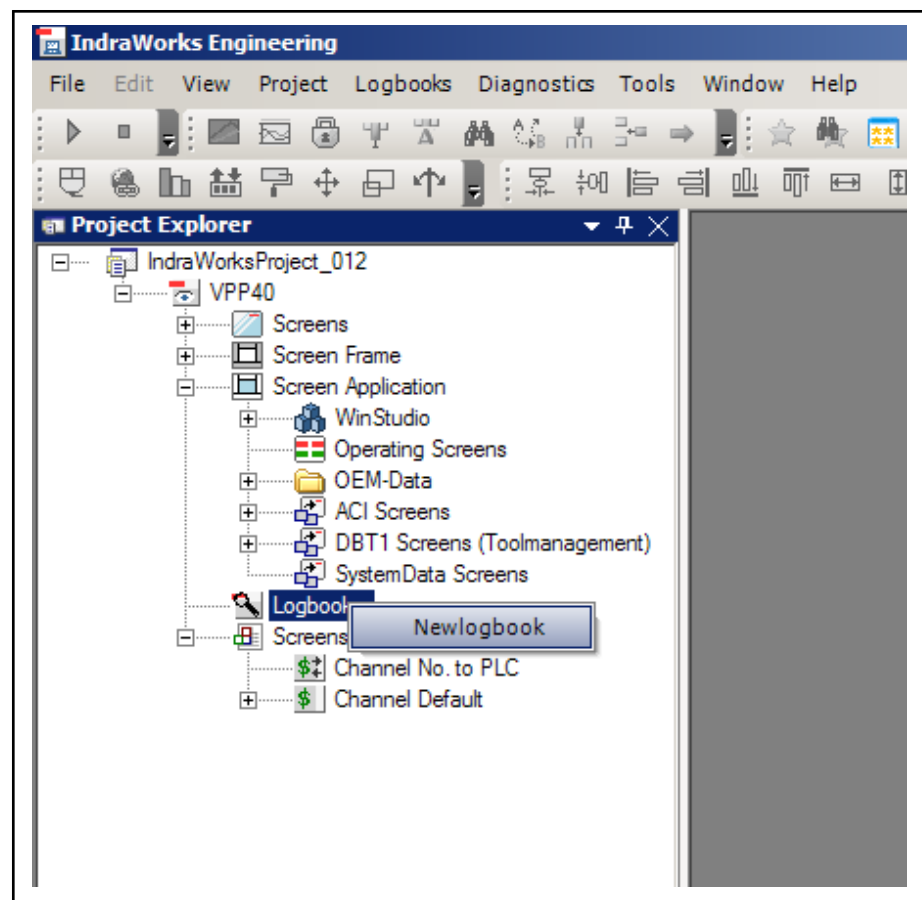


Fig. 5-75: Creating a new logbook



Initially, each logbook is created with the default name "Logbook". To edit the name, select the entry and click on it again. The entry changes to the input mode and can be edited. The **Rename** function can also be used. It can be selected in the context menu (right-clicking).

5.10.3 Logbook configuration

Double-click on the logbook to open the logbook configuration.

Configuring the user interface

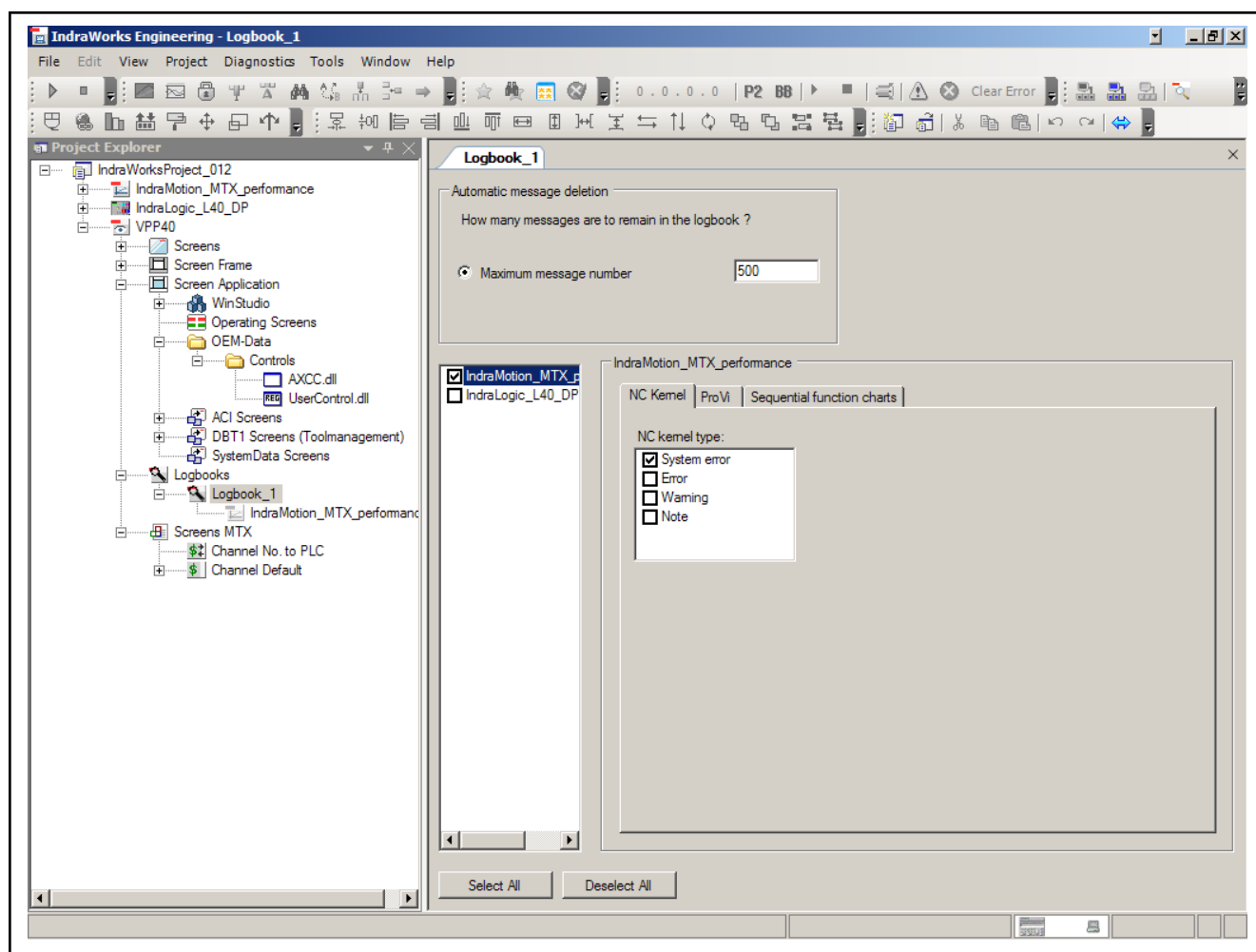


Fig. 5-76: Logbook configuration dialog

Automatic deletion of messages

Old messages in the logbook are automatically deleted. The number of messages to be saved can be specified. The default setting is the maximum number of 500 messages. If more messages are entered in the logbook than specified, the oldest messages is deleted. Messages are deleted as soon as there are more than 50 messages more than the maximum number of messages.



Only one option can be selected for automatic message deletion. It is not possible to combine the number and the date of origin of the messages.

Assigning controls

All controls (IndraMotion_MTX_performance and IndraLogic_L40_DP in the figure above) of the current IndraWorks project that can log logbook messages are displayed in the configuration dialog. Activate the desired controls which should record messages. The controls activated are also shown as subnodes of the logbook in the Project Explorer.

A control can also record in more than one logbook. For example, errors can be recorded in one logbook and warnings in another one.



Tip: Click on **Select All** and **Deselect All** to select or deselect all controls.

Specifying message types

Message types can be filtered for each control.

Configuring the user interface

A tab which can be used to configure the messages to be recorded is provided for each diagnostic type supported by the control. The fields, which can be edited, vary according to the control type.

Saving data Save the logbook configuration using the IndraWorks Engineering **File ► Save** or **File ► Exit** menu item.

5.11 Configuring WinStudio screens

5.11.1 Overview

Proprietary applications can be configured using the "WinStudio Screen" type of screen. Numerous preconfigured graphic control and display elements can be inserted to a WinStudio screen and, if necessary, linked to process variables. A WinStudio screen also allows to insert ActiveX and also .NET control elements, so that applications that were not created with WinStudio tools can also be implemented. Moreover, WinStudio applications also allow to perform complex computations using so-called WinStudio mathematics or VisualBasic scripts.

WinStudio screens can be created in the folder **Screen Application ► WinStudio**. In this folder and in further subfolders, a number of settings can be entered and all WinStudio objects required can be created.

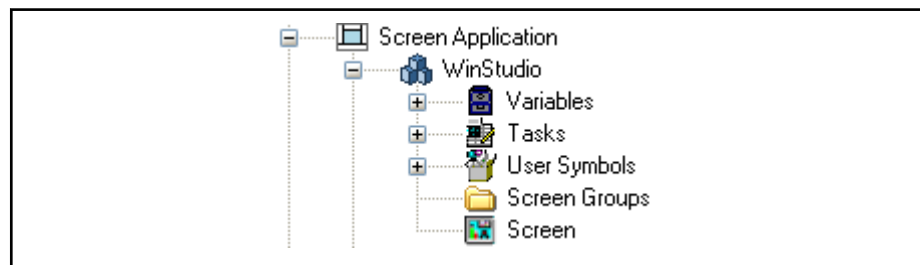


Fig. 5-77: WinStudio folder

Over and above the WinStudio screens, the follows subfolders can be found in the "WinStudio" folder:

- **Variables**
This folder contains editors for different variable types (user variables, system variables), editor for configuration of the process communication (driver screens) and for creation of own data types (classes)
- **Tasks**
Create and edit extended WinStudio objects (e.g. mathematics, alarms) in this folder
- **User Symbols**
WinStudio objects created by the user, which can be used in Win Studio screens, are displayed in this.
- **Screen groups**
The folder allows to create screen groups, which combine multiple WinStudio screens in a common screen

To create WinStudio screens and to configure extended function (e.g. processes, alarms) and the communication connection, different editors and dialogs are available. They can be opened via the context menu of the "WinStudio" or other subfolders.

The following WinStudio elements are only available if at least one license "WinStudio-Engineering und Runtime 1.5k" is enabled (see [chapter 5.11.13 "WinStudio licensing" on page 138](#)) and if as target system "WinStudio 1.5k"

Configuring the user interface

or higher is selected in the settings of the WinStudio project (see [chapter 5.11.2 "General WinStudio settings and functions" on page 117](#)):

- Alarms
- Recipes
- Reports
- Trends
- TCP/IP driver sheets
- DDE driver sheets

When opening a project in which "WinStudio 1.5k" or higher is selected as target system but no matching WinStudio license is active, the elements requiring a license are displayed. However, they are disabled and the user cannot edit them.

5.11.2 General WinStudio settings and functions

Project properties

Some basic WinStudio settings can be changed using the "Settings" dialog. This can be opened by calling up **WinStudio ► Settings....**

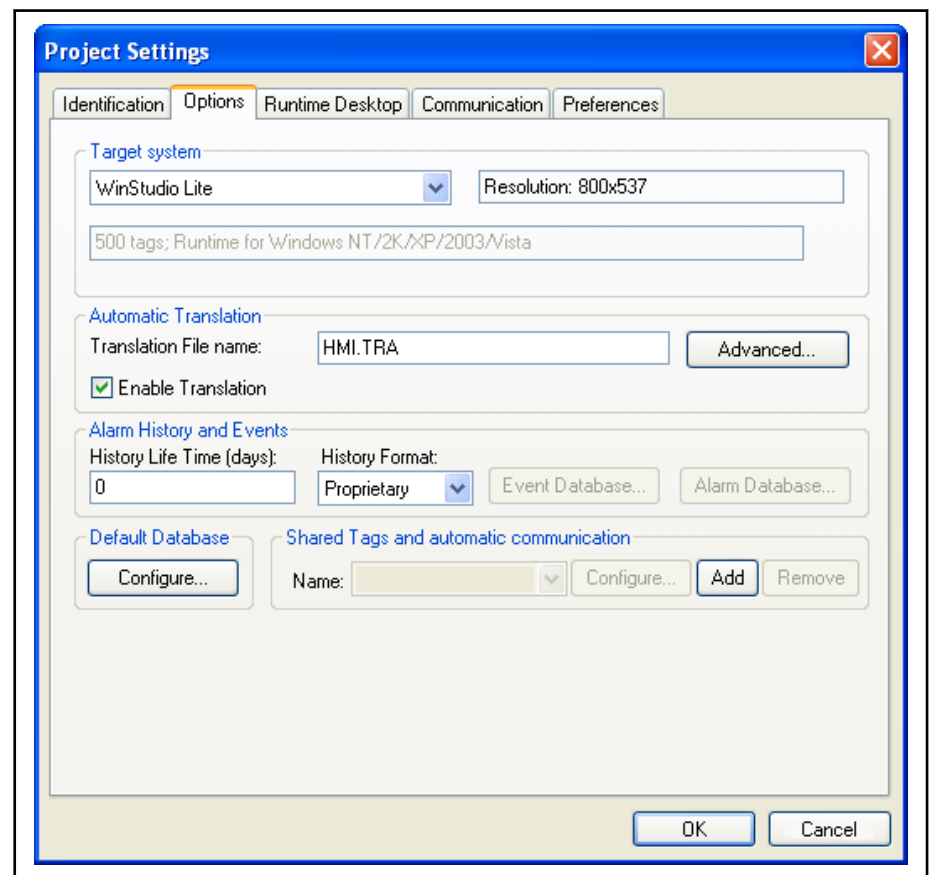


Fig. 5-78: WinStudio "Project Settings"

Key project settings:

- **Target system**

By selecting the target system, the number of available WinStudio variables, the available functional scope as well as the project type matching the target system (Windows XP/Windows 7 or Windows CE) is specified. Default setting: "WinStudio Lite". Other target systems with a higher number of variables and extended functional scope are available

Configuring the user interface

once the corresponding WinStudio Engineering license has been enabled

Please refer to the separate WinStudio documentation for a description of further options (see [chapter 1.3 "Required and supplementing documentation IndraWorks" on page 6](#)).

Project status

The runtime components have to be enabled to correctly execute some WinStudio functions. An overview of all WinStudio runtime components can be opened via **WinStudio ► Status...**

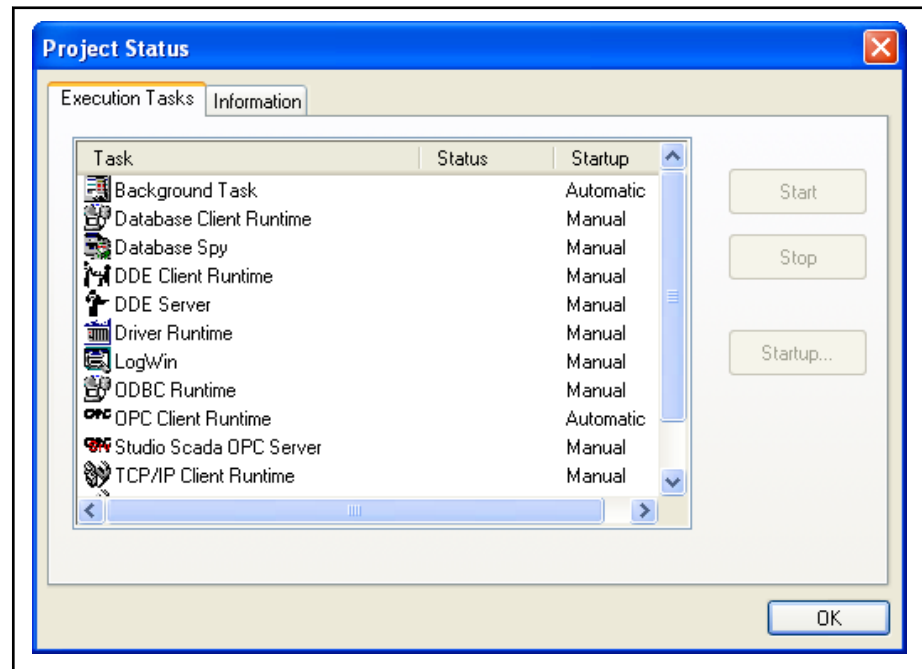


Fig. 5-79: WinStudio project settings

All available WinStudio runtime components are displayed in the status dialog. The status as well as the start mode are displayed for each runtime component.

By default, the following runtime components are enabled upon start of IndraWorks Operation:

- Background task
- OPC client runtime
- TCP/IP server

Load WinStudio project

The current WinStudio project including all WinStudio objects can be replaced by another WinStudio project to use an existing WinStudio project as template. To replace the current WinStudio project, run the function **WinStudio ► Load project...** A file browser opens with which a project file (file extension "*.app") can be selected. After the file selection has been confirmed, all current WinStudio elements are deleted and new elements of the selected project are inserted.

5.11.3 WinStudio screens

Creating WinStudio Screens

To create a WinStudio screen, select **Add ► WinStudio screen** (up to 13V04 **New screen**). A new element with the name "Screen" is added to the "WinStudio" folder. This name can be changed now or later.

Structuring WinStudio screens in folders (from 13V06)

WinStudio screens can be structured with folders. Use "Add", "Import", "Paste" or "Move" to assign the WinStudio screens to folders.

Create a new folder with the context menu function **Add ► Folder**. The number of folders and their nesting depth is not limited.

The mode of operation of the context menu functions "Add", "Export"/"Import", "Copy"/"Paste", "Delete" and "Rename" of the folders is analogous to the WinStudio screens.

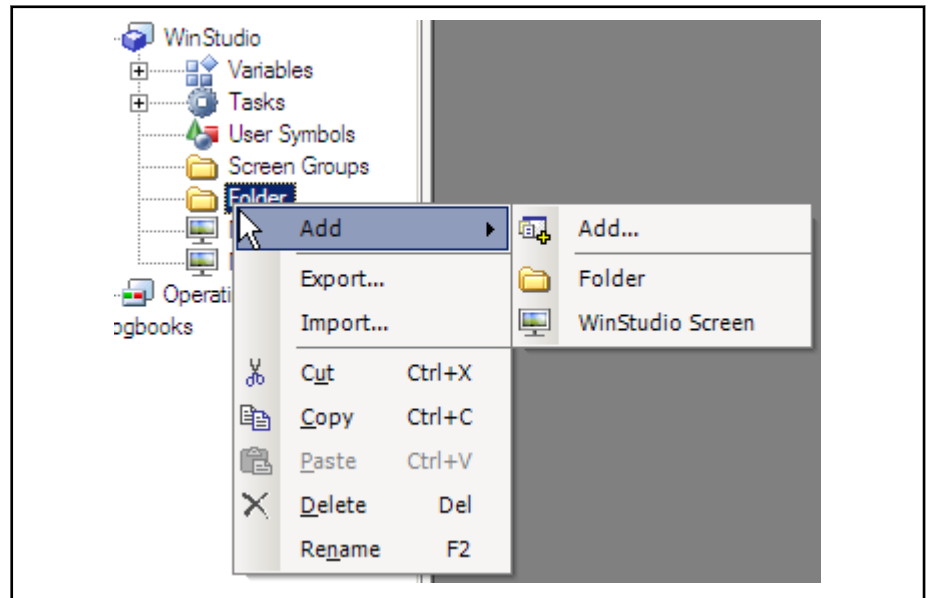


Fig. 5-80: Folder (context menu)

Editing WinStudio screens

To edit an existing WinStudio screen, open the WinStudio screen editor by double-clicking on the element in the project tree (see ① in the following figure). A tool window (②) also opens providing simple screen objects and control elements. To add a static object or a control element to a WinStudio screen, click on it in the tool window (②) and then click on the WinStudio screen (①).



Some elements can only be used if the WinStudio-Engineering license 1,5k or higher is activated. Elements with a license are disabled in case of an insufficient license.

Configuring the user interface

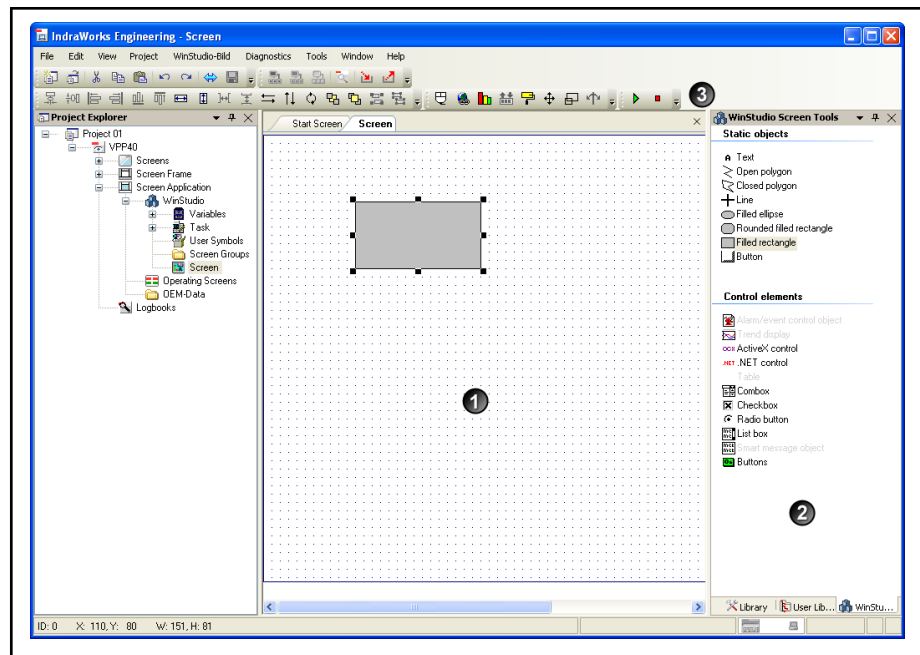


Fig. 5-81: WinStudio screen editor

When a WinStudio screen is opened, additional toolbars (③) with frequently used functions are displayed in the presetting. The presetting can be changed via **Tools ► Customize...** and the provided toolbars can be hidden or further toolbars can be added. All the editing functions of a WinStudio screen are also available via the IndraWorks Engineering menu bar via **WinStudio Screen**.

Screen attributes

There are other options available in the "Screen Attributes" to change the screen size and to use a background picture. Open the "Screen Attributes" dialog via the context menu of the "Screen Editor" via the menu item **Screen attributes**.

The relevant default values for the various systems are already entered in the "Size" field:

- MTX systems: Width: 620 Height: 412
- Other systems: Width: 620 Height: 435



Do not change the settings in the "Size" and "Location" fields, because this would result in faulty screen representations.

Configuring the user interface

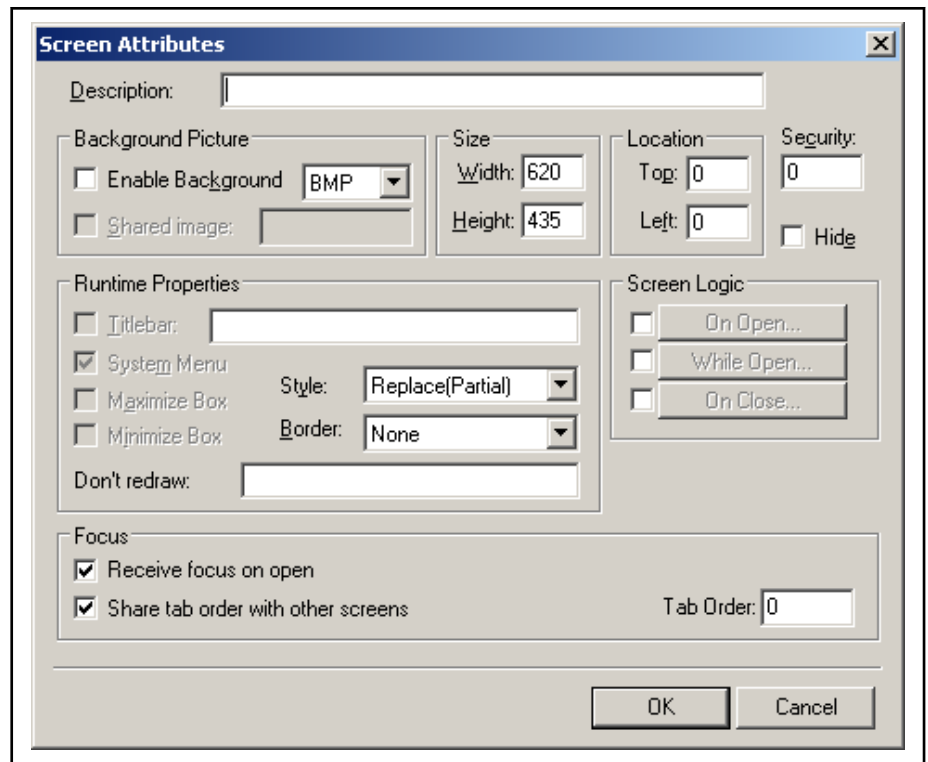


Fig. 5-82: Attributes of a WinStudio screen

Dynamic object properties

Some properties (color, size, position) of WinStudio objects that can be inserted in a WinStudio screen can be changed dynamically. Moreover, functions can be configured that can be executed when enabling WinStudio objects.

To add a dynamic attribute to a WinStudio object, select the WinStudio object (e.g. a rectangle) and toolbar activate the desired attribute in the "WinStudio - Dynamic Functions". The menu function **WinStudio Screen ► Dynamic Attributes** can also be used.

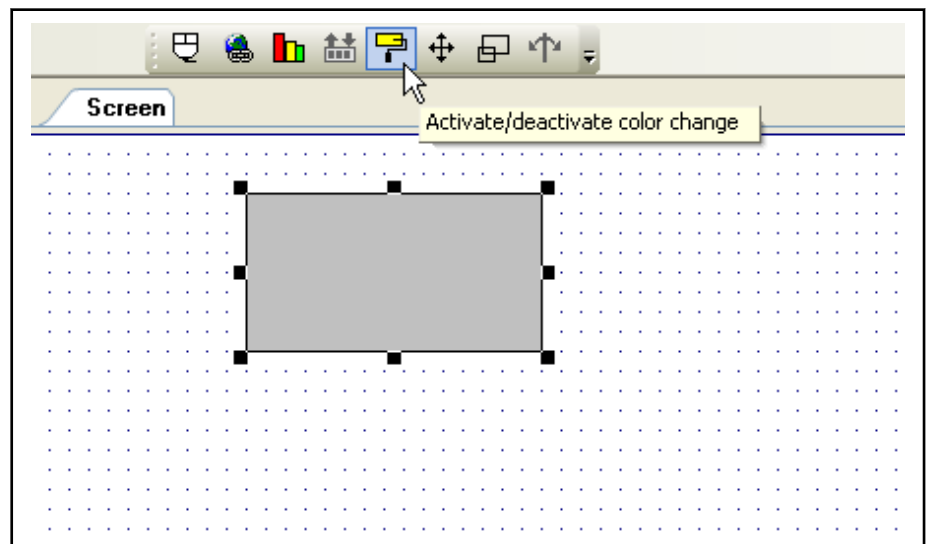


Fig. 5-83: Adding dynamic WinStudio object properties

Several dynamic properties can be assigned to a WinStudio object by enabling them individually.

Configuring the user interface

The following dynamic properties are available:

- **Command**
Function that is to be executed while enabling the element
- **Hyperlink**
opens when enabling the hyperlink
- **Bar diagram**
represents a variable value as bar diagram
- **Text input/output**
shows the value of a WS variable or applies the entered value
- **Color change**
changes the color depending on a WS variable
- **Position**
changes the X and Y position of an object depending on a WS variable
- **Size**
changes width/height of the object depending on a WS variable
- **Line rotation**
rotates a line depending on a WinStudio variable

Object properties and variable selection

The properties of a WinStudio object can be edited by means of the property dialog. Select the object and double click on it. In the properties dialog, the basic properties (e.g. line width) can be edited, where the scope of properties provided depends on the respective object type.

Some properties can be linked to a WinStudio variable (see also "[Dynamic object properties](#)" on page 121). Enter the name of a WinStudio variable or a valid WinStudio expression in the "Limit/Expr." field of the "Object Settings" dialog.

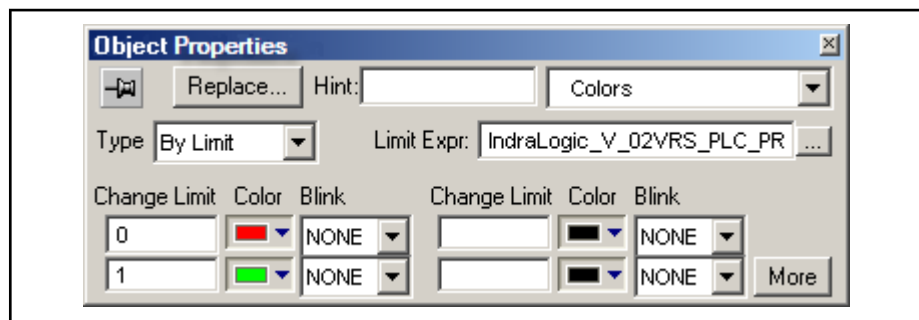


Fig. 5-84: Object properties

The "Variables" browser can also be used by clicking on the ... button.

Variables of the following areas can be selected in the selection dialog:

- Application
- Divided
- System

Triggering functions using the F-key or M-key

If a function in a user screen should be triggered with an F-key or an M-key at runtime of IndraWorks Operation, assign a control element (e.g. a button) to a command. This is achieved by adding the dynamic property "Command". Enter the desired actions in the "Tag / Expression" table. Additionally, a con-

Configuring the user interface

figuration dialog can be opened via the **Config...** button by selecting and configuring a key.

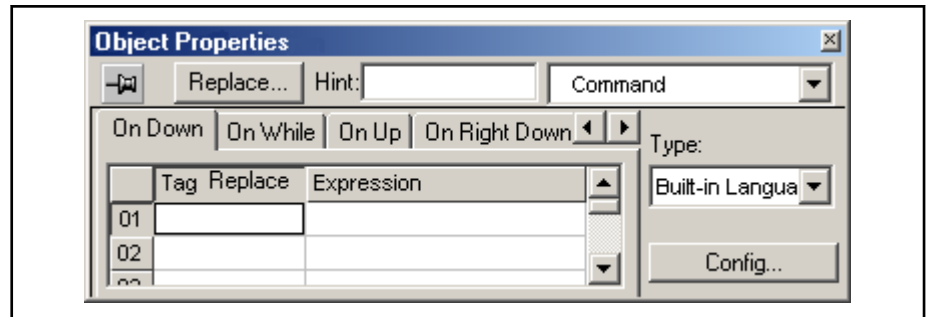


Fig. 5-85: Configuring keyboard code

If the shortcut is pressed, the configured commands are executed at runtime.

Alternatively, the pressed key can also be simulated via an F-key or an M-key. To do so, select the "Activate Keyboard Code" function for the key in question in the IndraWorks Engineering F- or M-panel editor and enter the key desired.

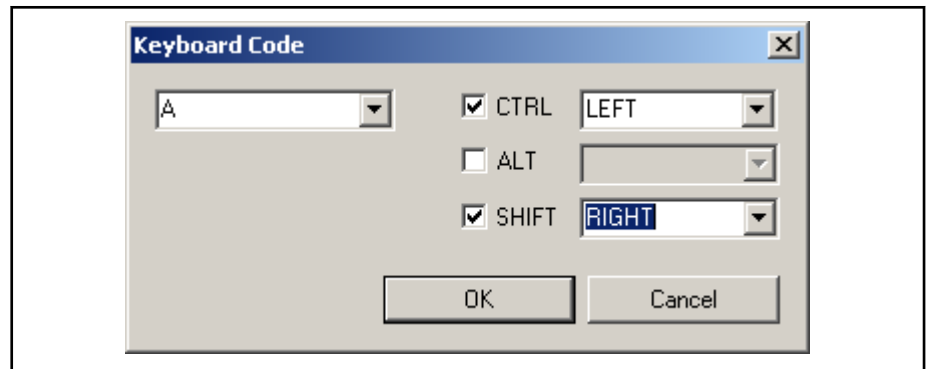


Fig. 5-86: Triggering keyboard code via F-key

5.11.4 WinStudio variables

WinStudio offers the possibility to execute actions (e.g. calculations) depending on a variable, to assign results of calculations to variables, and to link WinStudio variables to process variables.

The following categories of variables can be found in the "Variables" folder:

- **Internal variables**
Predefined, unchangeable variables
- **Distributed variables**
Variables linked to a process variables
- **User variables**
individual, user-specific variables

Classes WinStudio provided the following data types to individual variables:

- Integer
- Bool
- Real
- String

Moreover, individual data types, so-called "classes" can be defined that are composed of standard types.

Configuring the user interface

To add a new variable class, execute the function **WinStudio ► Variables ► Classes ► New Class...**. A new element is added to the "Classes" folder once a class name has been entered. This can be edited in the class editor by double-clicking on the new element (see [fig. 5-87 "WinStudio class editor"](#) on page 124).

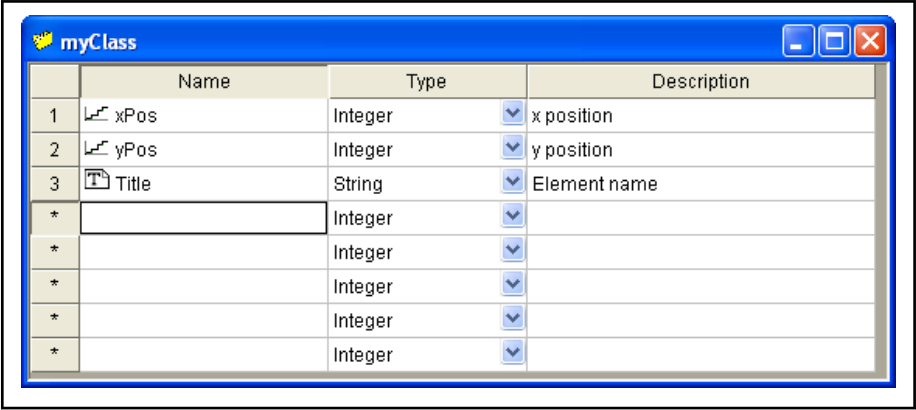


Fig. 5-87: WinStudio class editor

In the class editor, individual element names can be added to the "Name" column and a data type can be assigned under "Type". After the new data class has been saved, they can be used for WinStudio application variables.

Communication

WinStudio variables can be linked to PLC variables to show the value of the PLC variable in a WinStudio screen, to use it for calculations or to change it. WinStudio variables are linked to PLC variables via so-called driver sheets. A driver sheet contains different settings, apart from the variable list, that are required for the communication with the PLC. OPC as well as BRWS driver are available for the communication.

Adding driver screens

To link a WinStudio variable to a PLC variable, add a new driver screen via **WinStudio ► Variables ► Communication ► OPC ► New OPC Driver Screen** or **WinStudio ► Variables ► Communication ► BRWS ► New BRWS Driver Screen**. The driver sheet designation consists of a unique number and an optional description text. This number may be changed with the context menu function "Rename". The "optional description text" may be edited in the "Description field" in the driver sheet.

5.11.5 Linked symbols
WinStudio library

WinStudio provides a comprehensive library of symbols that can be inserted in WinStudio screens. The WinStudio library can be opened via the **WinStudio ► Tools ► WinStudio Library** menu item.

The WinStudio library is represented in a tool window in the lower section of IndraWorks Engineering (see next figure). The library consists of a number of folders (①) in which the WinStudio symbols are sorted. Clicking on a folder displays all of the symbols it contains in a preview window (②).

Insert symbol

To insert a symbol in a WinStudio screen, open the WinStudio screen. Select a symbol by clicking on it in the preview window (②). Add the symbol into the desired position by clicking into the WinStudio screen editor (③).

Configuring the user interface

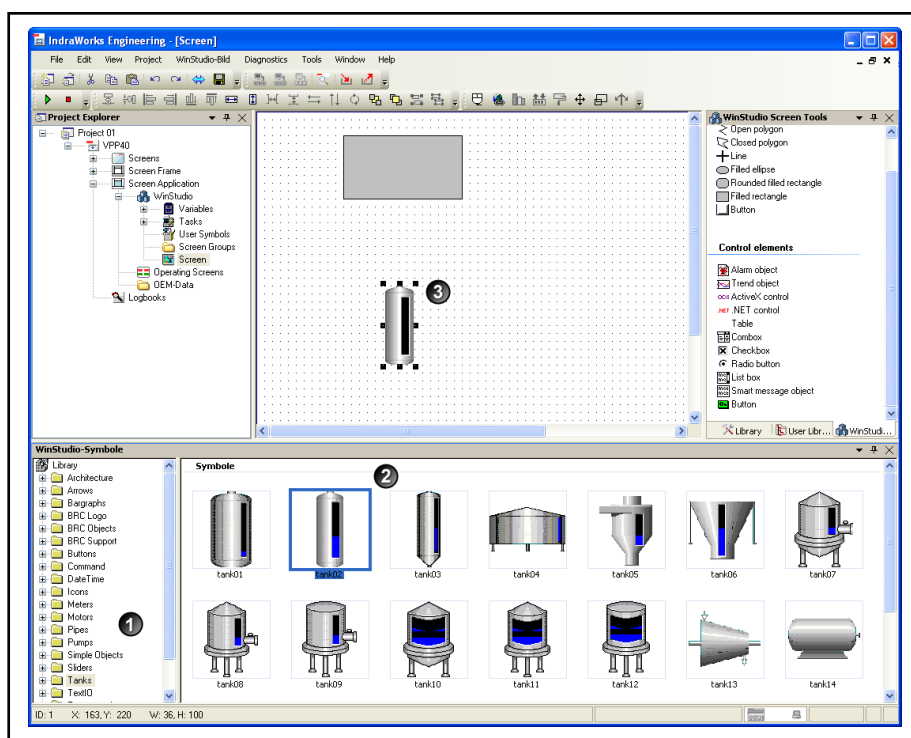


Fig. 5-88: WinStudio symbol library

Parameterize symbol

Each WinStudio symbol has several parameters that have to be adjusted correspondingly. The parameterization dialog opens by double-clicking on the symbol in the WinStudio screen. All symbol parameters are sorted by category and are displayed by clicking on the category. Individual parameters can be changed by clicking on the "Value" column and entering a value. A value or a WinStudio variable can be entered in this column.

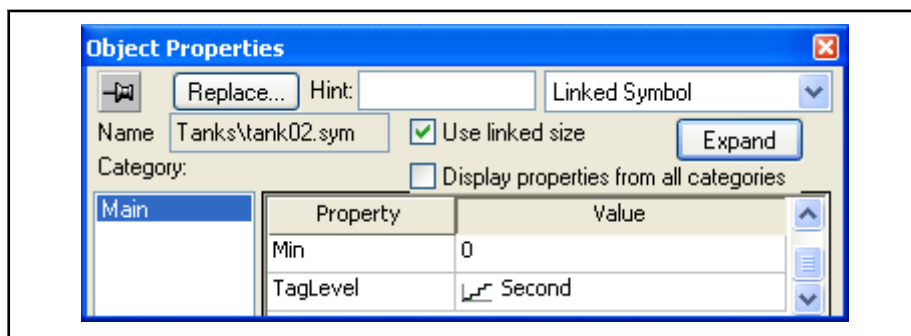


Fig. 5-89: Parameterization dialog of a WinStudio symbol

User symbols

Creating a symbol

Customized symbols can be created in addition to predefined WinStudio symbols. If an object is to be used for different purposes in a WinStudio project, creating a new symbol can be really helpful. If a symbol is changed, this is applied to all WinStudio screens in which the symbol is used. The symbol does not have to be inserted in the WinStudio screens again.

To create a symbol, firstly insert one or more objects into a WinStudio screen, highlight them and click on the function **Group** in the context menu. Then click on the function **Create Linked Symbol** in the context menu and enter a name in the file browser.

Configuring the user interface

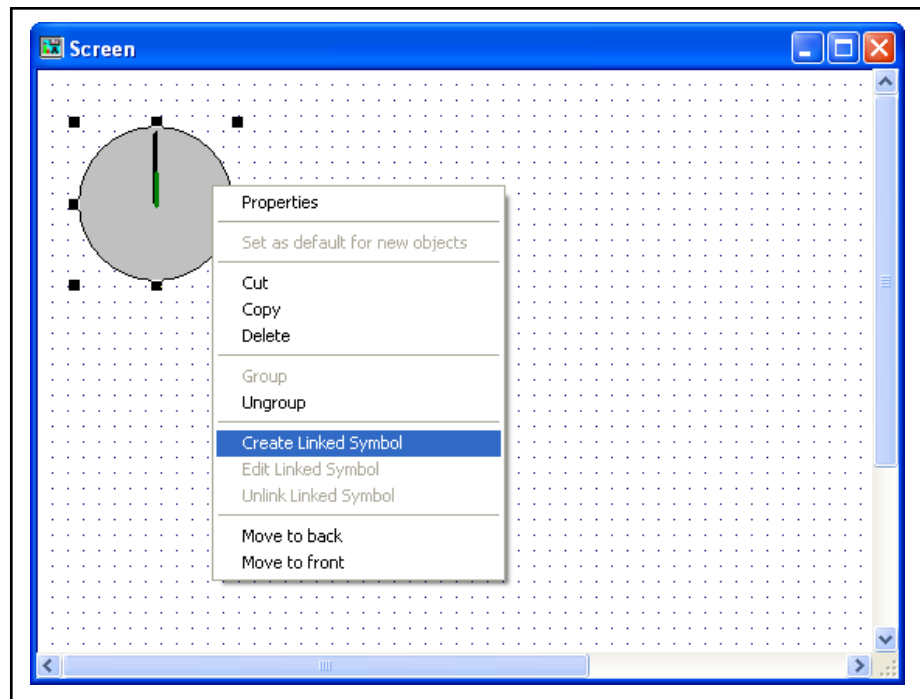


Fig. 5-90: Creating a user symbol

The new symbol appears in the "User Symbols" folder. Edit the symbol at any time by double-clicking on this element.

In a user symbol, a parameter name can be entered at any location where a WinStudio variable can be entered. Using the parameter name, the symbol can be used several times and each time a corresponding variable or a fixed value can be assigned to the parameter.

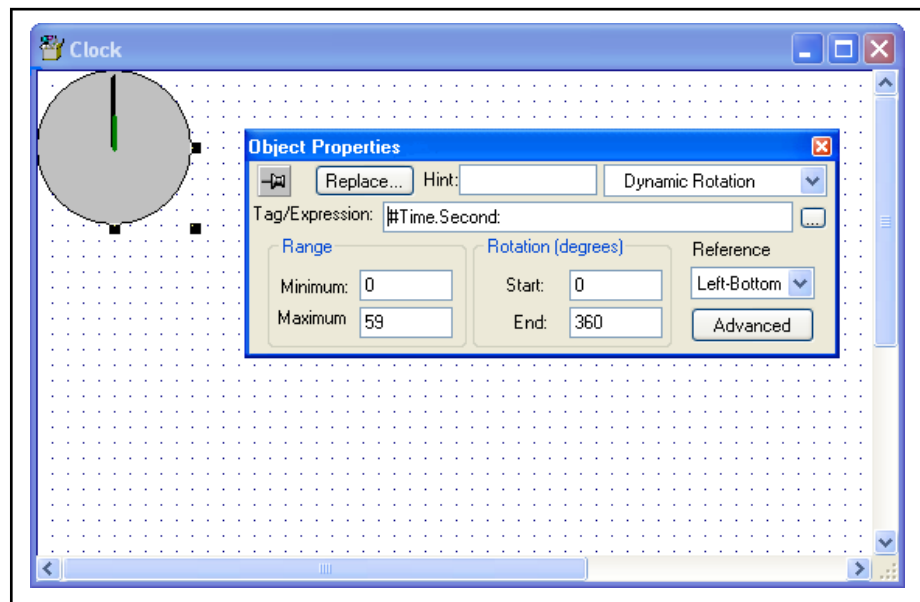


Fig. 5-91: Adding attributes to a user symbol

Use the following syntax in the input field to render a variable or an expression of a symbol parameterizable:

#[Category.]Property: [Value]

Configuring the user interface

The description of the category and the specification of a default value is optional. The category names and property names entered are displayed in the "Properties" dialog once the symbol has been inserted into a WinStudio screen.

Apply symbol to library

A created WinStudio symbol is initially only available in the same project. To use it in other projects, transfer it to the WinStudio library by clicking on the element in the project tree under **WinStudio ► User Symbols** and execute the function **Apply to WinStudio Library**. The symbol is copied to the "User Symbols" folder of the WinStudio library.

Password-protect symbol

You can assign a password to customized user symbols so that only authorized users can edit the symbols. A protected user symbol can thus be used and parameterized in WinStudio screens but the internal functionality cannot be edited.

To protect a user symbol with a password, open the editor. Open the input dialog in which any password can be entered via the **WinStudio user symbol ► Change password....** After the entry has been confirmed and the symbol has been saved, the password protection is active.

Upon opening a protected user symbol, an input dialog is displayed, prompting the user for a password. The editor only opens when the correct password is entered. To remove the password protection again, open the password dialog again via **WinStudio user symbol ► Change Password...** and leave the input field for the password empty.

5.11.6 WinStudio tasks

Mathematic sheets

A WinStudio mathematics sheet is used to program sequences, variable assignments and the execution of other WinStudio functions. To add a mathematics sheet, click on the function **WinStudio ► Tasks ► Add... ► Mathematics Sheet**.

	Tag Name	Expression
1	counter	counter + 1
*		
*		
*		
*		

Fig. 5-92: WinStudio mathematics sheet

The mathematics sheet is executed cyclically if the variable or the expression in the "Execution" field is TRUE.

Alarms

By means of alarms, messages can be defined that are output if a variable exceeds or falls below a value.

Configuring the user interface

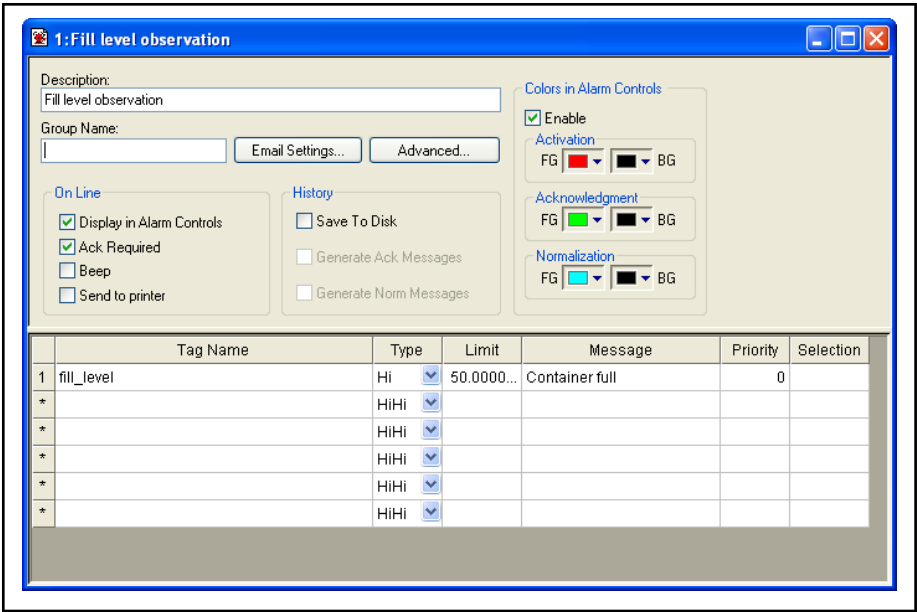


Fig. 5-93: WinStudio alarm editor

To add an alarm, click on the function **WinStudio ► Tasks ► Add... ► Alarm**. Alarm messages can be displayed by means of the alarm control element in WinStudio screens.

Recipes

In WinStudio, variable values can be saved in files. Variable values can also be loaded from files by means of so-called recipes. A WinStudio recipe is a list of variables and the assignment of variables.

To add a recipe, click on the function **WinStudio ► Tasks ► Add... ► Recipe**.

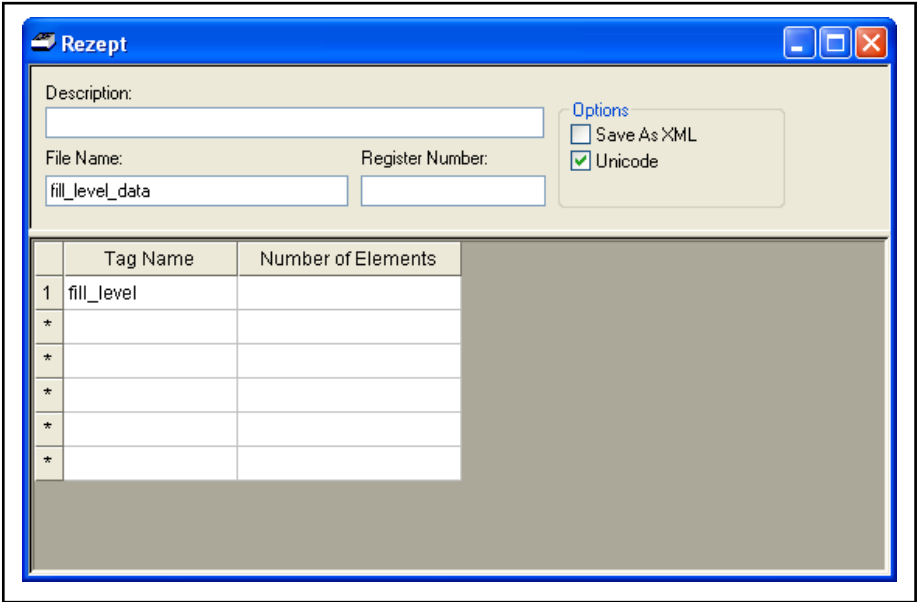


Fig. 5-94: WinStudio recipe editor

In the "Tag Name" column of the recipe editor, the variable names can be entered whose values are to be saved or loaded. If the variable is an array, the element number has to be entered in the "Number of elements" column.

Configuring the user interface

Loading and saving recipes

To save variables in a recipe, insert the following call at a suitable position (e.g. as interface command or in a mathematics sheet): `Recipe("Save:<RecipeName>")`. If the function is executed, the current variable values listed in the recipe are written to a file. Values that have been saved in a recipe can also be loaded again by using the `Recipe("Load:<RecipeName>")` command. In this case, the current variable values are overwritten with the recipe values.

Reports

A WinStudio report is used to create a documentation that can contain text as well as current WinStudio variable values. The report can be saved in a file or printed out.

To add a report, click on the function **WinStudio ► Tasks ► Add... ► Report**.

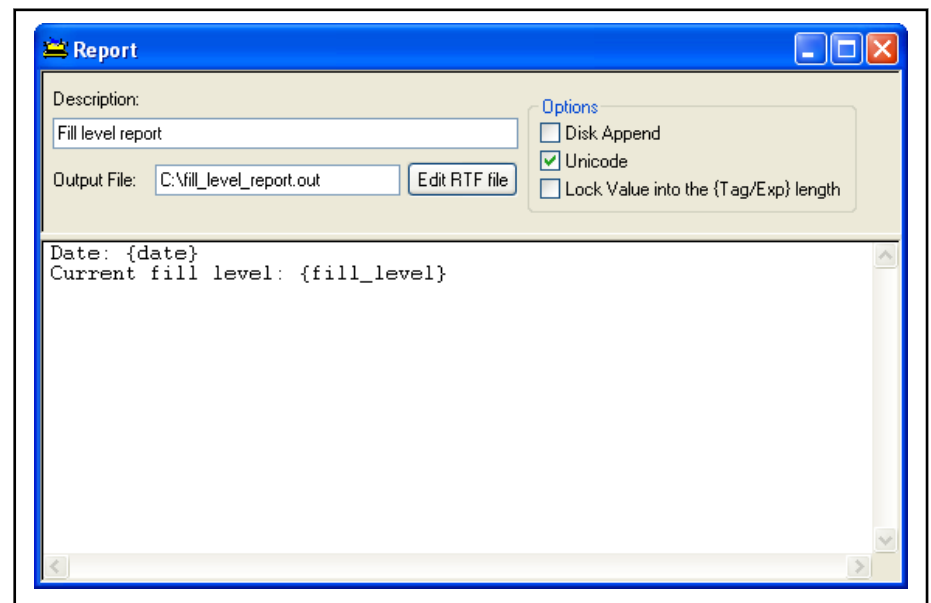


Fig. 5-95: WinStudio report editor

In the field "Output File", the report file name can be specified. In the lower editor area, text can be entered that is to be written in the report. Variable values can also be output with the help of the following syntax: `{<Variable-name>}`.

Output report

A report is generated at runtime by IndraWorks Operation by executing the `Report("<Reporttype>:<Reportname>")` command.

Parameters of the report command:

- **Report type**
"Disc" to output it in a file or "Prn" for printing the report on a standard printer
- **Report name**
Name of report in "Reports" folder

Trends

WinStudio trend data serve to record WinStudio variables. These variables can be represented in a WinStudio screen by means of a trend object to monitor the timing sequence of variables.

Create a trend data object to configure trend data. To do so, execute the function **WinStudio ► Tasks ► Add... ► Trend**.

Configuring the user interface

For details about the configuration of trend data objects and the visualization of trend data, refer to the WinStudio online help.

Sequences

Actions can be executed cyclically by means of WinStudio sequences. For example, it is possible to issue a report automatically every day (cf. [chapter 5.11.6 "WinStudio tasks" on page 127](#)) or to reinitialize a variable every 30 seconds.

To add a sequence, execute the function **WinStudio ► Tasks ► Add... ► Sequence**.

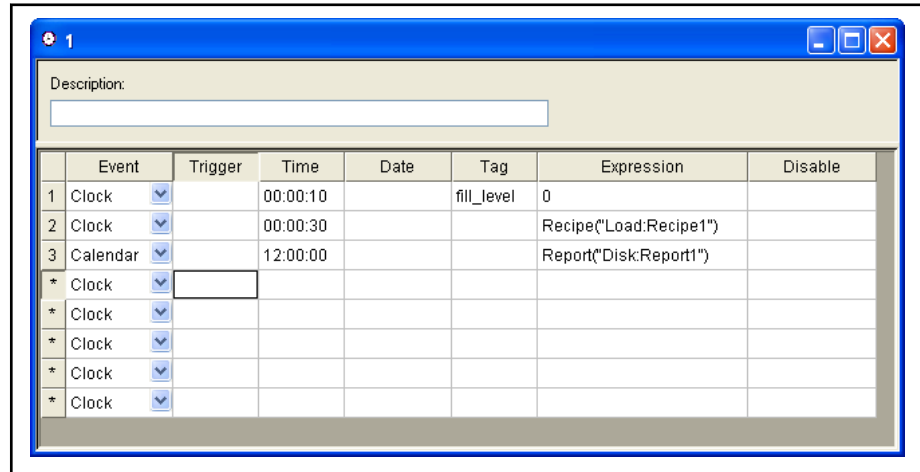


Fig. 5-96: WinStudio sequence editor

One event and one action can be defined in the sequence editor per line.

Enter the interval type of every event in the "Event" column. The following options exist:

- **Clock**
The event occurs at an interval between 1 second and 24 hours
- **Calendar**
The event occurs at an interval of > 24 hours
- **Change**
The event occurs when the variables in the "Trigger" field change

The interval variable can be specified by means of the "Time" and "Date" column:

- **Time**
Interval variable in hh:mm:ss for "Clock/time" event; time at event "Date"
- **Date**
Event date. If the field is blank, the event occurs on a daily basis

Each event can be assigned an action by means of the "Variable" and "Expression" columns. A WinStudio function or a value can be entered in the "Expression" column that is to be assigned to a variable.

ODBC

By means of ODBC driver sheets (Open Database Connectivity), WinStudio variables can be linked to external databases. Alarms, trends and events support the connection to an external database by means of ODBC.

To add an ODBC driver screen, execute the function **WinStudio ► Tasks ► Add... ► ODBC**.

Configuring the user interface

A detailed description of the ODBC configuration can be found in the WinStudio online help.

Scripts

Overview

WinStudio supports the creation and execution of user-defined functions in VisualBasic scripts. The VisualBasic script language is a simple and flexible tool to program functions and sequences. An introduction to VB (VisualBasic) and a language reference can be found in the MSDN (Microsoft Developers Network).

Global procedures

Some VB function can be declared as "Global Procedures" and a function library can be created. Functions that are declared under "Global Procedures" can be called in all WinStudio objects with VB support (e.g. as command on interfaces).

Open the editor for global procedures via double-click on **WinStudio ► Tasks ► Scripts ► Global Procedures**.

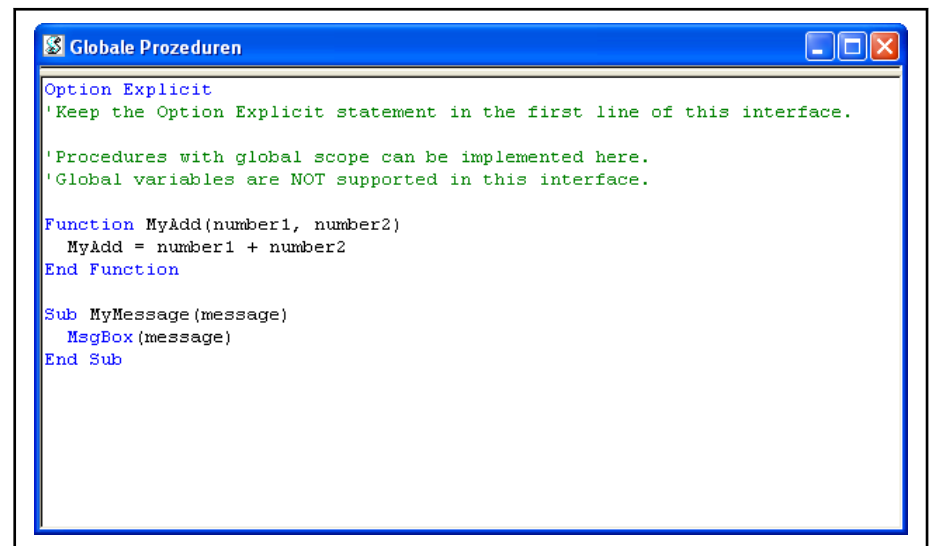


Fig. 5-97: Global procedure editor

A user-defined number of functions and subfunctions can be declared in the editor for global procedures.

Start script

In the "Start script" element, VB function and user-defined global procedures can be called that are to be executed once upon the start of the application. The start script can particularly be used to initialize variables.

To edit the start script, open the VB editor by double clicking on **WinStudio ► Tasks ► Scripts ► Start Script**.

Graphic script

The VB script "Graphic script" can be used for one-time execution of VB statements upon starting or closing the WinStudio graphics module or for continuous execution during the runtime of the graphic module. The graphic script contains the necessary subfunctions in which an individual VB code can be inserted. To correctly execute functions in the graphic script, the function names cannot be changed. More function can be added that can be called in the predefined graphic functions.

Configuring the user interface

To edit the graphic script, open the VB editor by double clicking on **WinStudio ► Tasks ► Scripts ► Graphic Script**.

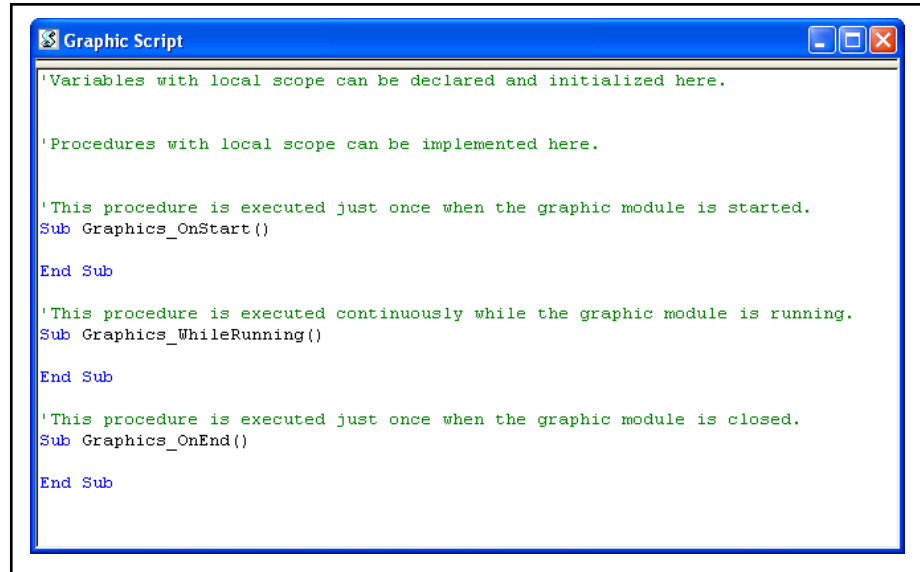


Fig. 5-98: WinStudio graphic script editor

Screen scripts

VB statements can be programmed for each WinStudio screen. The statements can be continuously executed upon the opening or closing of the screen while the WinStudio screen is still opened. The screen scripts already contain all required subfunctions that can be inserted in the user-defined VB code. The names of the predefined functions cannot be changed.

There is a VB script for every WinStudio screen, which can be opened and edited in the context menu of the open screen editor using **Screen Script**.

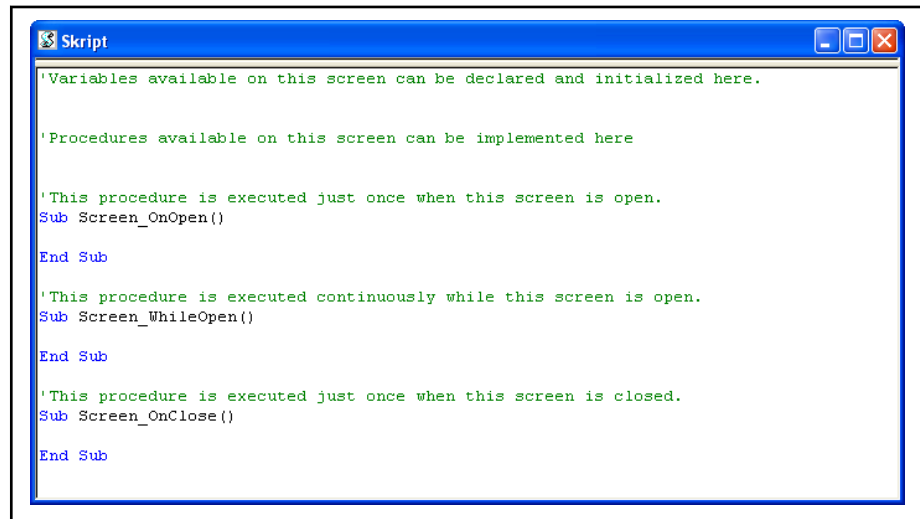


Fig. 5-99: WinStudio screen script editor

User scripts

In addition to the predefined VB scripts ("start scripts", "graphic script" and "screen script"), any number of VB scripts can be added in which VB scripts can be inserted. Functions cannot be declared in user scripts.

To add a user script, execute the function **WinStudio ► Tasks ► Scripts ► New Script**.

User scripts contain a continuous number that specifies the execution sequence of the user scripts. To exclude a script from the execution at runtime, enter a WinStudio variable or expression in the "Execution" field. In this case, the script is only executed if the variable or the expression has a value greater than zero.

5.11.7 WinStudio tools

Verify application

Variables can be used in all WinStudio elements that are required for the respective function. During the configuration, inconsistencies can occur by deleting or renaming variables so that the affected WinStudio functions and element do not function correctly anymore.

Using the function **WinStudio ► Tools ► Verify Application** it is possible to check whether variable names are used in WinStudio elements (mathematics screens, WinStudio screens, scripts etc.), which do not exist in the variable database. When the function is selected, a dialog opens in which the verification can be started by clicking on the **Verify** button. Close the dialog after the verification has been completed.

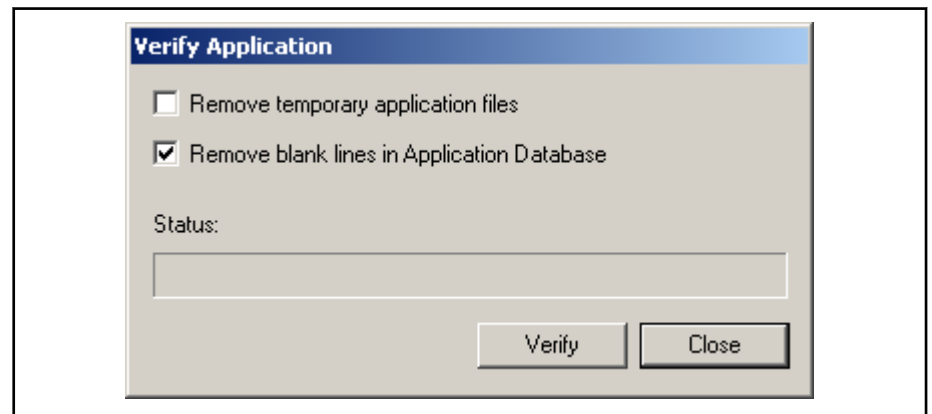


Fig. 5-100: Verify application

Warnings and error messages are output in the output window of IndraWorks Engineering after the verification dialog is closed.

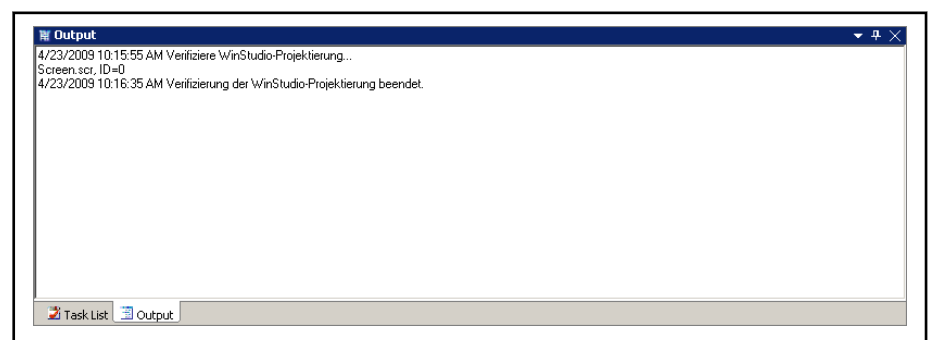


Fig. 5-101: Output of messages of the function "Verify application"

If errors occur during the verification, they are output in the task list of IndraWorks Engineering. The error source is entered in the "Origin" column. The object ID number is displayed in the "Position" column. By double-clicking on the error line, the editor opens and the incorrect element is automatically selected. That way, all inconsistencies can successively be corrected and the function "Verify Application" can be restarted if required.

Configuring the user interface

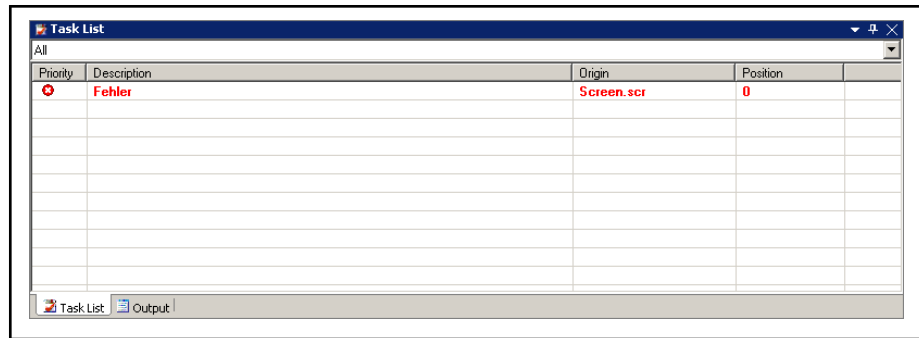


Fig. 5-102: Error list of the "Verify Application" function

Start application

The function of individual WinStudio screens can be checked using the function "Start application" before enabling the entire package in IndraWorks Operation. The test window can be opened using **WinStudio ► Start Application**.

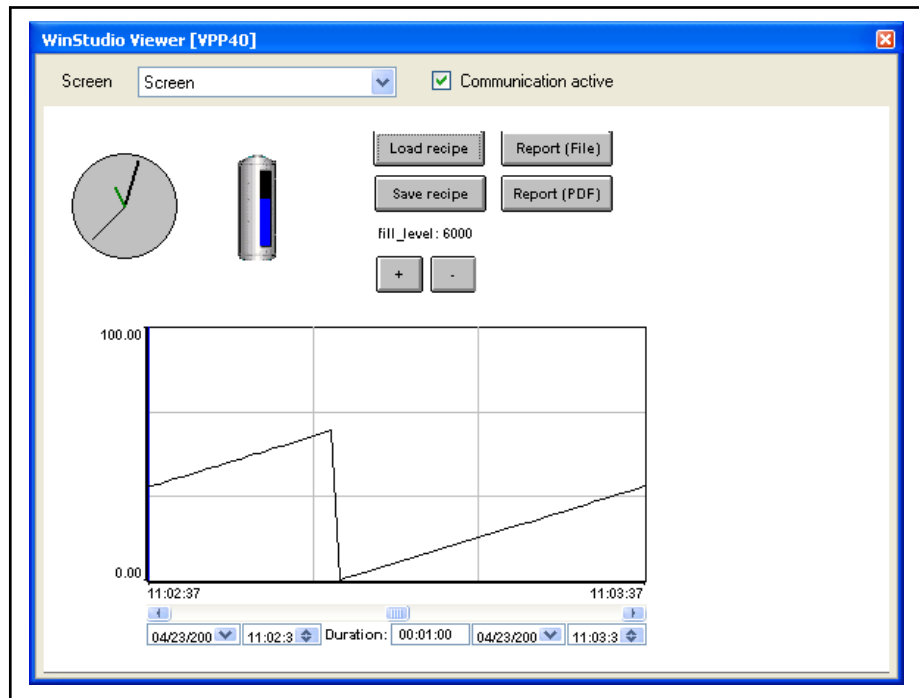


Fig. 5-103: Testing WinStudio screens

All screens are contained in the current WinStudio application in the "Screen" selection list. When selecting a screen, it is displayed in runtime mode and the variable display, commands on interfaces or the execution of scripts can be tested.

If WinStudio screens of another visualization device from the same IndraWorks project are to be tested, close the application window and then reopen it for the respective visualization device using **WinStudio ► Start Application**. The application window can be closed using **WinStudio ► Close Application**.

5.11.8 Database spy

By means of the WinStudio database spy, the value and the status of individual WinStudio variables of an active WinStudio application can be monitored. Execute the function **WinStudio ► Tools ► Database Spy** to open the Database Spy. In the input field of the database spy, the WinStudio variable name

Configuring the user interface

can be entered and applied to the list of variables to be monitored by clicking on **Add**. The current value as well as the connection status is displayed for each variable in the variable list. Click on **Delete** to remove all variables from the Database Spy.

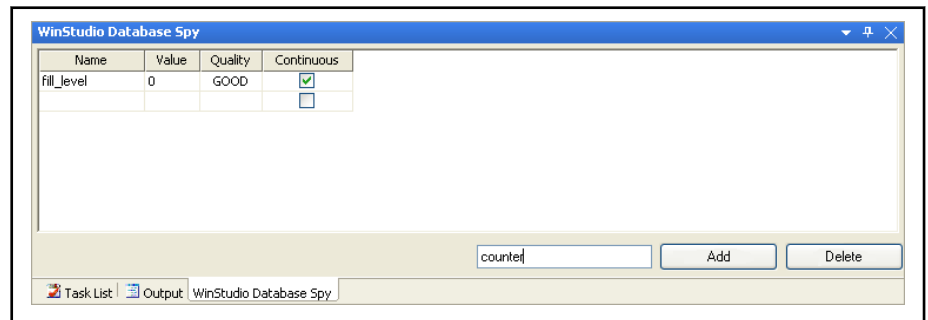


Fig. 5-104: WinStudio database spy

5.11.9 LogWin

WinStudio LogWinTo can be used to analyze WinStudio LogWin application sequences and to search for errors. In LogWin, OPC communication messages, individual trace messages and the recipe execution can be protocolled, for example. The LogWin Tools window can be opened using the menu item **WinStudio ► Tools ► LogWin**.

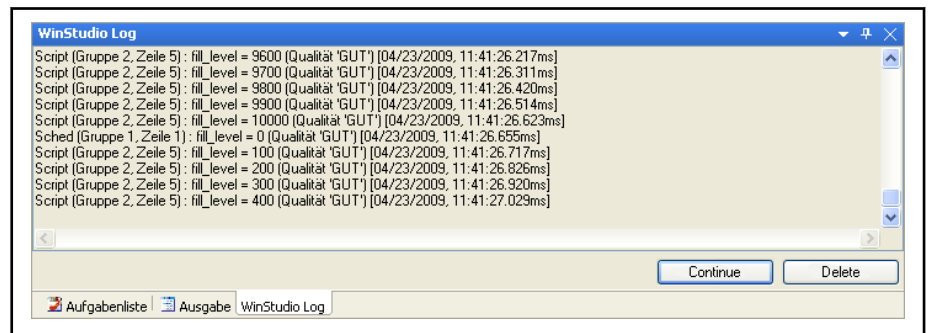


Fig. 5-105: WinStudio message output in WinStudio LogWin

In the context menu of the LogWin window, a configuration dialog to select messages to be recorded can be opened via **Settings**. It is also possible to select WinStudio variables (see [fig. 5-106 "Monitoring WinStudio variables in LogWin" on page 135](#)) for which a message with the current value is to be issued if they are changed.

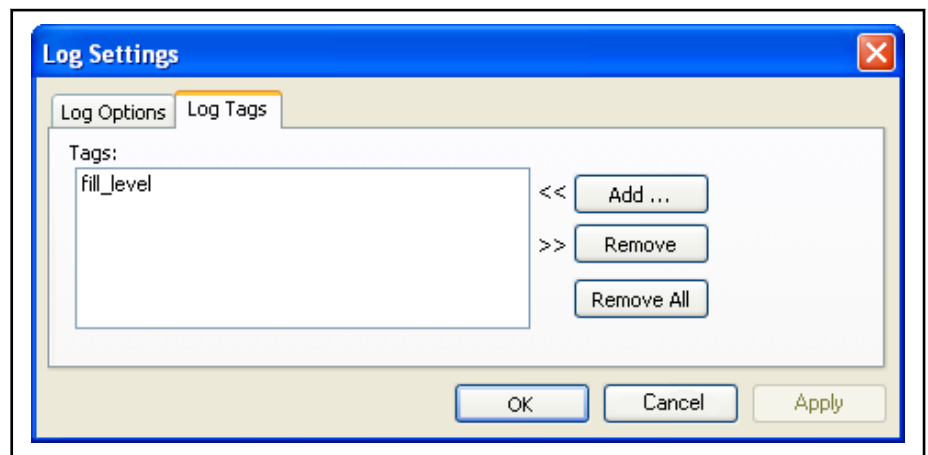


Fig. 5-106: Monitoring WinStudio variables in LogWin

Configuring the user interface

5.11.10 Cross reference search

Use the cross reference search to find certain WinStudio variables. To start the cross reference search, select **WinStudio ► Tools ► Cross References**. The displayed tool window opens.

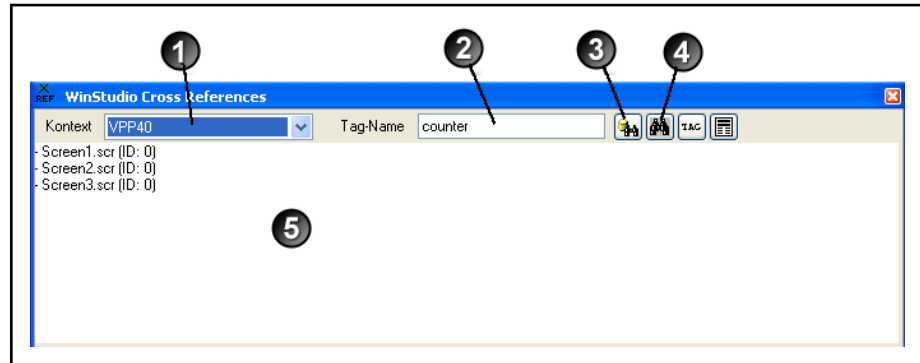


Fig. 5-107: Cross reference search of WinStudio variables

If the IndraWorks project contains multiple visualization devices, select the search context in the drop-down list ①, since it can only be searched for one visualization device at a time. Then, enter the name of the WinStudio variable to be searched for into the input field ②. Alternatively, use the button ③ to open the WinStudio variable browser and select a variable from the existing variables. Click on the button ④ to start the search; the elements found are displayed in the hit list ⑤. Double-click on a hit to open the displayed element in the respective editor. The button ④ is only enabled if a valid WinStudio variable was entered in the input field ②.

The cross reference window can be closed at any time via the **X** button or the IndraWorks menu **View ► Other Windows ► WinStudio Cross References** and can be opened again via the context menu function **WinStudio ► Tools ► Cross references**.

5.11.11 Variable import wizard

Often, it is practical to apply variable lists from another WinStudio application or a database to avoid that they have to be configured again manually. The transfer of variables is supported by the import wizard that is opened via **WinStudio ► Tools ► Variable Import Wizard**.

To import variables from another WinStudio application, select the entry "WinStudio Database" on the first screen of the import wizard under "Source" and click on **Next**.

Configuring the user interface

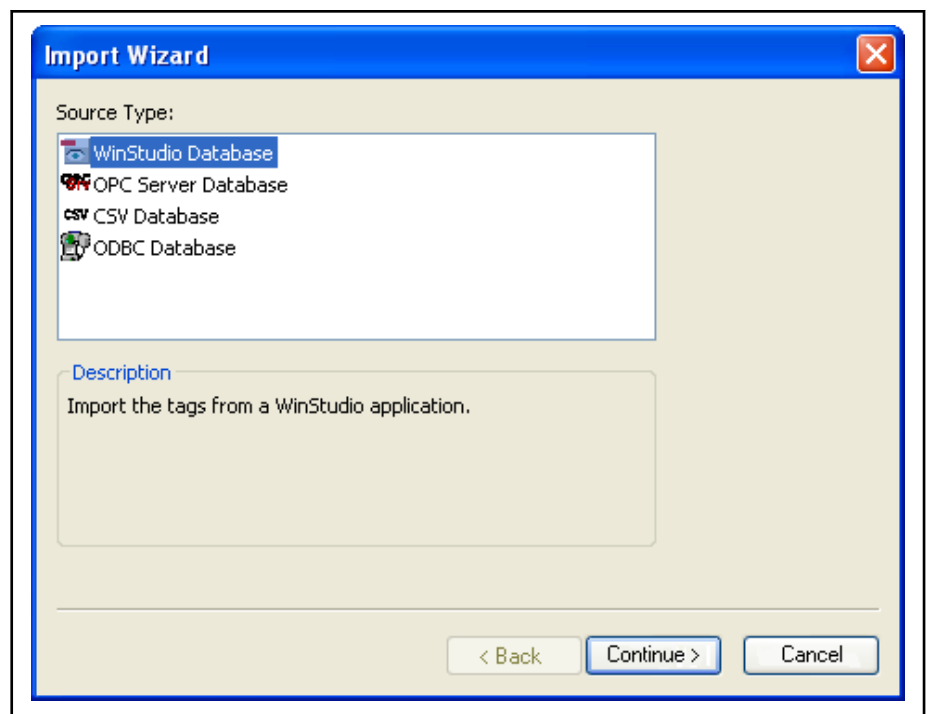


Fig. 5-108: Selecting a data source in the WinStudio import wizard

On the second screen, select the WinStudio project in the "Application" field.

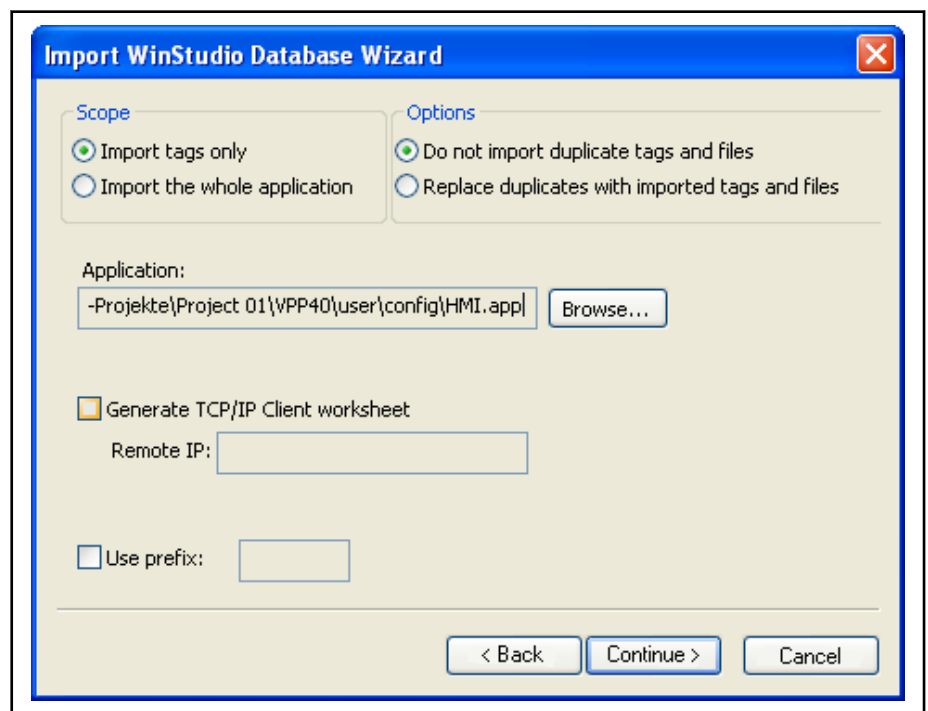


Fig. 5-109: Selecting a WinStudio application

The available variables are listed on the next screen of the import wizard. Select the variables required and complete the import with **Finish**.

Configuring the user interface

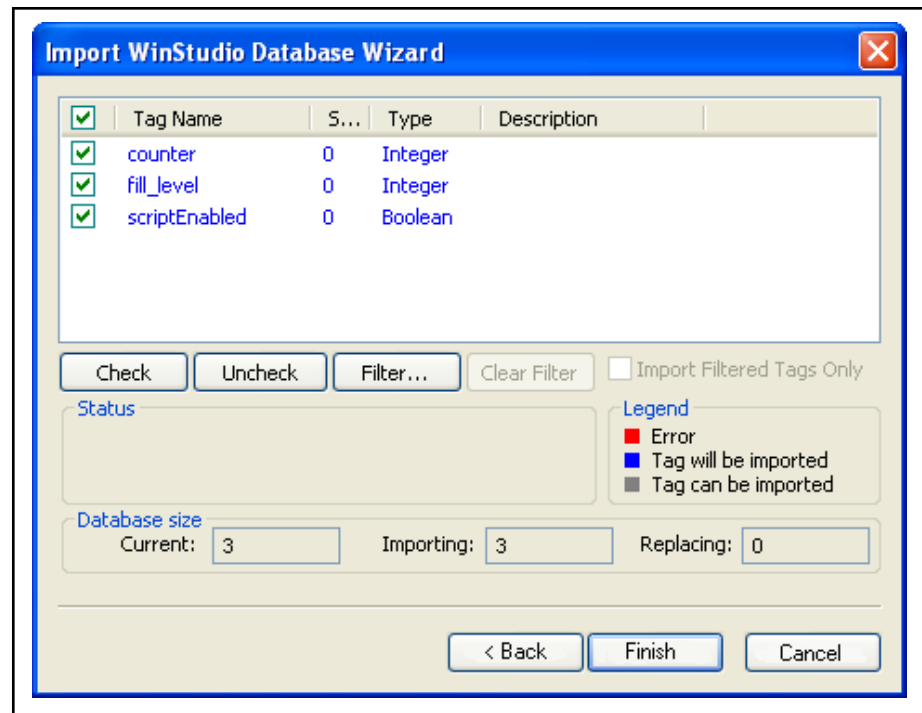


Fig. 5-110: Selecting the variables to be imported

5.11.12 Export/import

The "WinStudio" folder as well as all corresponding subelements support the Export/Import functionality in IndraWorks. By means of this functionality, the entire WinStudio project or only parts of can be exported and imported in another IndraWorks project.

Please note the following characteristics when using the Export/Import function on WinStudio elements:

- For each WinStudio element, the WinStudio variables used are determined and included in the export. When importing this WinStudio element, the required WinStudio variables are inserted again. If the variable to be imported already exists in the new project, all corresponding variable properties (e.g. description text, initial value) are overwritten. If the variable type differs, the variable remains unchanged. If you still want to import the variables, change the variable type or rename the variable and repeat the import.

If an existing variable has been overwritten during the import or if the new variable could not be imported, the import dialog displays a corresponding message.

- If the WinStudio variable to be exported is a user-defined WinStudio data class, the corresponding class definition also has to be exported so that the variable can be created correctly. Therefore, a query is displayed during the export, asking the user if he wants to import the required class if the class has not been selected yet. It is recommended to confirm the query with "Yes"

5.11.13 WinStudio licensing

Introduction

The graduated licensing model allows an optimum adjustment to the required number of variables and the necessary scope of functions. In this process, the larger licensing package covers the scope of functions of the smaller

package. An upgrade to the next licensing version is possible without a problem.

Overview on WinStudio licenses

The following licenses are available:

- **WinStudio Lite**
This license is part of the IndraWorks installation and allows the creation and execution of WinStudio projects with a maximum of 500 variables (tags). Furthermore, all CE devices are equipped with a Lite license for the execution of Lite runtime projects. In case of this license, alarms, recipes, reports, trends as well as TCP/IP driver screens and DDE driver screens are not executed and can neither be created nor edited
- **WinStudio Runtime 1K5 / 4K / 16k / 64K / 512K (RUN licenses)**
For the execution of WinStudio Runtime projects, licenses with a maximum of 1,500, 4,000, 16,000, 64,000 and 512,000 WinStudio variables (tags) are available.
- **WinStudio Engineering and Runtime 1K5 / 4K / 16k / 64K / 512K (RUD licenses)**
This license allows the creation and execution of WinStudio projects with a maximum of 1,500, 4,000, 16,000, 64,000 or 512,000 WinStudio variables (tags).
- **WinStudio Demo license**
This license allows the testing of the entire scope of functions of WinStudio for a period of 30 days. After the expiration of the test period, the last valid license (e.g. Lite license) is active
The demo license may only be activated once.



For the activation of an IndraControl VEx device (Windows CE), only 1K5 and 4K runtime licenses are available.



For the activation of an IndraControl VEx device (Windows XP Embedded/Windows 7 compact), only 1K5, 4K and 16k runtime licenses are available.

Enabling a WinStudio license

The individual "WinStudio Runtime" and "WinStudio Engineering and Runtime" licenses are subject to a charge and must be activated by entering a licensing key. Find your personal license key on the licensing screen.

To activate a license, open the licensing dialog in IndraWorks Engineering via **Tools ▶ Options ▶ General ▶ Software Licenses** and select the entry "WinStudio". Use the **Licensing...** button to open the activation dialog and enter the activation key.

After activation of the license, it may be necessary to restart WinStudio Engineering or IndraWorks Operation.

Saving the license on a dongle

The activation key is generally only valid for one PC. In order to use one license on several PCs one after the other, you have the option of transferring the license to a dongle. The license is valid as long as the dongle is plugged into the USB port of the PC. Windows CE licenses (VEx) cannot be transferred to a dongle.

Activation of licenses on VEx devices

Configuring the user interface

To activate the WinStudio license on a VEx device using the Windows CE operating system, open the licensing dialog on the VEx device in the Windows Start menu via **Program Files ► Rexroth ► WinStudio ► License Management** and enter the activation key. The licensing software is preinstalled on new VEx devices. If you already have a VEx device without pre-installed licensing software, install the IndraWorks visualization software on the VEx device and then activate the WinStudio license.

To save the license settings, start IndraWorks Operation on the VEx device and save the Windows Registry.



It is no longer possible to activate the license using the WinStudio Remote Agent with the introduction of the new licensing process. Use the IndraWorks licensing dialog (see above) to activate the license.



It is no longer possible to transfer the license key using the integrated WinStudio function with the introduction of the new licensing process. The license key must be entered directly on the VEx device.



IndraWorks license components for Windows CE are already pre-installed on new devices. To update an existing VEx device with the new software, install first the runtime components on the VEx device via the context menu of the VEx device in IndraWorks Engineering via the function **Install Software**. The link might not exist. If this is the case, start the licensing dialog via `\Storage-Card\Rexroth\WinStudio\bin\IndraWorks.LicenseImport.CE.exe`. The licensing component requires ".Net CompactFramework 2.0" so that the firmware can be updated if required.

Effects of a missing or limited license

If a project is opened with the WinStudio editor and that project was created on another computer with a higher WinStudio license, the elements subject to license in WinStudio Engineering are disabled, i.e. it is not possible to create or edit these elements.

At IndraWorks Operation start, a message appears if the WinStudio license is not sufficient for the current project. In the event of an inadequate or missing licenses, no WinStudio screens will be displayed.



If the current project should still be used, enable the demo license. Use the IndraWorks licensing dialog (see above) to activate the license.

5.11.14 Multilingualism

WinStudio user screens can have multiple languages. All texts in the WinStudio screen are displayed in the selected project language during the IndraWorks Operation runtime.

Therefore, create a text file for every language needed, by running the function **WinStudio ► Tasks ► Add ► Translation File (TRA)**. A new element is created in the "Translation Files" folder. Rename this element "HMI_XX", whereby "XX" corresponds to the country code - for example an "HMI_EN" element should be created for English texts, "HMI_DE" for German texts and "HMI_FR" for French texts.

Configuring the user interface

Open the "translation editor" and enter all texts to be translated. The original text is to be entered in the first column and the translated text in the second column.



With WinStudio version 6.3 and higher, the file format of the translation file has changed (.tra-> csv). As the new format is not yet supported, new translation files have to be still saved in the old format. When saving data, the file type "All Files (*.*)" has to be selected instead of "Text Files (*.csv)".

5.11.15 WinStudio applications

Overview

Optionally, applications can be created in IndraWorks that are exclusively based on the WinStudio runtime environment. This is required if you want to use the WinStudio web functionality. Creating and enabling WinStudio applications differs from the previously described application within the IndraWorks runtime environment. These differences are explained in the following.

Creating a WinStudio application

To create a WinStudio application, insert an VPP or VEx device from the device library to an IndraWorks project. Select the "WinStudio application" type in the HMI wizard. After confirming the entries by clicking on **Complete**, the corresponding WinStudio elements are created in the project tree. Elements such as F-, M- and OP-panels, operating screens, etc. requiring an IndraWorks runtime environment are not offered for selection.

User management

The WinStudio user management can be used to protect different WinStudio functions in WinStudio applications. The WinStudio user management is configured via **WinStudio ► Tools ► Security Settings**. The safety system can be enabled in the configuration dialog. Groups and users can be created and the corresponding rights can be assigned. For more information on how to use the WinStudio user management, please refer to the WinStudio online help "Studio Technical Reference".



Users and groups that have been configured in the IndraWorks user management are not available in the WinStudio user management. If required, they have been created again in the WinStudio user management.

Web functionality

WinStudio applications can be visualized by means of a web browser at runtime. Applications have to be saved in HTML format.

To save a WinStudio screen in HTML format, open the context menu of the WinStudio screen and execute the function "Save as HTML". Optionally, the function "Save all documents as HTML" can be used to save all existing WinStudio screens in HTML format.

After the configuration and saving of all required WinStudio screens in HTML format, the web server has to be configured. Therefore, open the WinStudio project settings and select the "Web" register. For information on the configuration and use of the WinStudio web server, refer to the WinStudio online help "Studio Technical Reference".

Configuring the user interface



The functions "Save as HTML" and "Save all documents as HTML" as well as the "Web" tab in the WinStudio project settings are only available for a "1,5 k" WinStudio license or higher.

Transferring the application and activation on the target device

To transmit the WinStudio application to a target device, start the WinStudio remote agent. Subsequently, the "Execution Platform" dialog can be opened via the WinStudio context function **WinStudio Target Platform**. In this dialog, it can be selected which target station you want to connect to. In the "Local" setting, no other inputs are required. Start and/or exit the WinStudio runtime environment by clicking on the **Start** and/or **Stop** buttons in the "Application" tab.

If you select the option "Network IP" on the "Target" tab and enter a valid IP address, you can connect to the target device by means of the remote agent by clicking on **Connect**. If the connection was established successfully, the suggested target directory can be changed by means of the directory browser on the "Application" tab. The **Send to target** interface copies all files of the current WinStudio application to the selected target device directory. The **Start** and **Stop** buttons start or stop the WinStudio application on the target device.

5.11.16 Further information

WinStudio offers comprehensive functions to create user screens. The complete functional scope is described in the documentation on Rexroth WinStudio (see [chapter 1.3 "Required and supplementing documentation IndraWorks" on page 6](#)).

5.12 Configuring operating screens

To create a new operating screen, in the context menu of the "Operating screens" folder, select the menu item **New Operating Screen**. Then, configure the operating screen using the editor described below and assign it to a screen element. The screen editor can be used to select F-panels and M-panels to be displayed together with the operating screen.

Operating screen editor

The operating screen editor provides an overview on the configuration of the individual operating lines and the M-key assignment of the operating screen (see [fig. 5-111 "Operating screen editor" on page 143](#)). Select the control intended for communication with the operating screen elements from the "Control" (①) list box. Hide unnecessary operating lines by means of the checkboxes (②). In the default setting, the two top operating lines are hidden because this space is occupied by the header. To hide the header, open the screen element to which the operating screen is assigned and activate the "Header hidden" checkbox. Save the setting. The two top lines appear when the operating screen editor is opened.

The PLC flags to be used for the M-keys (③) and the status displays (④) as well as the texts to be displayed and the colors can be configured for each line. To check the configuration, set the preview to display the variables used by setting the appropriate radio button (⑤).

For the operating screen, either 5 or 7 lines may be configured (⑥).

Configuring the user interface

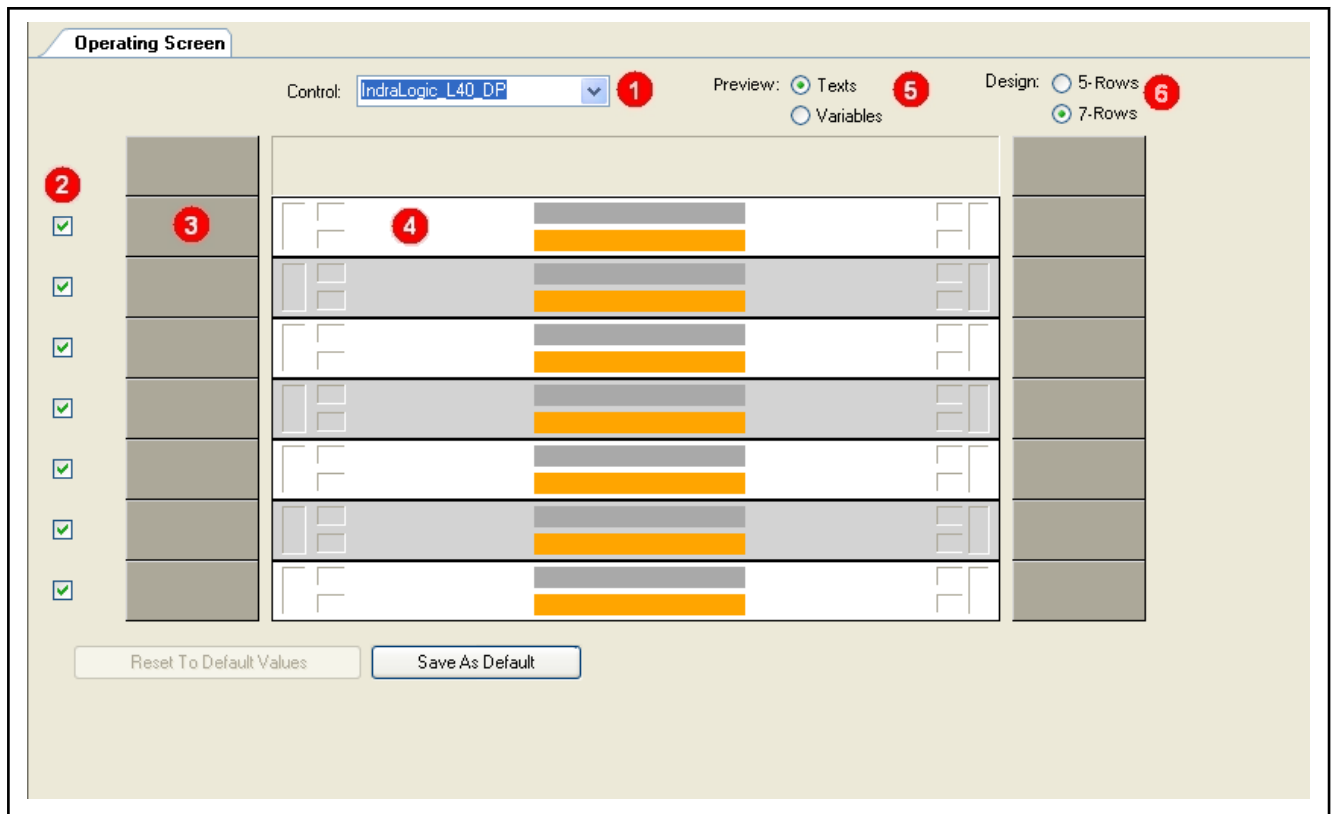


Fig. 5-111: Operating screen editor

Click on an operating line to open the configuration dialog. The dialog automatically selects the tab that allows the selected element of the operating line to be configured.

Browsing PLC variables

Drag and drop the browser elements to the respective input fields of the editor. It is also possible to edit them manually. [chapter 6 "IndraWorks process variables" on page 199.](#)

"M-key" tab

The "M-key Left" and "M-key Right" tabs can be used to define the M-key labels. The "PLC flag" group contains settings for the flag to be set when the M-key is actuated.

If SFC flags are used in the PLC program, the POU name and the name of the step or action must be entered in the input fields and, if necessary, the action element ("Force", "Force on", "Force off") must be checked. At runtime, the executability at the left (or right) margin of the operating line indicates whether the step or the action can be executed or not. If no feasibility is indicated and the M-key is actuated nevertheless, pressing the key during runtime opens the criteria analysis, which provides information about the lack of feasibility.

If free flags are used instead of SFC flags, it is not possible to define an automatic feasibility display. In this case, any PLC flag controlling the feasibility display can be entered in the "Feasibility display" field. This does not affect the setting of softkey flags.

Configuring the user interface

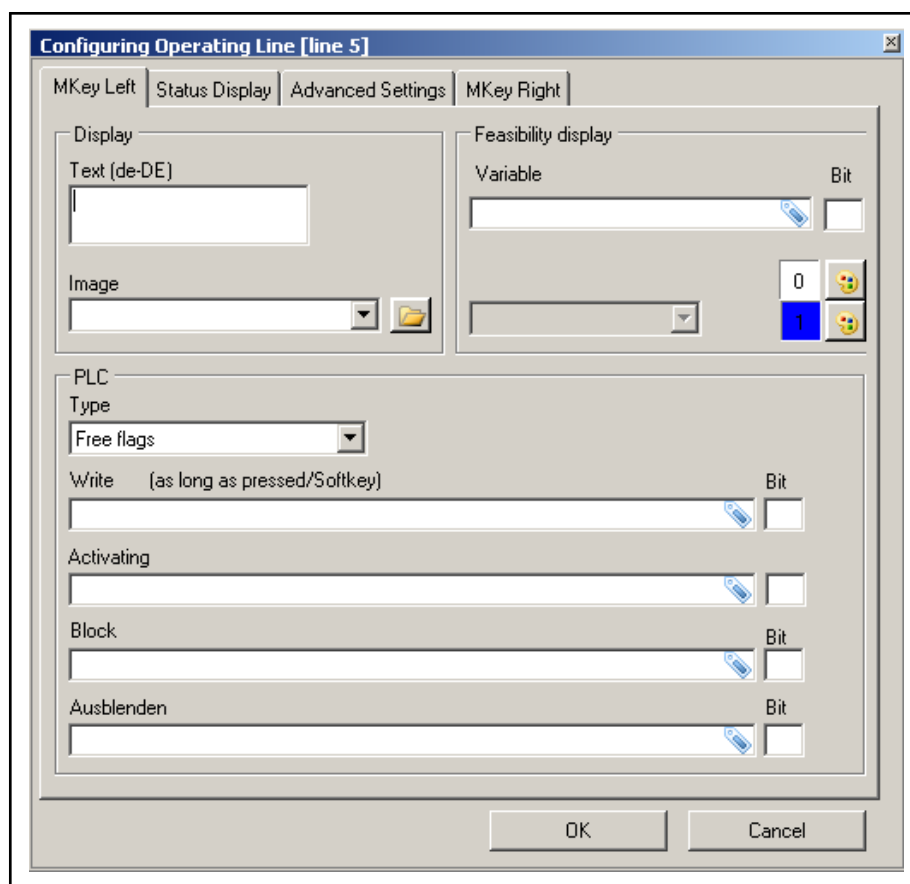


Fig. 5-112: Operating line configuration dialog, "M-key Left" tab

"Status display" tab

This tab can be used to configure a title and a variable for each of the status displays. In addition, colors indicating the value of the variables can be selected for the status display.

Configuring the user interface

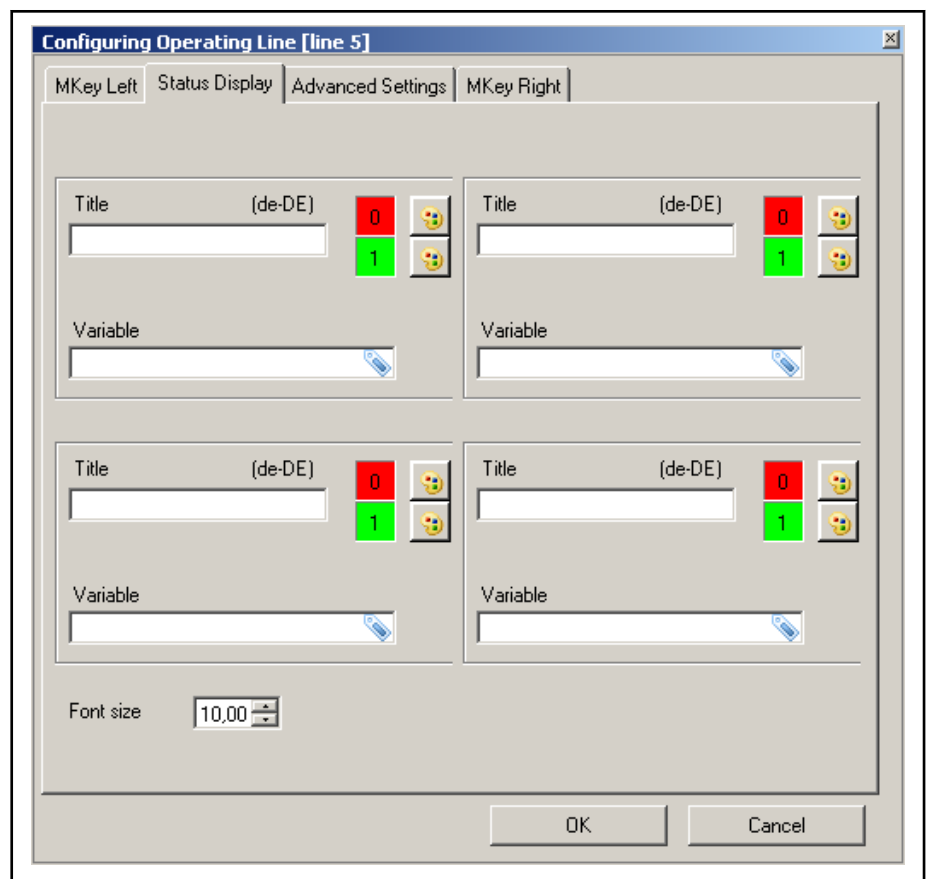


Fig. 5-113: Configuration dialog for an operating line, "Status display" tab

"Advanced Settings" tab

This tab can be used to enter a descriptive text for the entire operating line and to select the color for the text and the background. It is also possible to set the background color for the entire operating line.

"Axis" is used to configure the position display. The position display can either be read from an NC program by means of the axis and the channel numbers or determined via any PLC variable.

The "Format" group can be used to define the number of decimal places to be displayed as well as any text defining the unit of the positional value displayed. Additionally, font and background color of the position display can be configured.

Configuring the user interface

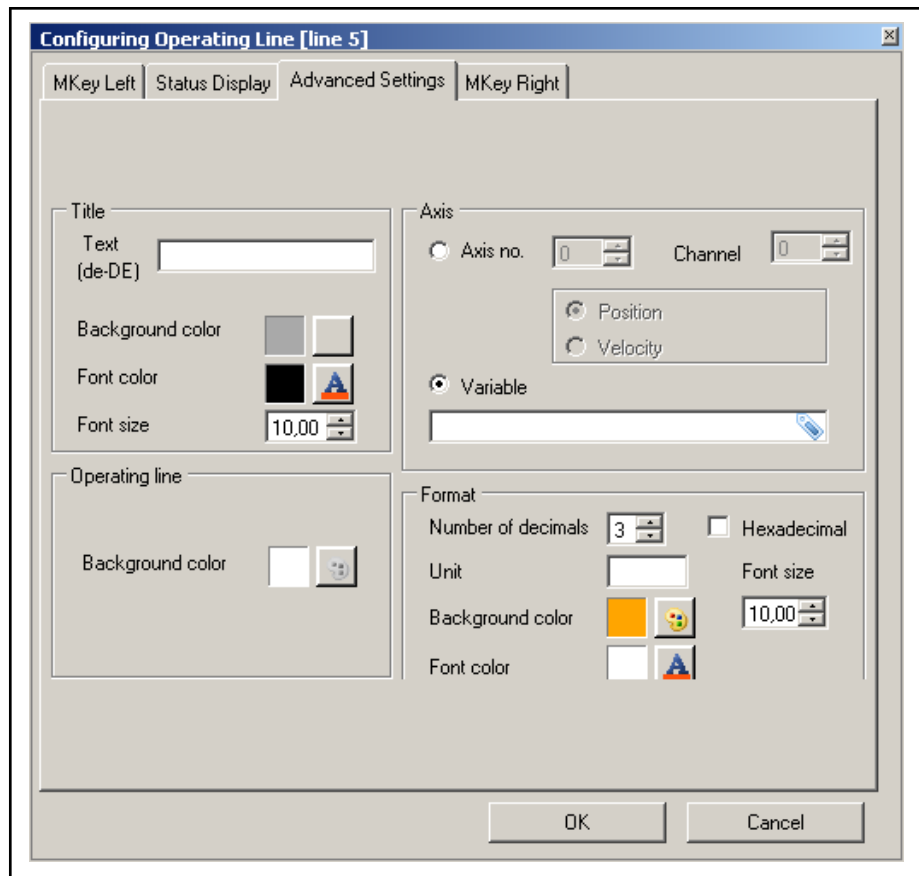


Fig. 5-114: Configuration dialog for an operating line, "Advanced settings" tab

Operating screen editor for multi touch user interface

The operating screen is adjusted as follows for the multi touch user interface.

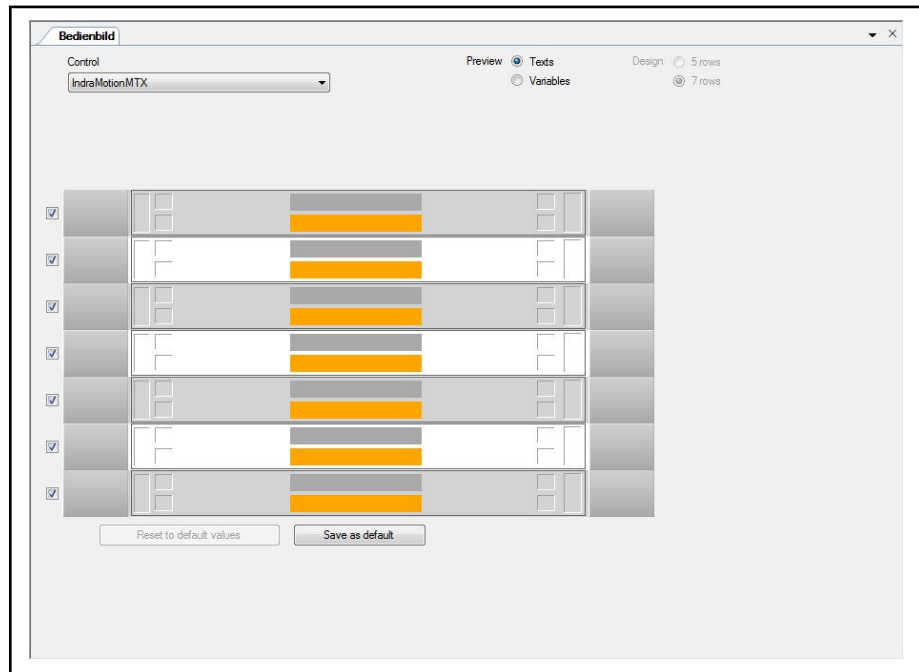


Fig. 5-115: Operating screen editor for multi touch operation

Only 7 lines can be configured in the operating screen.

Configuring the user interface

Displaying texts and graphics on M-keys is not possible. M-keys to the left and right are only part of the operating line, they are not summarized in the M-panel.



For multi-touch HMI devices, use the function block "IL_HmiKeys" instead of "IL_MKeys". It is located in the library "RIL_HMI_Utilites".

5.13 Configuring ACI screens

5.13.1 General Information

Brief description

In the ACI (Application Container Interface) screens, controls created by the customers (user controls) and controls delivered for the system installation (standard controls), may be embedded and arranged in screen segments. The ACI screens can then be integrated into the HMI screens as screens of the type "UserACIScreen".

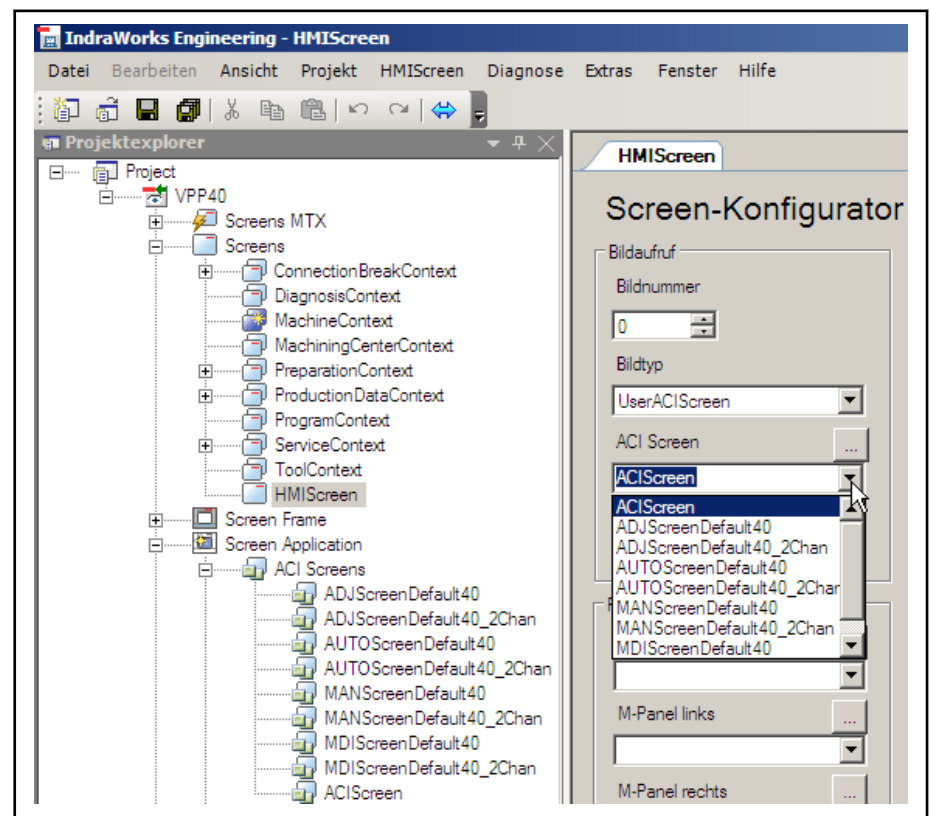


Fig. 5-116: Dialog to add a UserACIScreen to an HMI screen

Description

As it is the case for the user screens created in WinStudio, the displays of a wide variety of process data can be summarized in the ACI screens. For this purpose, the ACI screens are divided into screen segments and the respective controls are inserted.

The various process data of the control is shown using separate displays (e.g. position display). These are called controls. These controls are implemented as software ActiveX or .NET controls. To provide the controls with an advanced functionality within the ACI screens, their implemented interface has to be the type library "ACI.Interface.dll". This is used to report additional events from the ACI screen and to activate commands. The advanced controls are called ACI Controls.

Configuring the user interface

In IndraWorks Engineering, change the segmentation, control selection and control arrangement of the existing ACI screens using the ACI screen editor. Create new ACI screens and delete ACI screens that are no longer needed.

In IndraWorks Operation, a special configuration mode is available to adapt the screen segments of the ACI screens and the control displays to the actual size of the ACI screens in the real user environment using the OP, M and F-keys and the header.



The standard ACI screens, standard controls and other standard entries in the dialogs and in the ACI screen editor depend on the respective application type of the visualization device.

5.13.2 Configuration in IndraWorks Engineering

General information

Brief description In IndraWorks Engineering, use the respective editor and functions that can be called directly to modify and delete existing ACI screens and to create new ACI screens.

Description In IndraWorks Engineering, the ACI screens are managed under the project node "ACI Screens".

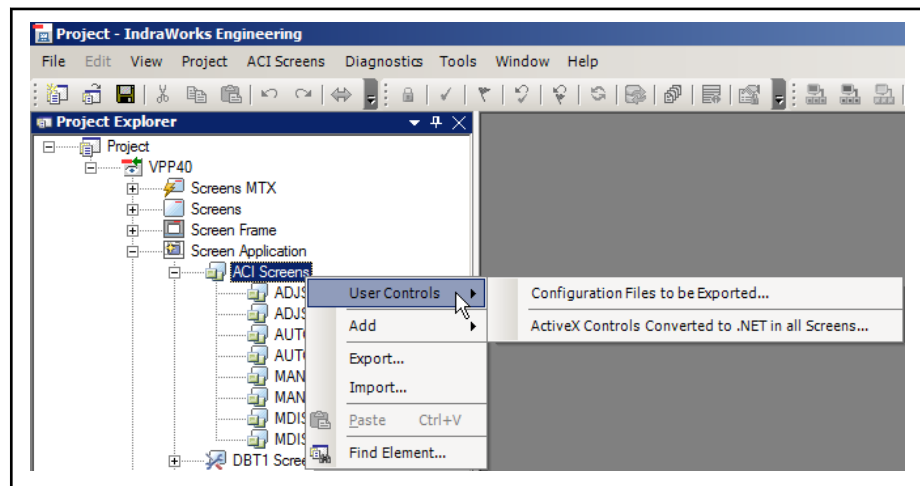


Fig. 5-117: ACI screens (context menu)

The dialogs and functions to configure properties shared by all the ACI screens as well as those to create a new ACI screen can be opened using the context menu of the project node "ACI Screens".

The ACI screen editor and additional dialogs and functions for the specific configuration of an individual ACI screen, such as type, number and position of process data displays, can be accessed via the context menu of the project node of the respective ACI screen.

Configuring the user interface

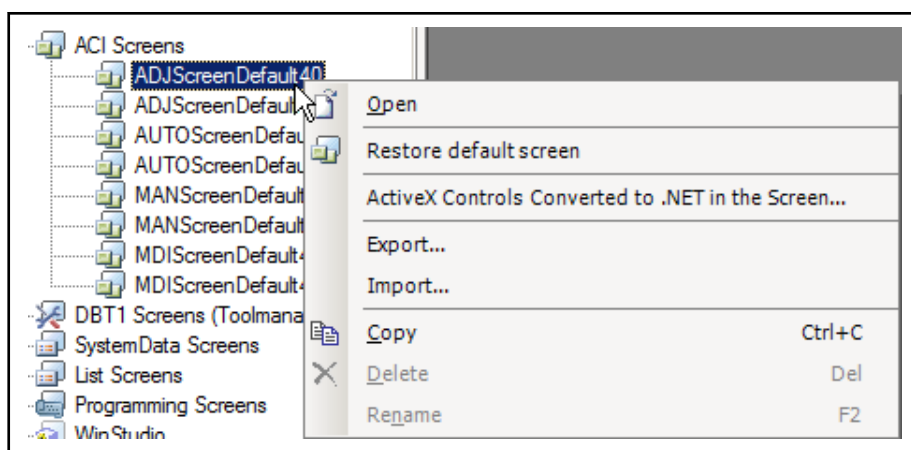


Fig. 5-118: ACI screens (ADJScreenDefault40, context menu)

In the following, the individual dialogs and functions are described in more detail.

ACI screen editor

Call the ACI screen editor via the context menu item "Open" or by double-clicking the left mouse button on the project node of the ACI screen to be edited.

The division of the screen into segments, including their number and position, can be changed in this editor. Furthermore, certain focusing properties of the screen segments as well as the type, number and order of the controls in the screen segments and their respective F-panel may be configured.

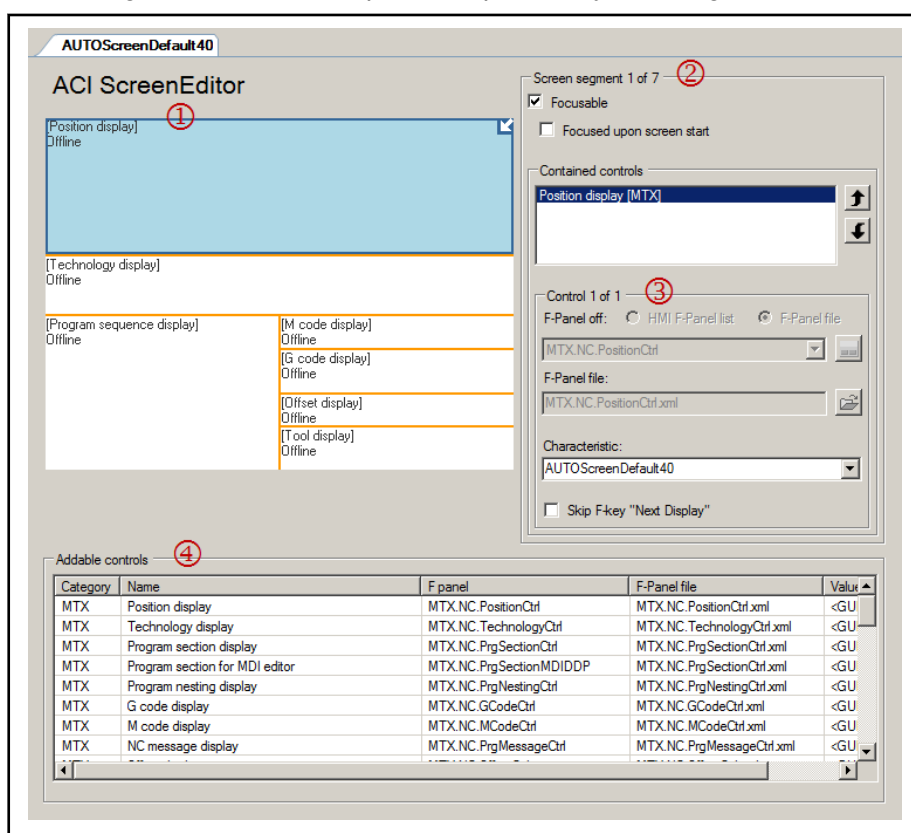


Fig. 5-119: ACI screen editor (with controls without process connection inserted into segments)

Configuring the user interface

Sections of the ACI screen editor

Digit	Range
①	Screen layout
②	Properties of the screen segments and contained controls
③	Control-related settings
④	Addable controls

① Screen layout

In the context menu of the focused screen segment, there are the following functions to structure the screen layout into screen segments:

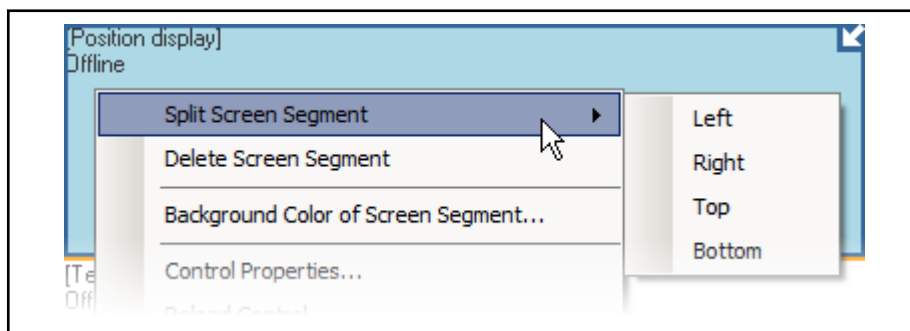


Fig. 5-120: Focused screen segment (only screen segment functions, context menu)

- **Split screen segment - left, right, top, bottom:**

The screen segment is divided horizontally or vertically through the middle.



Screen segments can only be divided until the minimum size of screen segments is reached.

The number of screen segments is only limited by the minimum size.

- **Delete screen segment:**

The screen segment is deleted. The space of the deleted screen segment is taken up by expanding the neighboring screen segment that was generated before by division.

- **Background color of screen segment...:**

Select the background color of the screen segment in the "Color" dialog to provide a better visual separation of the screen segments.

Configuring the user interface

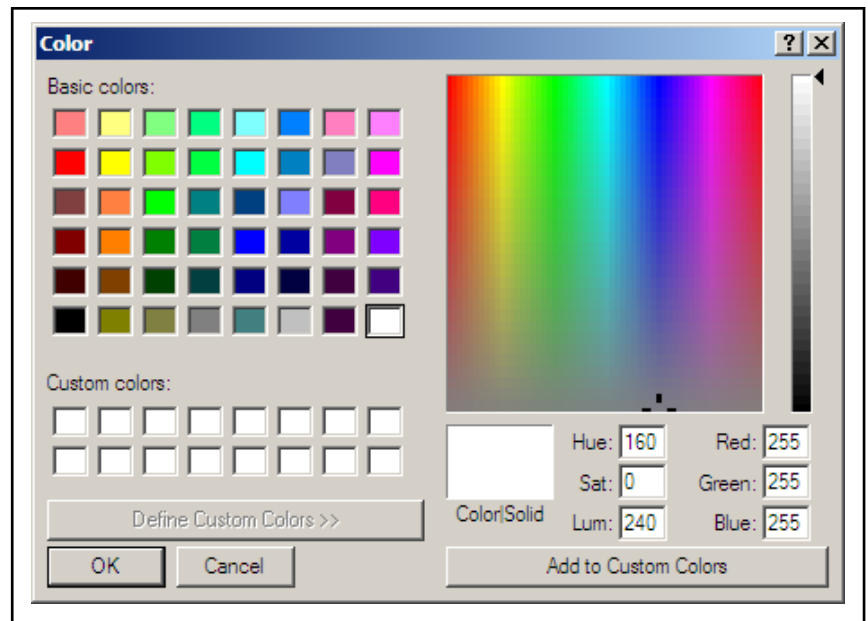


Fig. 5-121: Color selection

The size of the screen segments can be changed by moving their dividing lines. Move the cursor to the dividing line. The cursor icon changes. While holding the left mouse button down, drag the dividing line with the mouse.



The controls are instantiated in the screen layout. For the controls to display process data, a control has to be assigned to the visualization device and be online.

② Properties of the screen segments and contained controls

Use the screen segment properties to define the properties of the focused screen segment that it is to have in the ACI screen of IndraWorks Operation.

- In IndraWorks Operation, press the F-key <Next window> to switch to the next screen segment in a specified order and focus it. The screen segment index "Screen segment x of n" indicates the position of the screen segment in this focusing order. This order is specified due to the position of the screen segments in the screen - according to the rule "from the top to the bottom, beginning at the left-most" - and cannot be changed.
- Use the checkboxes **Focusable** and **Focused upon screen start** to determine whether the screen segment is to be focusable and whether it is to be immediately focused automatically when the screen is displayed for the first time after starting IndraWorks Operation.



A screen segment that contains no control, is not focusable in IndraWorks Operation (except in configuration mode) even if the **Focusable** checkbox is selected.

- The "Contained controls" are controls of the screen segment listed in the display order used for the F-key <Next display> in IndraWorks Operation. Use the **Arrow up** and **Arrow down** buttons to change the control position (index) in the display order.

The following control functions can be called in the context menu:

Configuring the user interface

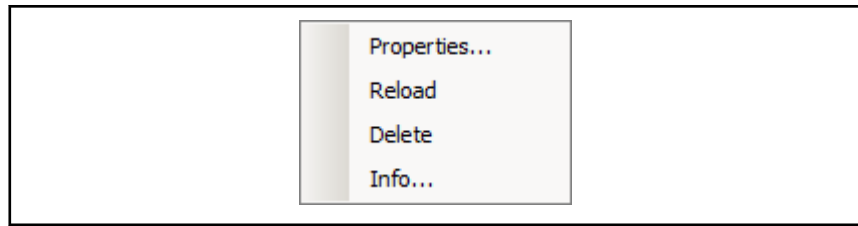


Fig. 5-122: Contained controls (context menu)

- **Properties...:**
Displays the configuration dialog of the control if supported by the control.
- **Reload:**
Ends the old instance of the control and generates a new one.
- **Delete:**
Deletes the control from the screen segment.
- **Info...:**
Displays detailed information on the control instance in a dialog and the information the control was provided with during creation.

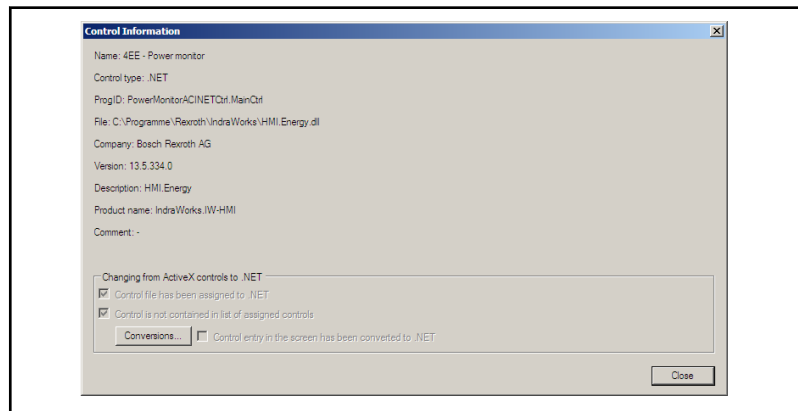


Fig. 5-123: Control information

In addition, the "Changing from ActiveX controls to .NET" information indicates whether the control file was changed to .NET, i.e. whether Assembly and Class of the control exist and the control type is therefore ".NET", whether the control is contained in the list of the controls changed to .NET, and whether the control input in the screen was changed to .NET, i.e. from ProgID to Assembly and Class. Use the **Conversions...** button to open the respective "Changing from ActiveX controls to .NET" dialog (["Dialog ActiveX controls converted to .NET" on page 163](#)).

The same functions can be accessed via the context menu of the screen layout.

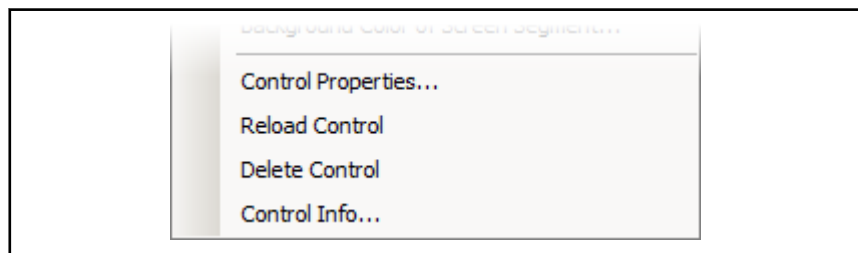


Fig. 5-124: Screen layout (only control functions, context menu)

Configuring the user interface

③ Control-related settings

In this section, settings for the display of the control selected in the list "Contained controls" in the ACI screen of IndraWorks Operation can be carried out.

- The control index "Control x of n" indicates the control position in the display order of the controls in the screen segment in IndraWorks Operation with the F-key <Next display>.
- Each control may be assigned an F-panel that is displayed when the control is focused in the ACI screen of IndraWorks Operation.

Select the respective F-panel as follows:

- From the F-panels created in the visualization device (HMI F-panel list)
- From an F-panel file that may have been created by another program than IndraWorks Engineering and has to be selected in the dialog "Select F-panel file"

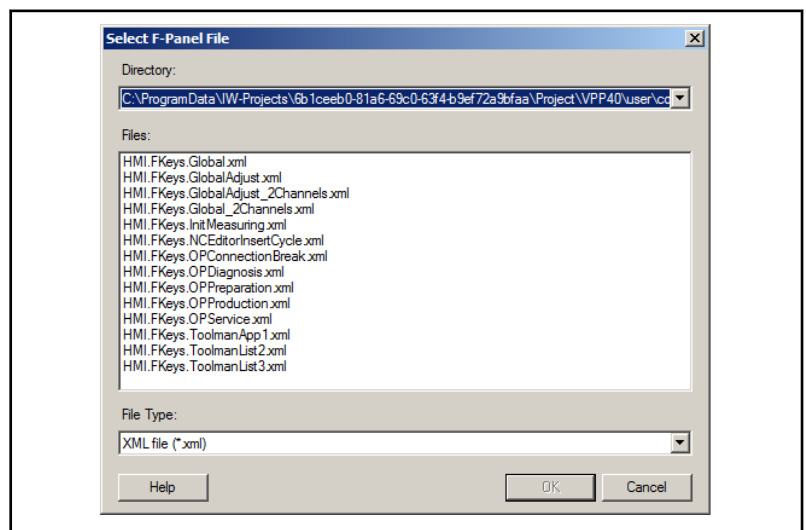


Fig. 5-125: Select F-panel file



In IndraWorks Operation, F-panel files are taken from the directory where they are first found according to the following order:

1. <IndraWorks ProjectDirectory>\<VisualizationDevice>\user\config
2. <IndraWorks InstallationDirectory>\config

If a file from the installation directory was selected in the "Select F-panel file" dialog, it is taken from the project directory in IndraWorks Operation if it exists there.

The button **F-panel symbol** opens the editor of the F-panel selected and created in the visualization device.

The button **Select file** opens the dialog "Select F-panel file".



The F-panel and F-panel file may not be changed for standards controls.

- If a control contains the implementation of the ACI interface, the configuration parameter that can be entered in the field "Characteristic" can be

Configuring the user interface

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 instantiation of the control to determine which of these characteristics the control is to use.

If the control has several characteristics, the configuration parameter may be assigned the following values in addition to direct input:

- With a new GUID (unique combination of numbers and letters)
- With another value from the selection list



The respective control entry in the list "Addable controls" ("There are two possibilities:" on page 156) determines which of these additional assignments is possible.

- The command "Next Control" of the F-key function "Command to ACI screen " (fig. 5-140 "F-panel editor: Assign ACI functions to F-keys" on page 175) creates the F-key <Next display> for IndraWorks Operation. If the control is not to be displayed with this F-key in IndraWorks Operation, the checkbox **Skip F-key "Next display"** has to be enabled.

④ Addable controls

The list "Addable controls" includes the controls that may be inserted in the screen segments of an ACI screen. It consists of the entries of frequently used standard controls and may be extended by entries from user controls.

The following columns define the entry of a control:

- **Category:**
Application type that the control is assigned to. By default, the "User" category is assigned to controls that are added by the user at a later date.
- **Name:**
Designates the purpose of the control. By default, the ProgID (in case of ActiveX controls) and/or the Class (in case of .NET controls) is used for controls that are added by the user later on.
- **F-panel and F-panel file:**
F-panel from the F-panel file that is displayed when the control is focused in the ACI screen of IndraWorks Operation.
- **Characteristic:**
Configuration parameter transferred to the control via the ACI interface that defines the characteristic of the control.

The following functions can be called in the context menu:

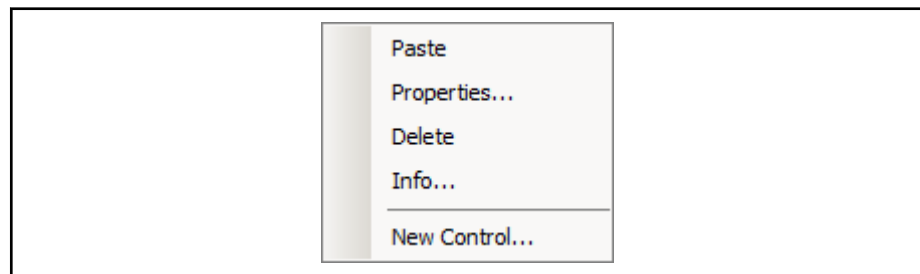


Fig. 5-126: Addable controls (context menu)

- **Paste:**
Inserts the control defined by the list entry into the focused screen segment of the screen layout.

Configuring the user interface

Insertion is also possible via drag-and-drop.



The number of controls in a screen segment is not limited.

- **Properties....:**

Displays the dialog where the settings stored for the control entry may be changed.



The settings of standard control entries may be displayed but not changed.

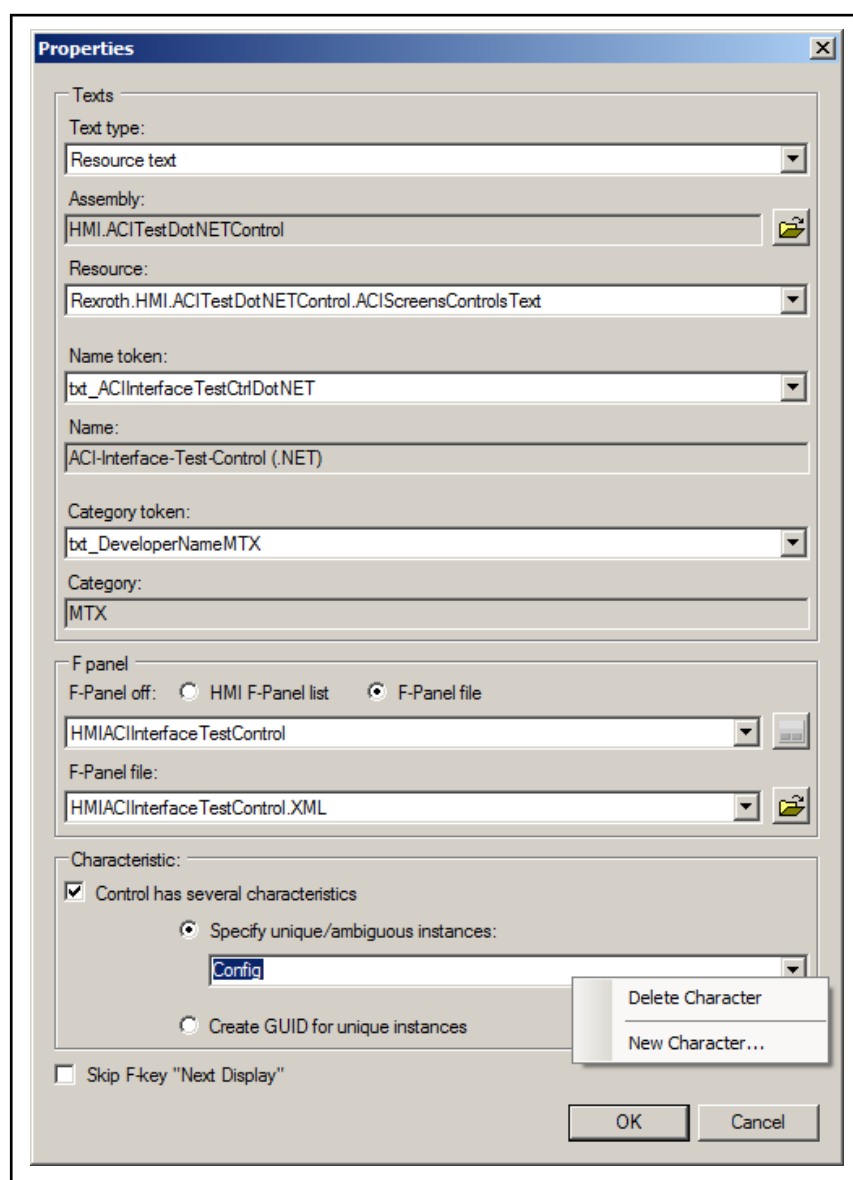


Fig. 5-127: Properties (of a control entry in the list "Addable controls")

- **Texts:**

The texts for name and category of the control may be preset by the selected text type ("Fixed text") or language-dependent ("Resource text", "User text").

Configuring the user interface

Select the language-dependent texts as follows:

- **Resource text:**
 1. Indicate the assembly by clicking on the **Select file** button to open the "Select assembly file" dialog and selecting the assembly
 2. Select the resource for the text in the assembly
 3. Select the token for the text in the resource



The resource texts have to be created during assembly creation.

- **User text:**
There is the list "ACI ScreenControls" for individual user texts. Create and select the texts there. Use the functions that may be called from the context menu of the token field to create the text.

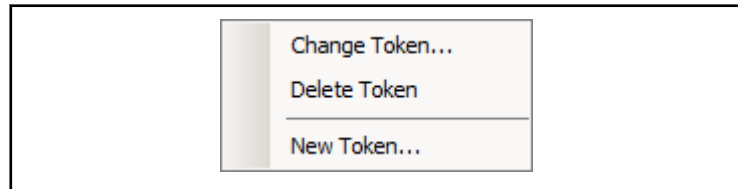


Fig. 5-128: Token (context menu)

- **F-panel:**
"Select the respective F-panel as follows:" on page 153
- **Characteristic:**
Here, the configuration parameter transferred to the control via the ACI interface can be created and selected. This parameter defines the respective characteristic of a control with several types of process data and/or displays.

There are two possibilities:

- **Specify unique/ambiguous instances:**
If the control already has the respective sections in its configuration file for its characters, the character to be used during instantiation of the control has to be specified. It may be entered directly or selected from a list. Use the functions that may be called from the context menu of the characteristic field to create the list.

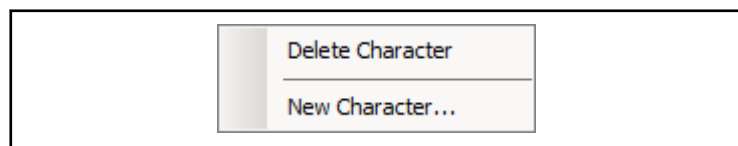


Fig. 5-129: Characteristic (context menu)

- **Create GUID for unique instances:**
A unique identifier is required if the control is to create a new section in its configuration file during instantiation where it saves changes to its settings. This is achieved by generating a GUID (unique combination of numbers and letters).

Configuring the user interface

- The command "Next Control" of the F-key function "Command to ACI screen " (fig. 5-140 "F-panel editor: Assign ACI functions to F-keys" on page 175) creates the F-key <Next display> for IndraWorks Operation. If the control is not to be displayed with this F-key, the checkbox **Skip F-key "Next display"** has to be enabled.

- **Delete:**

Deletes the control entry from the list.



"Deletion" is not possible with standard controls.

- **Info:**

Displays the information for the control defined by the list entry in the dialog (fig. 5-123 "Control information" on page 152).

- **New control...:**

Select the control in the "New control" dialog for which a new control entry is to be created in the list of insertable controls.

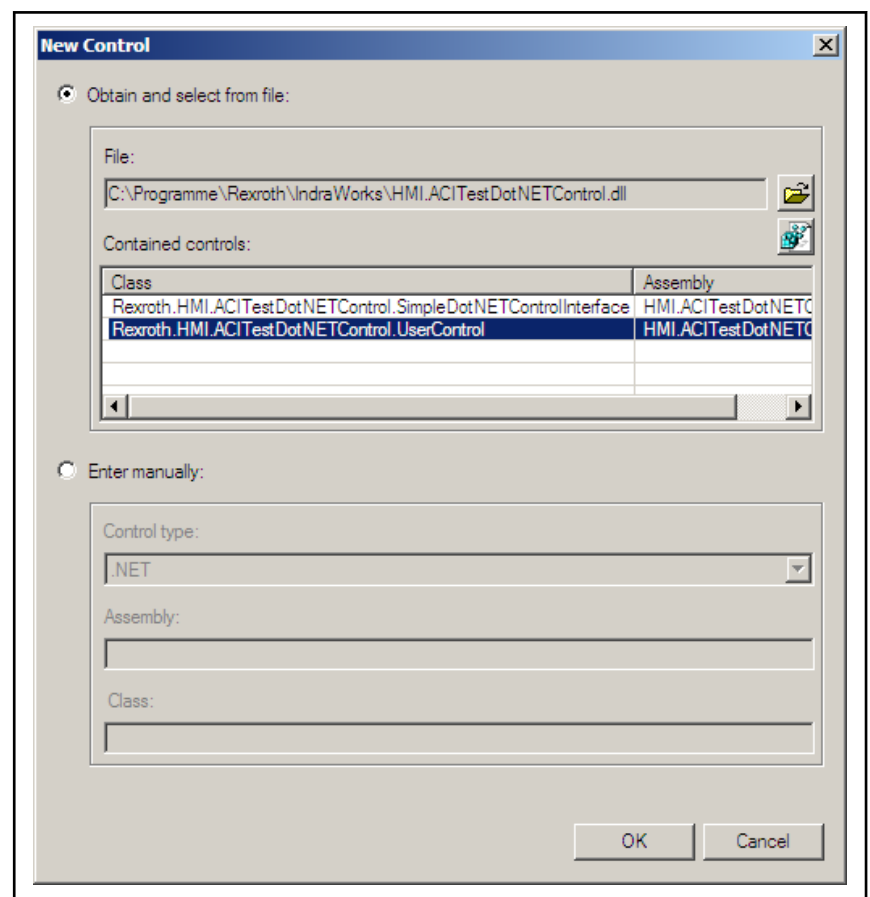


Fig. 5-130: New control

To identify the control, the ProgID (for ActiveX controls) or the Assembly and Class (for .NET controls) have to be indicated.

This may be done in two ways:

- **Obtain and select from file:**

Click on the **Select file** button to open the "Select file" dialog and select the file (*.dll or *.ocx) that contains the control. Then, the

Configuring the user interface

program identifies all controls contained in this file and lists their ProgIDs or Assemblies and Classes. Select the respective control from this list.



If the program cannot determine a ProgID from an ActiveX file, the reason may be that the controls from this file were not yet entered into the registry (Windows data base). Use the **Register** button to carry out registration.

– Enter manually:

Depending on the control type the ProgID or Assembly and Class may be entered directly.



If a standard control is selected, its known settings are adopted for the control entry.

Context menu functions of ACI elements

Function "Add ► ACI screen..."

A new ACI screen can be created using the "New screen" dialog which is opened by selecting the entry **Add ► ACI screen...** in the context menu of project node "ACI Screens" (up to 13V04 **New screen...**).

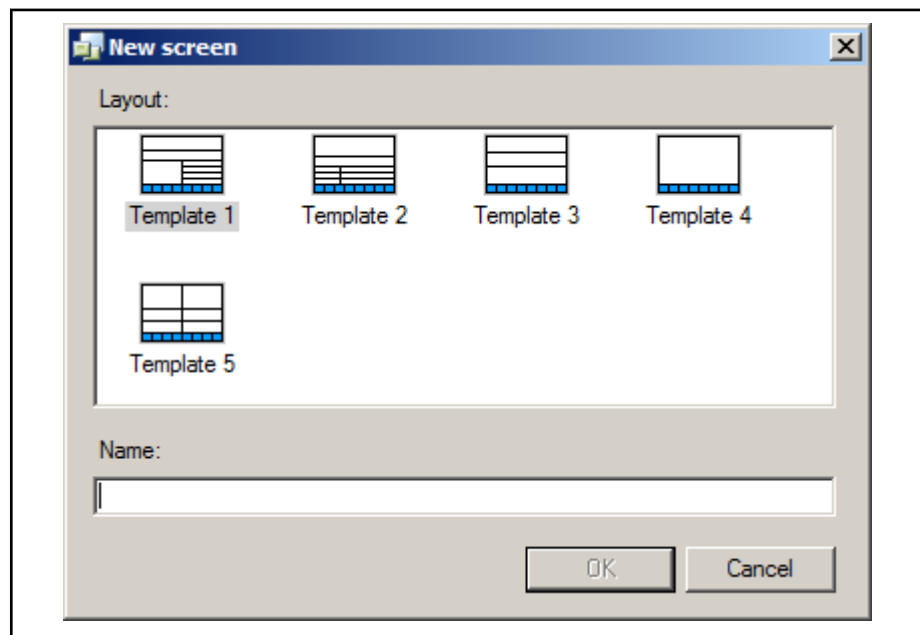


Fig. 5-131: New screen (layout templates)

To create a new screen, select a layout template and enter a unique screen name.

The layout and the content of the screen segments of the created screen can be adapted as required in the ACI screen editor (fig. 5-119 "ACI screen editor (with controls without process connection inserted into segments)" on page 149).



It is not possible to create a new screen with a name that is already in use, even if upper and lower case differ. When this is tried, a respective message appears.

The number of new screens is not limited.

Configuring the user interface

Function "Restore standard screen"

If the ACI screens provided during system installation (standard screens) were changed, they can be reset to their delivery state with the function "Restore standard screen".

Functions "Export"/"Import"

The functions "Export" and "Import" are called via the respective context menu entry in the project node of the respective ACI screen or via the project node "ACI Screens".

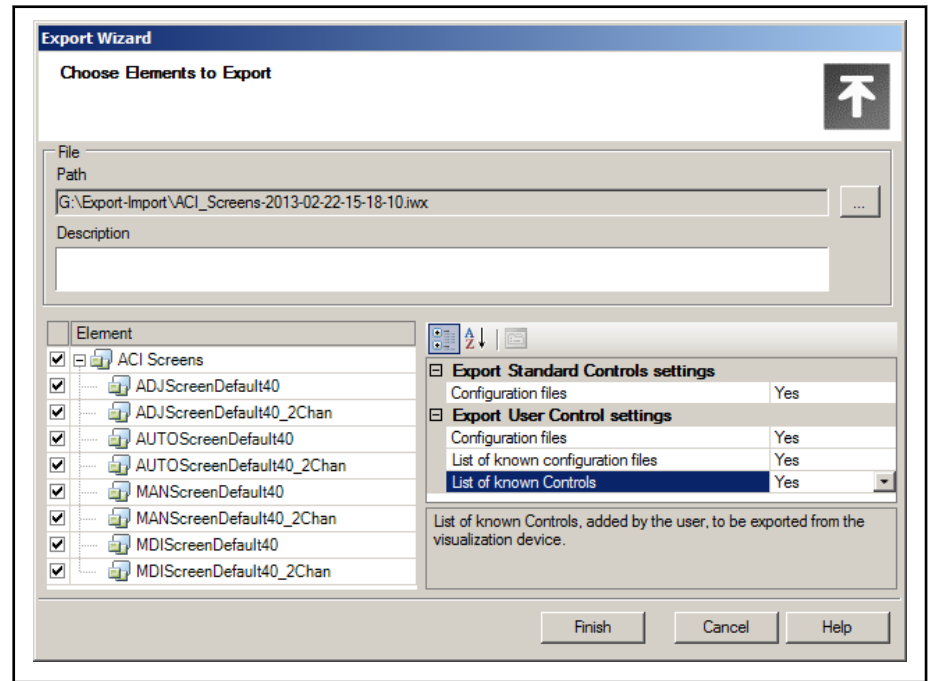


Fig. 5-132: Export of ACI screens

- **ACI screens:**

During export, the data of the selected ACI screens is written to the export file. During import, new ACI screens are created that correspond to the ACI screens previously exported, including their names, provided that no ACI screens with the same name (even with differing case) exist. Otherwise, select whether the already existing ACI screens are to be overwritten or whether the exported ACI screens are to be inserted as additional screens. In the latter case, an incremental counter is added to their names.



If during import of the ACI screen another ACI screen already exists with an identical name (including in another case) in the same or in another visualization device and if the exported ACI screen is to be inserted as an additional screen, the configuration settings of the controls contained therein are reset and initialized in the original version. This prevents changes of the controls of another screen, while changing the options of the controls.

- **User control settings:**

When the project node "ACI Screens" is exported, the following user control settings are saved to the export file and written to the project during import:

- **List of controls changed to .NET:**

Configuring the user interface

List of ActiveX user controls changed to .NET ("[Dialog ActiveX controls converted to .NET](#)" on page 163)

- **List of known controls:**

List of known controls to be exported in the visualization device that were added by the user ("[④ Addable controls](#)" on page 154)

- **List of known configuration files:**

List of known configuration files to be exported in the visualization device that were added by the user ("[Dialog Configuration files to be exported](#)" on page 161)

- **Configuration files:**

Configuration files of user controls whose names are entered into the list of the known configuration files to be exported in the visualization device ("[Dialog Configuration files to be exported](#)" on page 161)

- **Standard control settings:**

When the project node "ACI Screens" is exported, the following standard control settings are saved to the export file and written to the project during import:

- **Configuration files:**

Configuration files of standard controls

Functions "Copy"/"Paste"

Call the functions "Copy" and "Paste" via the respective context menu entry in the project node of the respective ACI screen or via the project node "ACI Screens".

A new ACI screen is created that is equal to the previously copied ACI screen. Its name is that of the copied ACI screen plus an incremental counter. In the event that the copied ACI screen is inserted into another visualization device of the project and does not yet exist there, the incremental counter is not added.



If an ACI screen with the identical name (including another case) already exists in the same or in another visualization device when inserting the ACI screen, the configuration settings of the controls in the inserted screen are reset and it is initialized in the original version. This prevents changes of the controls of another screen, while changing the options of the controls.

Function "Delete"

Use the context menu entry at the project node of the respective ACI screen to delete the respective screen after confirming a security prompt.

Deletion may also be executed with the key.



It is not possible to delete standard screens.

Function "Rename"

The name of an ACI screen can be changed directly in the respective project node.

Configuring the user interface

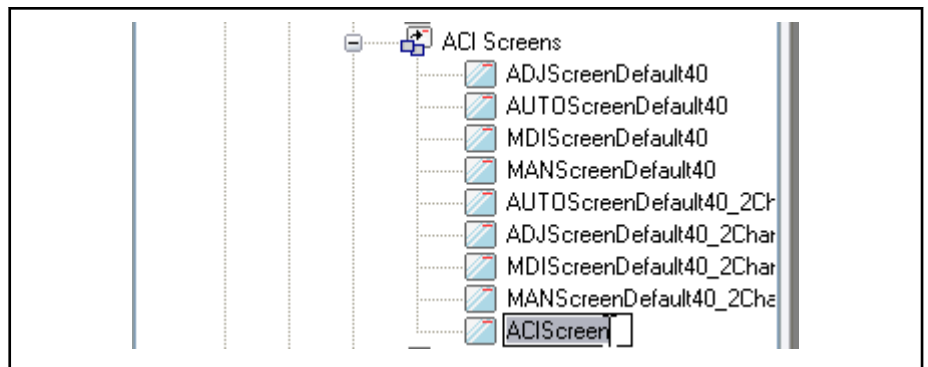


Fig. 5-133: Renaming an ACI screen in the project node

For this purpose, select the function "Rename" via the respective context menu entry in the project node of the ACI screen or click on the project node name again after selecting it with the mouse.



It is not possible to use an existing name (even with a different case) when renaming a screen. When this is tried, a respective message appears.

It is not possible to rename standard screens.

Structuring in folders (from 13V06)

ACI screens can be structured with folders. Use "Add", "Import", "Paste" or "Move" to assign the ACI screens to folders.

Create a new folder with the context menu function **Add ► Folder**. The number of folders and their nesting depth is not limited.

The mode of operation of the context menu functions "Add", "Export"/"Import", "Copy"/"Paste", "Delete" and "Rename" of the folders is analogous to the ACI screens.

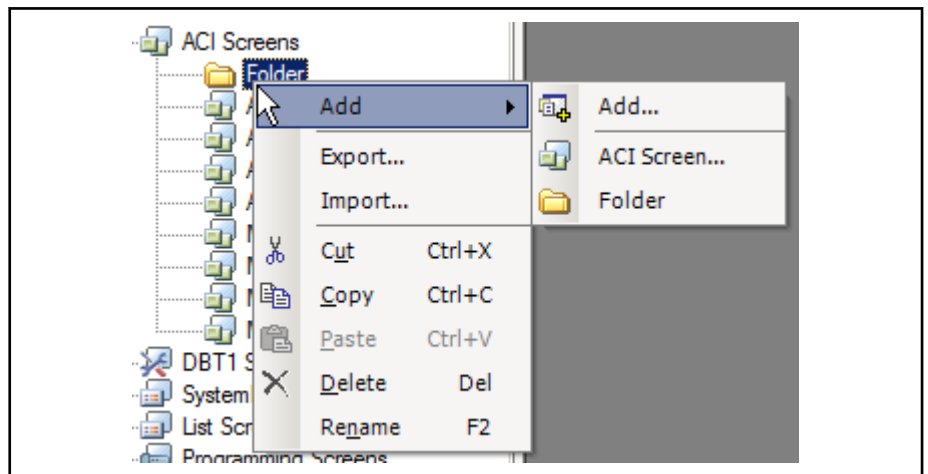


Fig. 5-134: Folder (context menu)

Dialogs for user controls

Dialog "Configuration files to be exported"

Open this dialog in the context menu of the project node "ACI Screens" with the entry **User controls ► Configuration files to be exported...**

Configuring the user interface

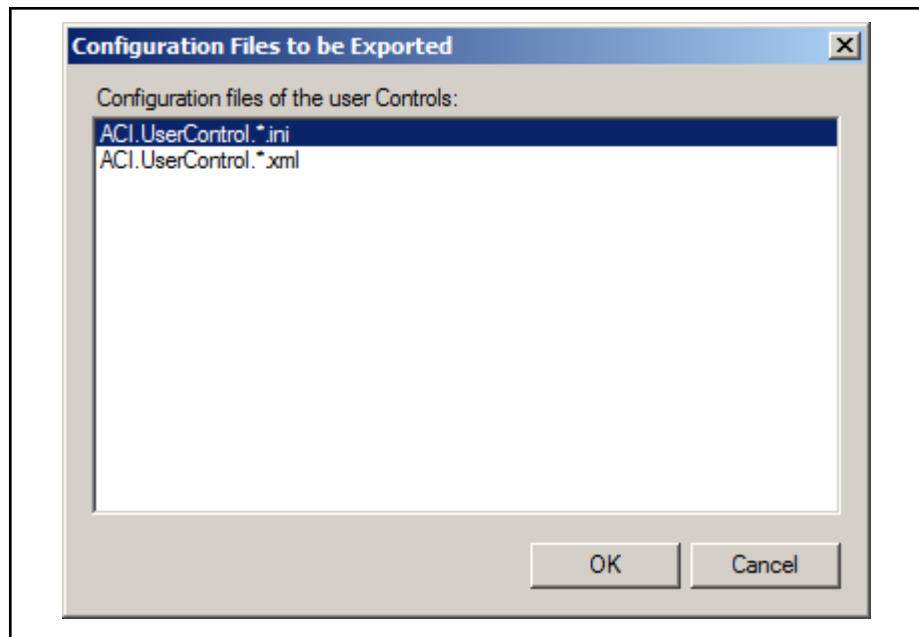


Fig. 5-135: Configuration files to be exported

In addition to the configuration files of standard controls, the configuration files of user controls may also be exported/imported and subjected to the VCS that are stored in the directory "<IndraWorks project directory>\<visualization device>\user\config". This is done by default for all user control configuration files that are named according to the naming patterns "ACI.UserControls.*.ini" or "ACI.UserControls.*.xml".

If the configuration files are named differently, they may be added to the list of configuration files of user controls via the context menu.

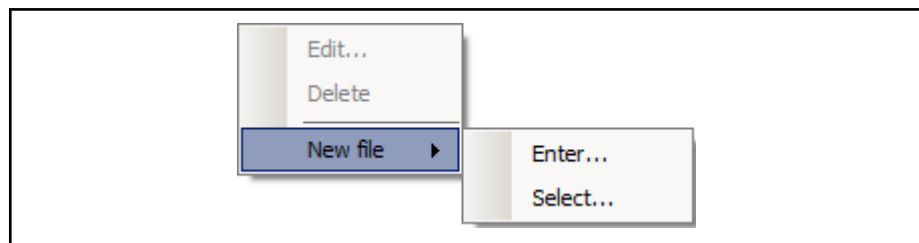


Fig. 5-136: Configuration files of the user controls (context menu)

- **Edit...:**
Edit file names added to the list later on in a dialog.
- **Delete:**
Delete file names added to the list at a later date.
- **Enter new file...:**
Enter new file names in a dialog.



Use the placeholder symbol "*" to cover several files with the same naming pattern.

- **Select new file...:**
Select files that already exist in the directory "<IndraWorks project directory>\<Visualization device>\user\config" directly in a dialog.

Configuring the user interface

Dialog "ActiveX controls converted to .NET"

Call this dialog in the context menu of the project node "ACI Screens" with the entry **User controls ► ActiveX controls converted to .NET in all screens...**, in the context menu of an ACI screen with the entry **ActiveX controls converted to .NET in the screen...** and in the dialog "Control information" with the **Conversions...** button.

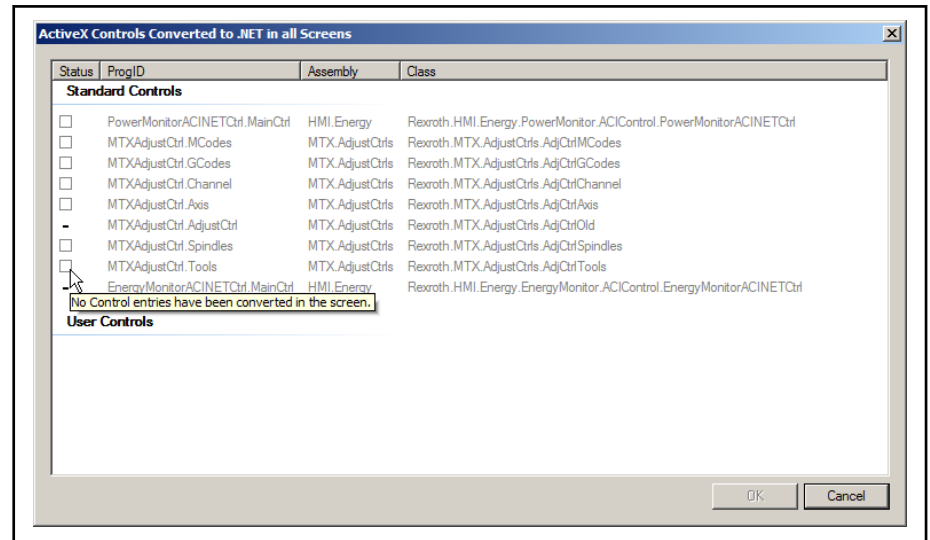


Fig. 5-137: ActiveX controls converted to .NET

From IndraWorks version 13 it is also possible to include purely .NET controls, i.e. without COM registration, into ACI screens. These .NET controls are no longer generated via ProgID but via their Assembly and Class. Therefore, for pure .NET controls that are to replace ActiveX controls or .NET controls with COM elements their existing control inputs in the ACI screens have to be rerouted from ProgID to the Assembly and Class. The dialog lists the conversions from ProgID to Assembly and Class for the respective controls.

Via the list context menu the inputs for user controls may be created.

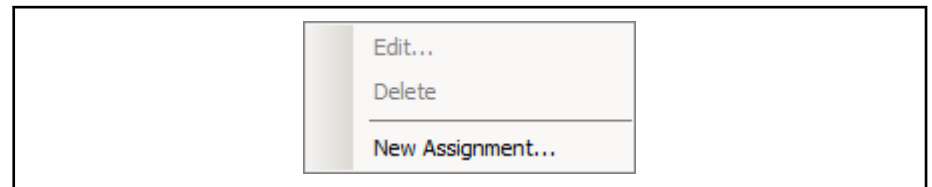


Fig. 5-138: ActiveX controls converted to .NET (context menu)

- **Edit...:**
Change entries for user controls in a dialog.
- **Delete:**
It is possible to delete entries for user controls.
- **New conversions...:**
Enter new conversions in a dialog.

Depending on the context of the dialog call, the "Status" column indicates if the control inputs were changed from ProgID to Assembly and Class in the selected ACI screen or in all ACI screens.

Configuring the user interface

Status

Icon	Description
-	The control was not found in the screen.
☐	No control entries are changed in this screen.
☒	Not all control entries are changed in this screen.
☒	All control entries are already changed in this screen.

Dialog "Apply changes from operating station"

If the entry **Visualization data ► Load changes from operating station** is called in the context menu of the visualization device, the configurations of ACI screens and the configuration files of standard controls are provided among others in the "Apply changes from operating station" dialog if they were changed.

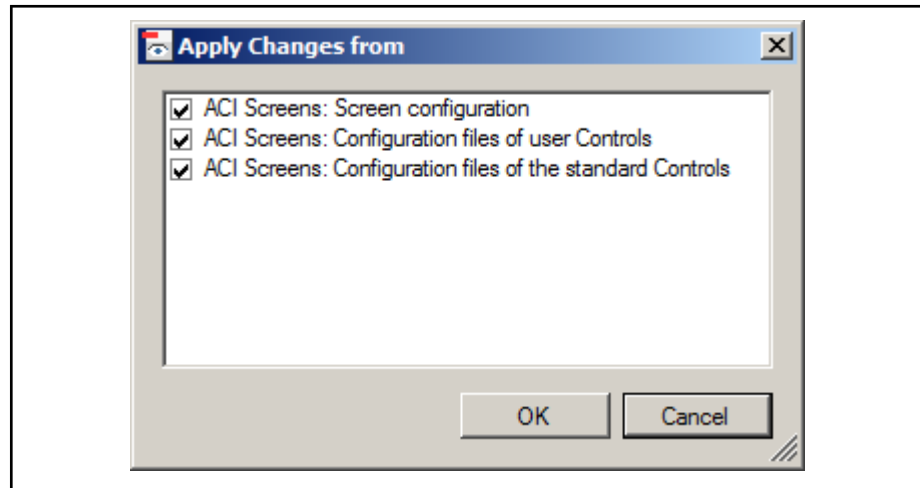


Fig. 5-139: Apply changes from operating station

The changed user control configuration files stored in "<IndraWorks project directory>\<Visualization device>\user\config" may also be provided for selection. This is done by default for all user control configuration files that are named according to the naming patterns "ACI.UserControls.*.ini" or "ACI.UserControls.*.xml". If the configuration files are named differently, the section of the configuration file has to be extended by the entry "<Responsible>ACIUserControls</Responsible>" in the respective file "HRDApplication.*.Files.xml" in the directory "<IndraWorks installation directory>\config".

Example:

```
<File> <Path>%activehmiconfig%</Path> <Name>ACI.UserControl.*.ini</Name> <Responsible>ACIUserControls</Responsible> </File>
```

Validation of ACI screens

Checked criteria

For a general description of the validation of visualization data in the HMI, see [chapter 5 "Configuring the user interface" on page 49](#)

When validating ACI screen visualization data, the following criteria are checked:

- Uniqueness and validity of the screen names
- Use of the screens

Overview on error numbers

An ID is assigned to each discovered problem during the validation of visualization data. Using this ID, the following table can be used to gather information on cause and its recovery:

ID	Cause	Recovery
MTX-00010101	The screen name is not unique.	Rename the screen.
MTX-00010102	The screen name contains at least one of the following characters: \ / : * ! ? " ' < >	Remove the special characters from the screen name.
MTX-00010103	The screen is not used.	Select the screen in a HMI screen of the type "UserA-CIScreen".
MTX-00010111	Some configuration files of the standard controls are not available in the project.	Copy the files listed in the message dialog from the template directory to the project directory.

Tab. 5-8: Overview on error IDs of the validation messages

5.13.3 Integrating User Controls into ACI Screens

General Information

Brief description

It is possible to insert self-created user controls in the screen segments of an ACI screen that were created as ActiveX or .NET controls. This makes it possible to create user-specific data visualizations and to embed them in existing ACI screens. By using the ACI interface in user controls, the behavior of system-specific displays is guaranteed. Additionally, the user control can use the advanced ACI functionality to react on certain events and to communicate with its environment.

Special Features of ActiveX Controls (Visual Basic 6)

Reporting the user control focusing to the ACI via mouse click event

One way of focusing a user control in IndraWorks Operation is using the F-key command "Next Window" (fig. 5-140 "F-panel editor: Assign ACI functions to F-keys" on page 175) to go to the next screen segment. The other way is to click directly on the user control. In the second case, the user control has to trigger the "Click" event so that the ACI is informed and thus displays the focus frame, the background color and the F-panel belonging to the user control. The user control has to trigger the "MouseMove" event if the mouse cursor is in the range of the user control.

```
Event Click()
```



When focusing a user control, all user entries are directed to this control.

When focusing an ACI control to the ACI ("Next Window") using a command, the ACI interface command "SetFocus()" of the control is additionally called.

"Enabled" property of the user control has to be "True"

If the "Enabled" property of a user control is set to "False", the ACI can change the background color of the user control using "BackColor", but the user control cannot trigger the "Click" and "MouseMove" events and the ACI cannot call the control method "SetFocus". To provide a consistent function in the ACI, the "Enabled" property of the user control has therefore to be "True".

Configuring the user interface



If the purpose of setting the "Enabled" property to "False" is to prevent the user control from being focused, this can instead be accomplished by deselecting the checkbox **Focusable** for the screen segment (see [chapter 5.13.2 "Configuration in IndraWorks Engineering" on page 148](#)).

Setting the Background Color of a User Control

The background color of a user control in an ACI screen indicates whether the user control is focused and improves the visual separation of the ACI screen.

It has to be possible to change the background color for the non-focused state during the configuration of the ACI screens. There are one color for the focused state of the user control and freely selectable colors for the non-focused state.

When a user control is inserted into an ACI screen, a default background color for the non-focused state is automatically assigned. This background color can be changed during the configuration of the ACI screens. The freely selectable background color is changed via a respective context menu function (see [chapter 5.13.2 "Configuration in IndraWorks Engineering" on page 148](#)). If a user control has a PropertyPage and/or an "Options" dialog in which the background color can also be changed, use the respective context menu function (see [chapter 5.13.2 "Configuration in IndraWorks Engineering" on page 148](#)) to open these forms and change the background color.

For the ACI to be able to set the focus-dependent background color for all user controls - whether they have been developed especially for the use in ACI or not - the user controls must be provided with the public property "BackColor". .NET controls and general controls, such as Microsoft "Chart" control, are provided with this implementation as a standard.

.NET controls If an embedded control completely covers the background in a .NET control, the event "BackColorChanged" has to be evaluated and the background color of the embedded control is to be set to the color of the .NET control.

(C#)

```
this.BackColorChanged += new System.EventHandler(
    this.UserControl_BackColorChanged);
```

ActiveX controls In ActiveX controls, setting the background color is carried out by the procedure Let for the "BackColor" property in the user control:

Program:

```
(VB6)
Public Property Let BackColor(ByVal New_BackColor As OLE_COLOR)
    UserControl.BackColor = New_BackColor
End Property
```

Alternatively, the control may also react to its environment:

Program:

```
(VB6)
Private Sub UserControl_AmbientChanged(PropertyName As String)
    If PropertyName = "BackColor" Then
        BackColor = Ambient.BackColor
    End If
End Sub
```

Implementing and Using the ACI Interface IACI

To use the ACI interface in the user control, "ACI.Interface.dll" has to be referenced and the ACI interface has to be implemented.

(.NET - C#)

Configuring the user interface

```
using Rexroth.HMI.ACI.Interface;  
(ActiveX - VB6)  
Implements ACI_Interface.IACI
```

In the following, the interface methods and properties are listed and their usage is described:

Active

Syntax: (C#)

```
public bool Active { set {} }  
  
(VB)  
  
Private Property Let IACI_Active(ByVal RHS As  
Boolean)  
  
End Property
```

Usage: This method is used by the ACI to switch on or off the functions of the ACI control.

This message can be used by the ACI control to establish or cancel process connections (start or stop of the OPC communication). If the ACI control is already communicating, it is sufficient to switch the OPC groups only active or inactive.

Parameters: True, if the ACI screen and control are visible

False, if the ACI screen and control are not visible

Return value: -

Channel

Syntax: (C#)

```
public short Channel { set {} }  
  
(VB)  
  
Private Property Let IACI_Channel(ByVal RHS  
As Integer)  
  
End Property
```

Usage: This method reports the active channel of the control to the ACI control during instantiation and channel change.

This parameter is for example required to check in / out channel-dependent items.

Parameters: Number of the active channel

Return value: -

Configuring the user interface

ConfigurationId

Syntax: (C#)

```
public string ConfigurationId { get { re-  
turn ...; } set {} }
```

(VB)

```
Private Property Get IACI_ConfigurationID()  
As String
```

```
WIACI_ConfigurationID = ...
```

```
End Property
```

(VB)

```
Private Property Let IACI_ConfigurationID(By-  
Val RHS As String)
```

```
End Property
```

Configuring the user interface

Usage: With this method, the ACI requests the ACI control to save its configuration settings instance-specifically (e.g. in an individual section of a control-related configuration file). To identify the instance-related configuration data, the ACI control has to specify a unique identifier that is saved as return value by the ACI. During the next instantiation of the ACI control, the ACI returns this identifier to the ACI control (in the "ConfigurationId" property) so that it can parameterize.

The ACI calls the "ConfigurationId" property at the first instantiation of the ACI control (when inserting into a screen segment) in IndraWorks Engineering. The ACI can use this call to display a configuration dialog to allow the user to set the initial settings.

In IndraWorks Operation, the ACI calls the property for a focused ACI control if the ACI control displayed the "Options" dialog previously (by calling the "ShowOptionDialog" method). The reason is that the ACI control might try to save the changed configuration settings under a different identifier that has to be queried and saved by the ACI.

If the ACI control was inserted in a screen segment in IndraWorks Engineering without instantiating it, the "ConfigurationId" property is called in IndraWorks Operation without displaying the "Options" dialog previously. Therefore, after starting IndraWorks Operation, a configuration dialog is displayed if the ACI control requires it for the initial instantiation. The identifier queried via "ConfigurationId" is only saved in the configuration mode of the ACI screen, after the display of the Options dialog, in IndraWorks Operation. Thus, it is required to first switch into configuration mode and then to open the configuration dialog ("Options" dialog) and to close the configuration mode after the changes have been completed and the prompt for saving has been confirmed. However, the project may also be opened in IndraWorks Engineering when IndraWorks Operation is closed. Then, call the ACI screens editor for the respective ACI screen, make the changes in the configuration dialog of the ACI control displayed automatically and save them when exiting the editor (confirm the prompt for saving).

With this property, the configuration identifier previously obtained by ConfigurationId from the ACI control is reported to the ACI control by the ACI during instantiation. Thus, an individual and instance-specific view is possible for each ACI control.

Parameters: Saved configuration identifier

Return value: Unique identifier of the instance-related saved configuration data

Configuring the user interface

Delete

Syntax: (C#)

```
public void Delete()
```

(VB)

```
Private Sub IACI_Delete()
```

Usage: This method is called by the ACI if the ACI control is removed again from the screen segment while configuring the ACI screen.

With this method, it is possible for the ACI control to delete the configuration data (e.g. a profile section in a control-related configuration file) created for the control instance to be removed since it is no longer required.

Parameters: -

Return value: -

Device

Syntax: (C#)

```
public string Device { set {} }
```

(VB)

```
Private Property Let IACI_Device(ByVal RHS As String)
```

```
WriteMethodText "IACI_Device Let: " + CStr(RHS)
```

```
End Property
```

Usage: This method is used to report the current device to the ACI control when it is instantiated. The transferred parameter is the active device GUID of the LDX that is required to establish an OPC process connection.

If in IndraWorks Engineering the control connected to the visualization device is offline, a process connection of the ACI control is not possible. Therefore, the GUID specifically defined for this case is reported:

```
E61948E9-9913-47c0-ACDB-7C716902CD66
```

Parameters: GUID of the active device | E61948E9-9913-47c0-ACDB-7C716902CD66

Return value: -

Configuring the user interface

InvokeCommand

Syntax: (C#)

```
public void InvokeCommand(CommandSettings settings)
```

(VB)

```
Private Sub IACI_InvokeCommand(ByVal settings As ACI_Interface.CommandSettings)
```

Usage: The ACI directly sends individual (subsequently defined) and external (from F-keys or from other user controls) commands to the ACI control using this method. Additional parameters can be transferred.

The transferred command and the parameters have to be known to the ACI control in order to evaluate it and to act correspondingly.

In the following, the predefined commands to report certain events to the ACI controls are shown:

Language has been switched

```
settings.Name = "LanguageChanged"
```

```
commandArgs[0].Value = LCID (1031=DE / 1033=EN / ...)
```

IndraWorks Operation is closed, the connection to the control is interrupted or the control is removed from the screen segment during the configuration of the ACI screen

```
settings.Name = "Terminate"
```

```
commandArgs[0].Value = -
```

Parameters: settings.Name = name of the command

settings.args = string array of parameters

Return value: -

ShowOptionDialog

Syntax: (C#)

```
public bool ShowOptionDialog() { return ...; }
```

(VB)

```
Private Function IACI_ShowOptionDialog() As Boolean
```

```
IACI_ShowOptionDialog = ...
```

```
End Function
```

Usage: If the ACI control can allow changes in its configuration settings by the user, it has to provide the respective "Options" dialog. The request to display this configuration dialog has to be sent to the ACI with the "Show_OptionDialog" via F-key. The ACI calls the "ShowOptionDialog" method in the focused ACI control.

The ACI control has to inform the ACI in the return value about how to proceed with possible changes.

Configuring the user interface

Parameters: -

Return value: True: Dialog was displayed (irrespective of whether changes were saved or cancelled)

False: Dialog not implemented

Zoom

Syntax: (C#)

```
public bool Zoom { set {} }
```

(VB)

```
Private Property Let IACI_Zoom(ByVal RHS As Boolean)
```

```
End Property
```

Usage: If the ACI receives the "Zoom_In" command (via F-key for example), it zooms in the currently focused screen segment and thus the controls it contains up to the complete ACI screen width and height. The "Zoom_Out" command returns to the original size of the screen segment. After zooming, the ACI changes the "Zoom" property to either "True" or "False" in the ACI controls changed in size. Thus, the ACI controls are provided with information on their zoom state and can adapt visually to the new size (changing the number of lines and columns, the font size, etc.).

Parameters: True: ACI control is in zoom mode | FALSE: ACI control is in normal display mode again

Return value: -

Use of the .NET basic UserControl IACIUserControl

The .NET basic UserControl ACI.Interface.dll is a control that contains only the IACI interface and any extensions (additional interfaces) and virtually encapsulates them to the outside. The virtual properties and methods of the control are identical to the definitions in the interface. The basic UserControl can be used to simply derive .NET controls and overwrite the virtual properties and methods if necessary.

Command sequence for the ACI control instantiation

For the first/each following instantiation of an ACI control (in IndraWorks Engineering/Operation), the following commands are called in a specified order:

Command sequence

(C#)

```
ConfigurationId set/get <ID>
```

```
Device <E61948E9-9913-47c0-ACDB-7C716902CD66 or number of current device>
```

```
Channel <No. of the active channel>
```

```
Active <False/True>
```

Reaction to a connection interruption in IndraWorks Operation

If the user interface detects that the connection to the control has been interrupted, the ACI informs all ACI controls using the command

```
InvokeCommand(settings.Name = "Terminate").
```

In this case, all ACI controls have to execute a Disconnect() on the SCP server/OPC server. The connection can only be restored if all clients have logged off.

Connection state to the control

If in IndraWorks Engineering the control that the visualization device is connected to is offline, the following GUID is transmitted in the interface property "Device" as an identifier for a non-existing "pseudo" device.

```
SetDevice()
```

In this case, no control communication is established for the ACI control.

Communication of User Controls via Notify Event



The Notify event is only recommended for experienced user control developers with a profound understanding of and extensive experience with ACI-internal communication options.

By using the Notify event, which was especially defined for the ACI, the user controls can communicate with the ACI and other controls within the ACI screen in IndraWorks Operation.

(.NET - C#)

```
public delegate void CommandEventHandler(string name,
    IList<string> parameters);
```

```
public event CommandEventHandler DoEvent;
```

Notify event (principle)

(.NET - C#)

```
if (DoEvent != null)
{
    IList<string> ilistParams = new List<string>();
    ilistParams.Add(...); //method (command target)
    ilistParams.Add(...); //parameter 1
    ...
    ilistParams.Add(...); //command parameter n
    DoEvent("Notify", ilistParams); //event type
}
```

(ActiveX - VB6)

```
Event Notify(sMethod As String, sParameters() As String)
```



This event declaration is strictly specified including the variable names and can only be identified by the ACI in this way.

Sending commands to the ACI

With "SendCommandTo_Application" as a method parameter (sMethod) in the Notify event, it is possible to send the commands to the ACI that can also be assigned to F-keys ([chapter 5.13.4 "F-Key Functions for ACI Screens and ACI Controls" on page 175](#)).

Notify event (ACI)

(.NET - C#)

```
if (DoEvent != null)
{
    IList<string> ilistParams = new List<string>();
    ilistParams.Add("SendCommandTo_Application"); //method (command target)
    ilistParams.Add("MTXACI"); //application name
    ilistParams.Add("MTXACI"); //application name
    ilistParams.Add("Show_FKeyPanel"); //ACI command
    ilistParams.Add("MDIEditor_Panel"); //command parameter 1
    ilistParams.Add("MTXMDIEditor.xml"); //command parameter 2
    DoEvent("Notify", ilistParams); //event type
}
```



Since the usage of this command direction is internal and thus not visible for the user, it is only suitable to a limited extent and has to be used carefully.

To trace changes better in the user interface, required command calls should be executed via F-keys by the user.

Sending commands to other controls

In IndraWorks Operation, a user control can send a command to all controls, all instances of a specified control or only to a specified instance of a specified control via the ACI interface method "InvokeCommand" within the ACI screen.

Configuring the user interface

The following transfer parameters of the "Notify" event can send a command:

Method (sMethod)	Explanation	Parameters	Explanation
SendCommand-To_All	Sends commands to all controls	i = 0	Command to be sent
		i = 1	Parameter array of the command
SendCommand-To_Control	Sends commands to all instances of a control	i = 0	ProgID of the control
		i = 1	Command to be sent
		i = 2	Parameter array of the command
SendCommand-To_Instance	Sends command to a determined instance of a control	i = 0	ProgID of the control
		i = 1	TimeID of the control
		i = 2	Command to be sent
		i = 3	Parameter array of the command



In order to specify the ProgID and the TimeID for a control, these have to be read from the configuration file.

The TimeID - a type of instance identification - unambiguously identifies the instance of a control. It is generated for the instance of a control when the control is inserted into a screen segment in the ACI screens editor.

Writing to the status line (only with the application type "MTX")

A user control can write its text messages in the status line of the ACI using a "Notify" event. The advantage of writing to the status line in this way is that the status line is used in the same way by the user controls. Therefore, user controls do not need to reference an additional library.

The following transfer parameters of the "Notify" event can be used to access the status line:

Method (sMethod)	Explanation	Parameters	Explanation
ClearStatusbar-Message	Clears the content of the status line	i=0	Corresponds to the message type: ClearStatusText ClearInformationText ClearErrorText
SetStatusbarMessage	It describes the content of the status line	i = 0	Corresponds to the message type: SetStatusText SetInformationText SetErrorText
		i = 1	Text to be displayed

SetStatusbarMessage

```
(ActiveX - VB6)
Dim ssmParam(1) As String
ssmParam(0) = "SetInformationText"
```

Configuring the user interface

```
ssmParam(1) = "Calculation fished"  
Notify "SetStatusbarMessage", ssmParam
```



The text to be displayed may contain umlauts. But text containing an XML syntax is not allowed and not displayed.

User controls are only supposed to write messages in the status line when they are focused. Otherwise, the origin of the message cannot be traced.

Configuring User Controls

ACI screens

User controls have to be inserted into the ACI screens like system-specific controls (standard controls).

The user controls have to save their configuration settings independently as well. Access to the options that can be set by the user is to be executed via configuration dialogs as PropertiesPages or "Options" dialogs (for ACI controls) by the user control.

ACI control

Controls that were especially developed for their usage in ACI screens, i.e. contain the ACI interface, have to save their instance-related data (e.g. in a section of a control-related configuration file) so that they access it via an identification text. This identification text is queried by the ACI via the ACI interface property "ConfigurationId" when the control is generated for the first time, usually when it is inserted into a screen segment. Additionally, the identification text is also queried after each display of the "Options" dialog if the ACI control returns the value "True" for the ACI interface command "ShowOptionDialog". In each future instantiation of the control, the ACI transfers the stored identification text (via the ACI interface property "ConfigurationId") to the control so that it can be displayed instance-specifically.

5.13.4 F-Key Functions for ACI Screens and ACI Controls

Brief description

To call certain actions of the ACI and specific methods of the ACI controls, the respective functions can be assigned to the F-keys.

Description

In the F-panel editor (fig. 5-140 "F-panel editor: Assign ACI functions to F-keys" on page 175), the functions "Command to ACI screen" and "Command to ACI control" can be used to assign public ACI functionalities to F-keys.

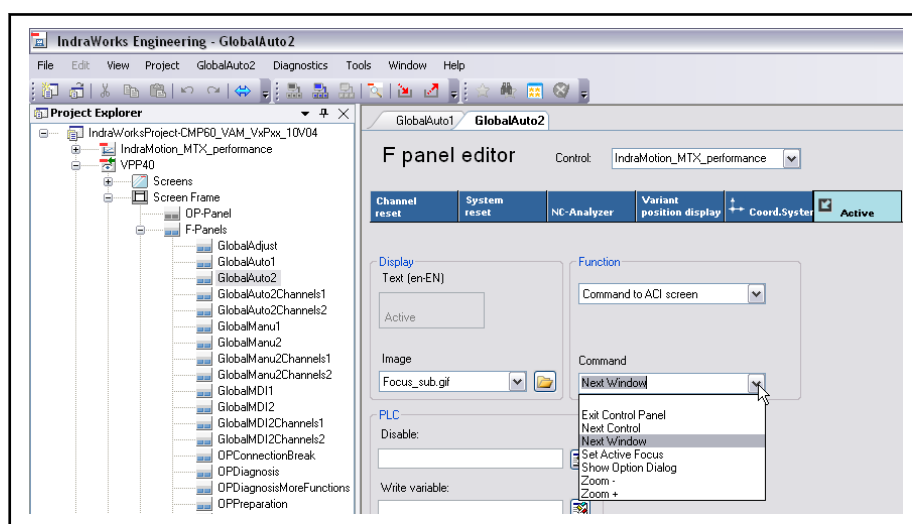


Fig. 5-140: F-panel editor: Assign ACI functions to F-keys

Configuring the user interface

The following commands can be sent to the ACI via the function "Command to ACI screen". The commands can be assigned to the F-panel for an ACI screen (A1), an ACI control (A2) or/and a control without ACI interface (A3).

- Exit Control Panel = ("Exit_Control_Panel") Exits the focusing of the active screen segment and shows the F-panel belonging to the ACI screen (A2, A3)
- Next Control = ("Next_Control") Goes to the next control and shows its F-panel within the screen segment (A2, A3)
- Next Window = ("Set_ActiveFocus") Changes the focus to the next screen segment in the order "from left to right - from top to bottom" and shows the F-panel of the control displayed in the screen segment (A2, A3)
- Set Active Focus = ("Set_ActiveFocus") Focuses the most recently active screen segment and shows the F-panel of the control displayed in the screen segment (A1)
- Show Option Dialog = ("Show_OptionDialog") Shows the "Options" dialog and/or the PropertyPages of the control if this is implemented in the control (A2, A3)
- Zoom + = ("Zoom_In") Zooms in the active screen segment to the complete screen width and height (A2, A3)
- Zoom - = ("Zoom_Out") Zooms out the zoomed screen segment to its original size. (A2, A3)
- Additionally, the commands "Zoom_In", "Zoom_Out", "Show_Control", "Call_CallCommand", "Call_CallCommandEx" and "SendCommand-To_..." that require additional parameters (P) can be entered.

The respective command and its individual parameters are separated by spaces:

- Zoom_In [P1 [P2 [P3 [P4]]]] = It is like +, but it provides additional options. (A2, A3)



P1=""/"Name"

F-panel is not switched / F-panel to be displayed

P2=0/1,...,n

Focused screen segment (default) / indexed screen segment

P3=-1/0/1

Not focused (default) / only focused in zoom mode (only if it is possible to be focused) / keep focus even after zoom mode (not possible if it cannot be focused)

P4=0/1

not to be focused if not configured as focusable / focusable in zoom mode even if it is not configured as focusable (forcing focus)



Parameters in []-brackets are optional. However, they have to be specified in order to use the following parameter.

- Zoom_Out [P1] = It is like a zoom -, but provides additional options. (A2, A3)

Configuring the user interface

	P1=""/Name"	Standard F-panel of the control / F-panel to be displayed
–		Show_Control P1 P2 [P3] = Switches to a specified control in a screen segment and sets/removes the screen segment focus. (A1, A2, A3)
	P1=-1/0/1,...,n P2=0/1,...,n P3=""/0/1	Next control / only screen segment to be focused / indexed control In the focused screen segment / in the indexed screen segment Keep focus unchanged (default) / remove focus / set focus
–		Call_ICallCommand <M> = Transfers the given method (M) to the focused ACI control in the ACI interface method "InvokeCommand". This method must be known to react respectively. (A2)
–		Call_ICallCommandEx <M> [<P1> <P2> ...] = Transfers the given method with the parameters to the focused ACI control in the ACI interface method "InvokeCommand" (A2)
–		SendCommandTo_All <M> [<P1> <P2> ...] = Transfers the given method without or with the parameters to all ACI controls in the ACI interface method "InvokeCommand" (A1, A2)
–		SendCommandTo_Control <ProgID> <M> [<P1> <P2> ...] = Transfers the given method without or with the parameters to the ACI controls that have the indicated ProgID in the ACI interface method "InvokeCommand" (A1, A2)
–		SendCommandTo_Instance <ProgID> <TimeID> <M> [<P1> <P2> ...] = Transfers the given method without or with the parameters to the ACI control that has the indicated ProgID and TimeID in the ACI interface method "InvokeCommand" (A1, A2)
		In order to specify the ProgID and the TimeID for a control, these have to be read from the configuration file.

Use the function "Command to ACI control" to enter the commands "Call_ICallCommand" and "Call_ICallCommandEx" (see above). The ProgID of the ACI control has to be in the front as an additional obligatory parameter. The method with or without parameters is the only control to be transferred to it.

- <ProgID> Call_ICallCommand <M> = Transfers the given method (M) to the ACI control with the specified ProgID in the ACI interface method "InvokeCommand". This method must be known to react respectively. (A1, A2)
- <ProgID> Call_ICallCommandEx <M> [<P1> <P2> ...] = Transfers the given method with the parameters to the ACI controls that have the indicated ProgID in the ACI interface method "InvokeCommand" (A1, A2)

5.14 User rights for screen and functions calls

The user management of IndraWorks Engineering can be used to disable individual screen calls and functions, allowing only authorized users to call these functions in IndraWorks Operation.

To do so, enable User Management in IndraWorks Engineering using the **Project ► User Management ► User Management Enabled** menu item. The **User Management ► Configure** menu item opens the "User Management Configuration" dialog. After the user group has been selected, the permissions for all members of the group can be configured. The "HMI" tab lists all screen and function calls which can be protected by user rights (see [fig. 5-141 "Configuration of user rights for HMI functions" on page 179](#)). The configuration of user rights can be set separately for each visualization device. Thus, the visualization device has to be selected first.

By default, all rights are enabled, i.e. every user with a valid account can execute all HMI functions. To disable selected rights, click on the entries. Some of the rights, e.g. screen calls, are combined in folders. In this case, all rights can be added or removed by the context menu of the folder.

If User Management is enabled when IndraWorks Operation is running, a user verification is performed as soon as a protected function is requested. This function is only executed if the user is provided with the right for the respective function.



User management settings is not applied to IndraWorks projects, which have been created with version V05 or lower.

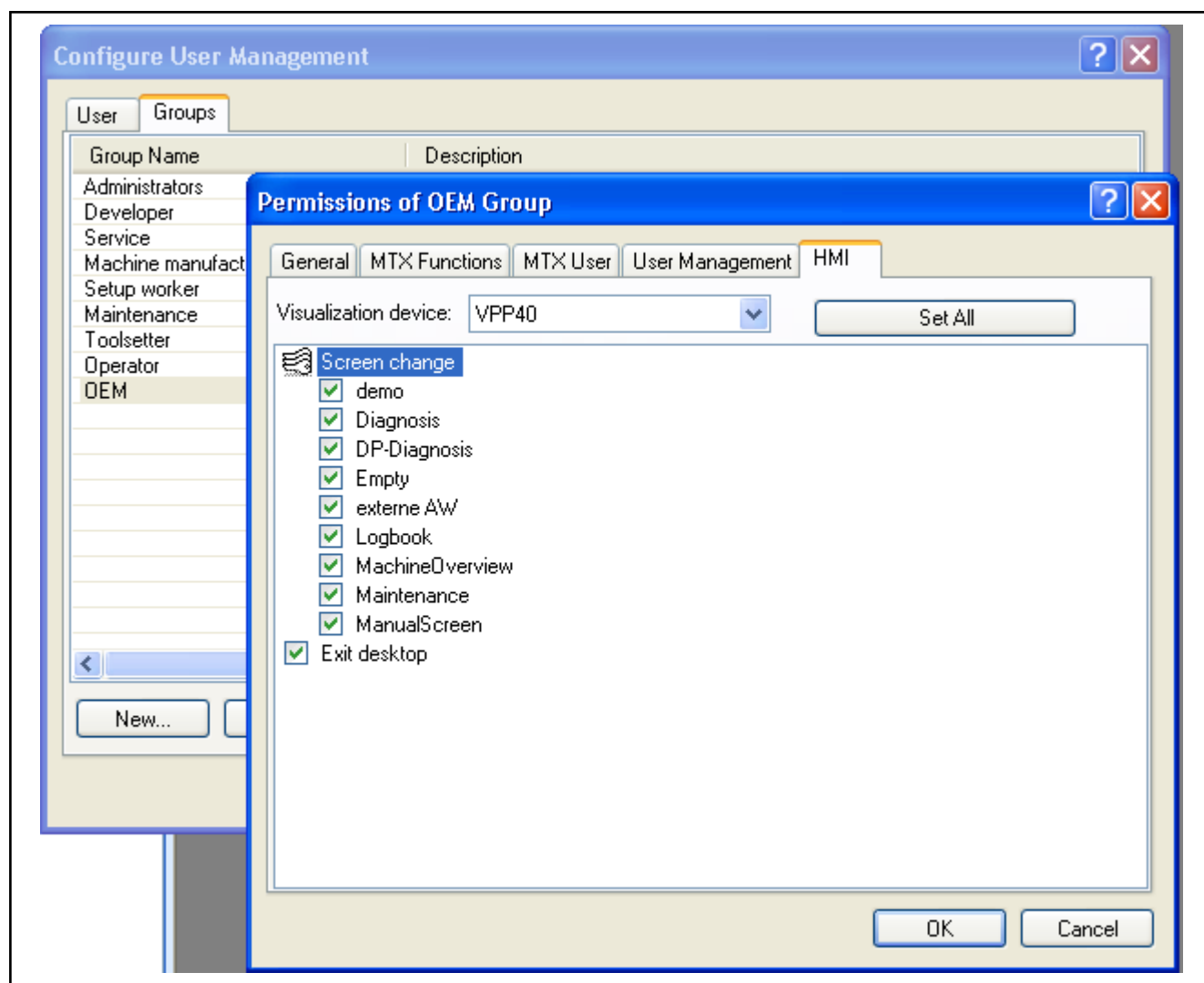


Fig. 5-141: Configuration of user rights for HMI functions

5.15 Locking and hiding F- and M-keys by the user management

OP-, F- and M-keys can be locked and hidden by the user management.

If the "Permissions" dialog of a group is opened, all configured and blocked OP-, F- and M-keys are listed in the "Hide Screen" folder via the "HMI" tab. The visibility is enabled in the presetting. That means that these keys are shown at IndraWorks Operation runtime. Clicking on the key allows to select the checkbox in front of the key name. This locks the key and it becomes invisible at IndraWorks Operation runtime.

Configuring the user interface

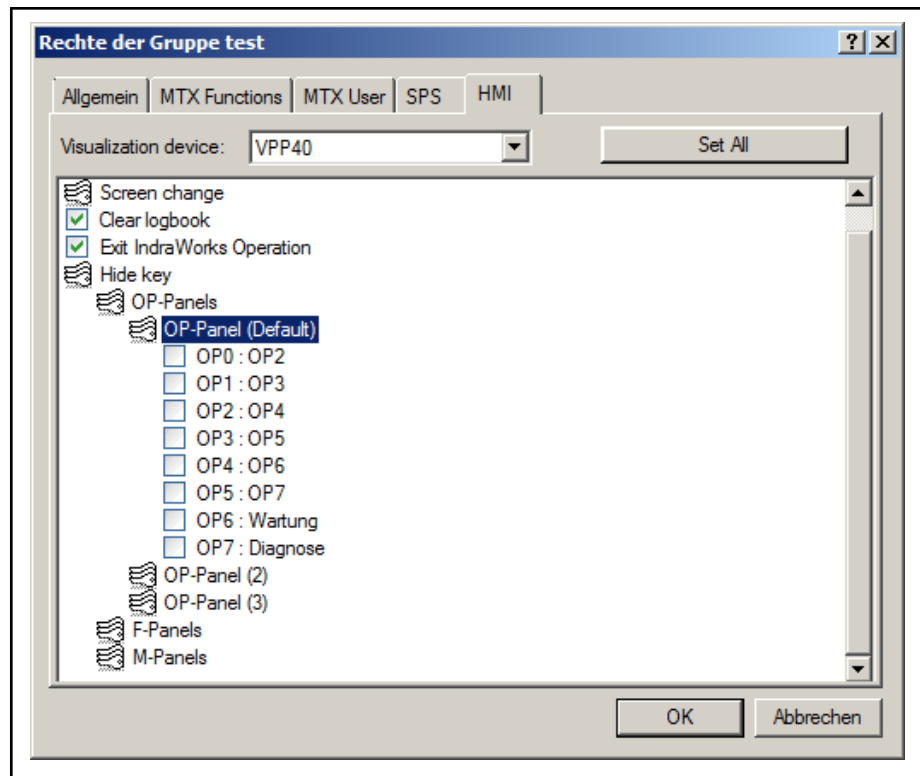


Fig. 5-142: Configuring user permissions for hiding OP-, F- and M-keys

In the "Apply key" folder, all configured and enabled OP-, F- and M-keys of all panels are listed. The executability of these keys is enabled. That means that these keys are displayed and can be displayed during IndraWorks Operation runtime. Clicking on the key allows to deselect the checkbox in front of the key name. This locks the key and cannot be executed anymore at IndraWorks Operation runtime.

Configuring the user interface

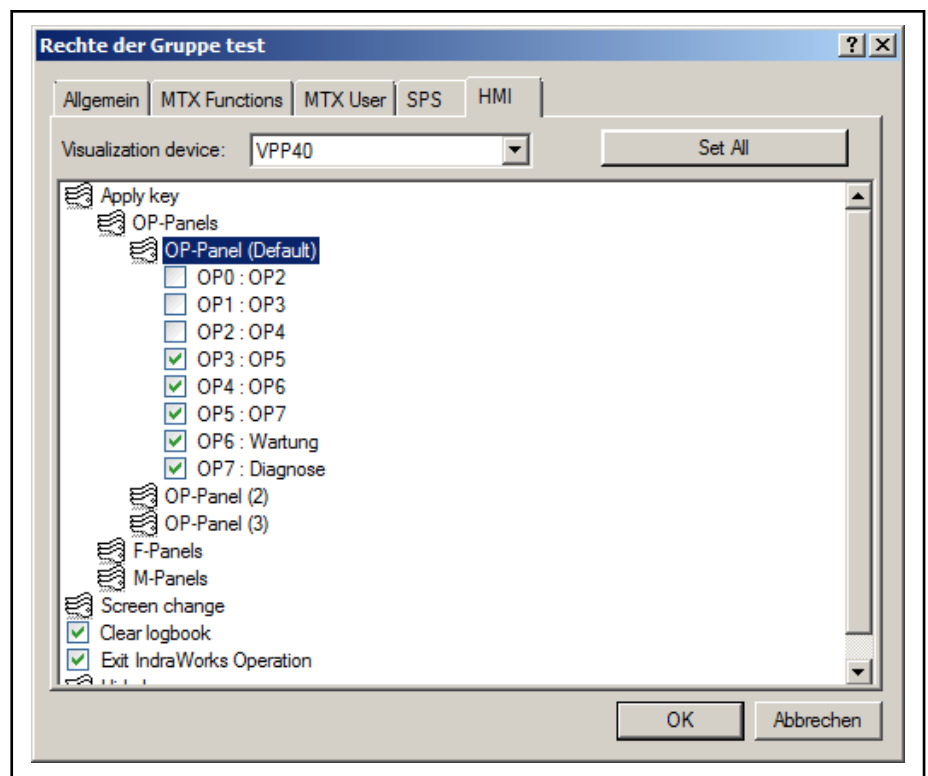


Fig. 5-143: Configuring user permissions for locking OP-, F- and M-keys

Lock and hide M-keys in the operating screen for multi touch user interface by user management

All M-keys in operating screens for the multi touch user interface are listed in the "Hide key" folder. The executability of these keys is enabled. That means that these keys are displayed at IndraWorks Operation runtime. Clicking on the key allows to select the checkbox in front of the key name. This locks the key and it becomes invisible at IndraWorks Operation runtime.

Configuring the user interface

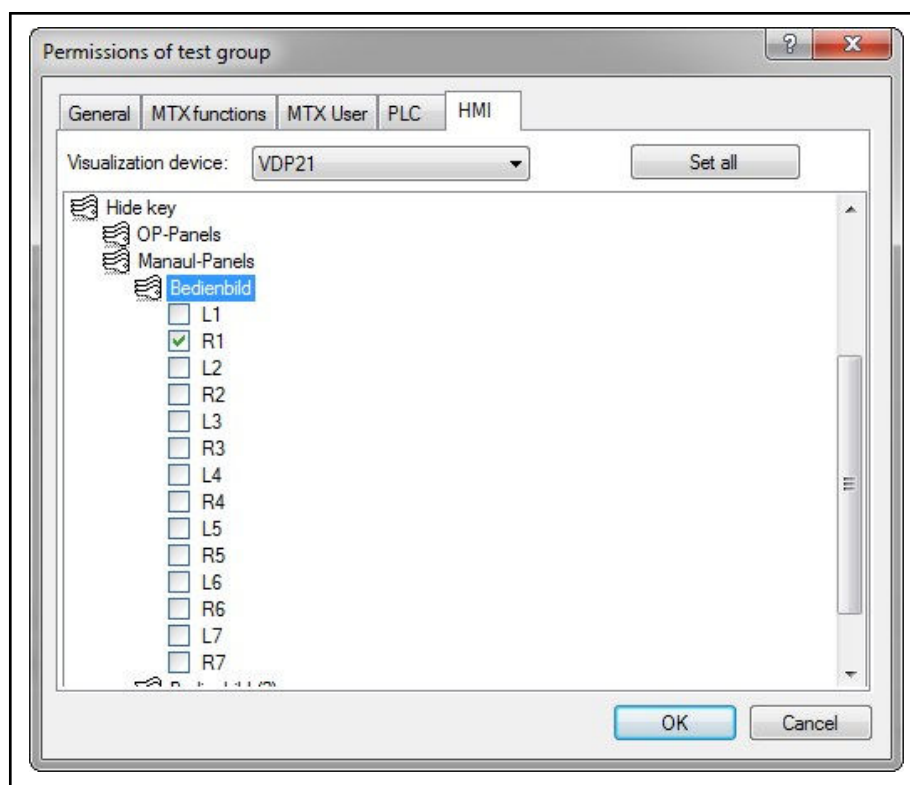


Fig. 5-144: Configuring user permissions for hiding M-keys

All M-keys in operating screens for the multi touch user interface are listed in the "Execute key" folder. The executability is enabled in the presetting. That means that these keys can be executed at IndraWorks Operation runtime. Clicking on the key allows to deselect the checkbox in front of the key name. This locks the key and cannot be executed anymore at IndraWorks Operation runtime.

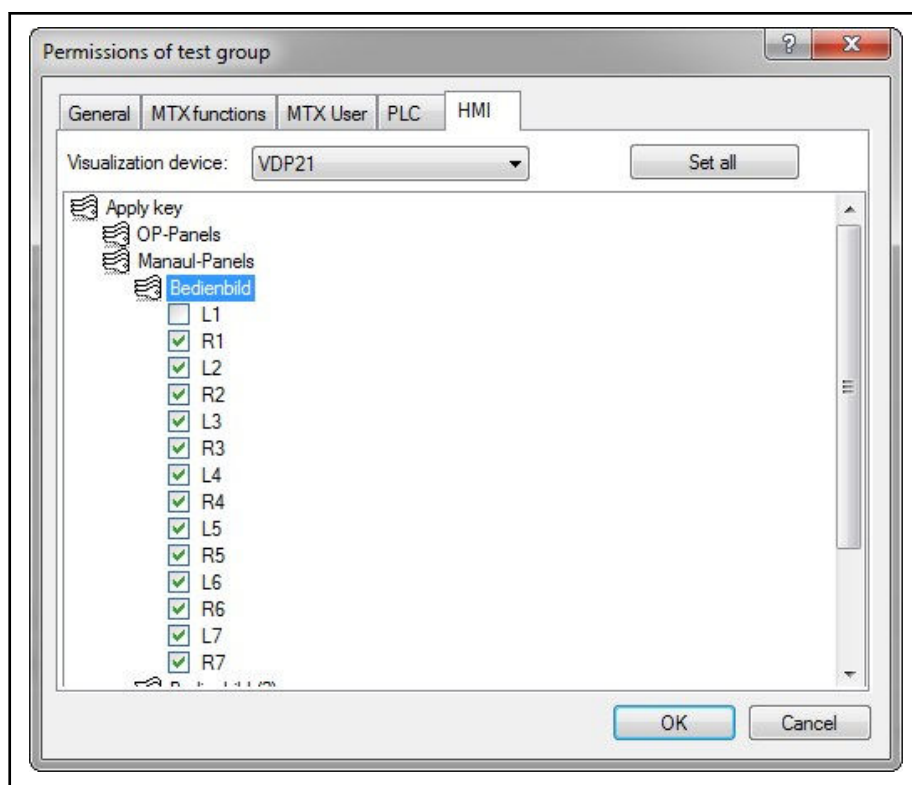


Fig. 5-145: Configuring user permissions for locking M-keys for multi touch user interface

5.16 Converting configuration data of old IndraWorks versions

Projects created with the IndraWorks version 04VRS or older have to be converted since they cannot be used immediately for IndraWorks Operation due to modified data formats. Start IndraWorks Engineering and open the project to be converted. IndraWorks Engineering converts the data to the new format. The project can be edited and enabled.

Projects that were created with IndraWorks version 13V02 or higher are no longer converted to the installed IndraWorks version when they are opened as it used to be the case. But they may be edited in the original version. If it is necessary to use a higher visualization data version, e.g. to use new functions or elements, the compatibility mode of the project has to be set accordingly first. Then, change the device version via the "Properties" dialog of the visualization device. The information, warnings and errors of the conversions carried out are displayed in the message window.

5.17 Configuring VCP devices

5.17.1 Overview

Version IndraWorks HMI 06VRS or higher supports the VCP devices.



The integration is limited to the VCP xx.2 device generation and VI-Composer 02VRS. The installation of the VI-Composer is not part of the IndraWorks installation. It is provided on a separate installation CD.

Configuring the user interface

5.17.2 Inserting VCP devices

Add the VCP device to the IndraWorks project from the device library.

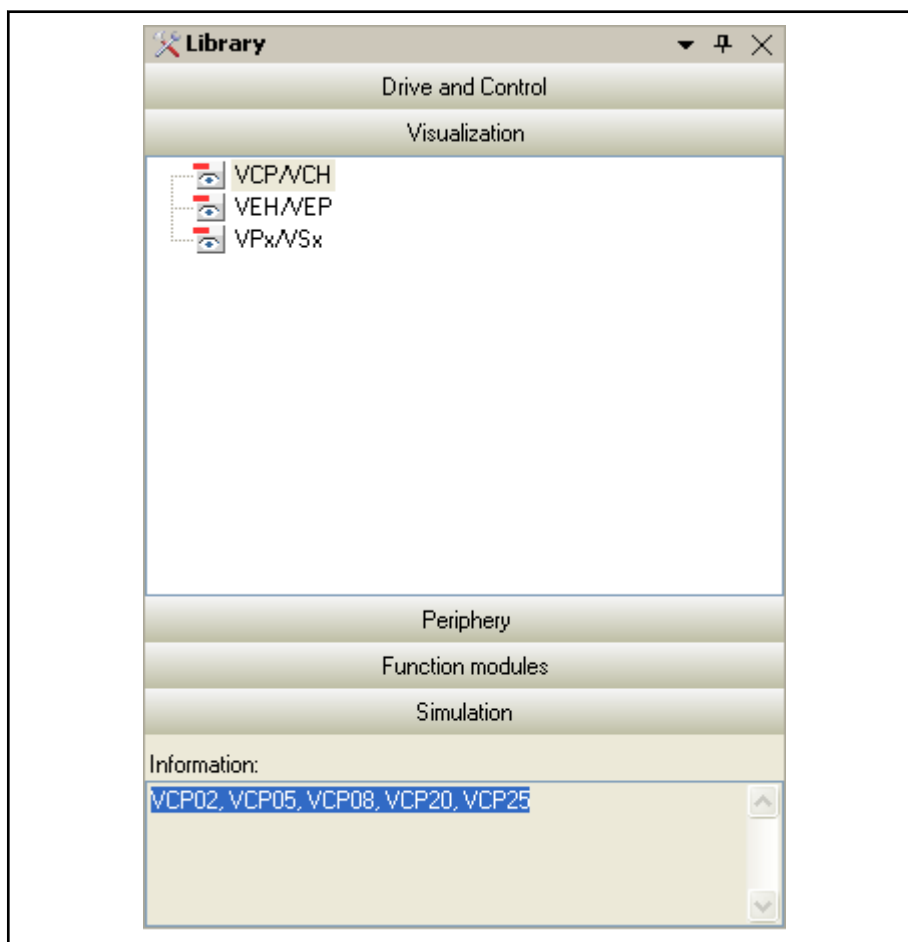


Fig. 5-146: VCP devices in the device library

The device and its properties can be defined using the HMI wizard.

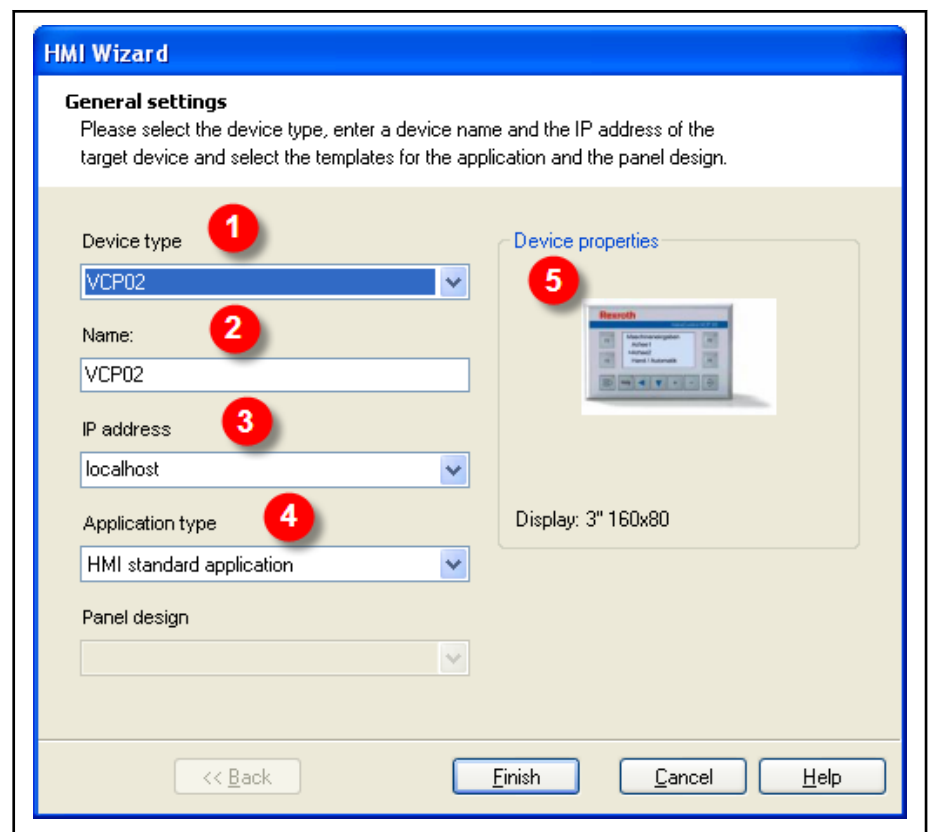


Fig. 5-147: HMI wizard

Device type ① Available device types in the device group selected: Select the desired device type. The "Device properties" field (⑤) shows a preview of the device selected.

Name ② Name of the visualization device used to display the device in the Project Explorer. The name has to be unique, i.e. a warning appears if the project already contains a device with the same name.

IP address ③ IP address of the visualization device. The IP address has to be unique within the project.

Application type ④ Configuration template for the visualization device. The application type templates vary according to the preconfigured screens and settings of the VCP device. The default application type is the "HMI standard application". Templates can also be created for individual application types that can be used as initial configuration.

Complete the settings with **Finish**. The wizard-guided basic settings are completed. The new VCP device appears in the Project Explorer. The settings can be changed subsequently (see [chapter 5.1.3 "Context menu of the visualization device" on page 53](#)).

5.17.3 Calling the VI-Composer

Open the VI-Composer by double-clicking on the "VI-Composer" subnode. Create and compile the application directly in the VI-Composer.

Configuring the user interface

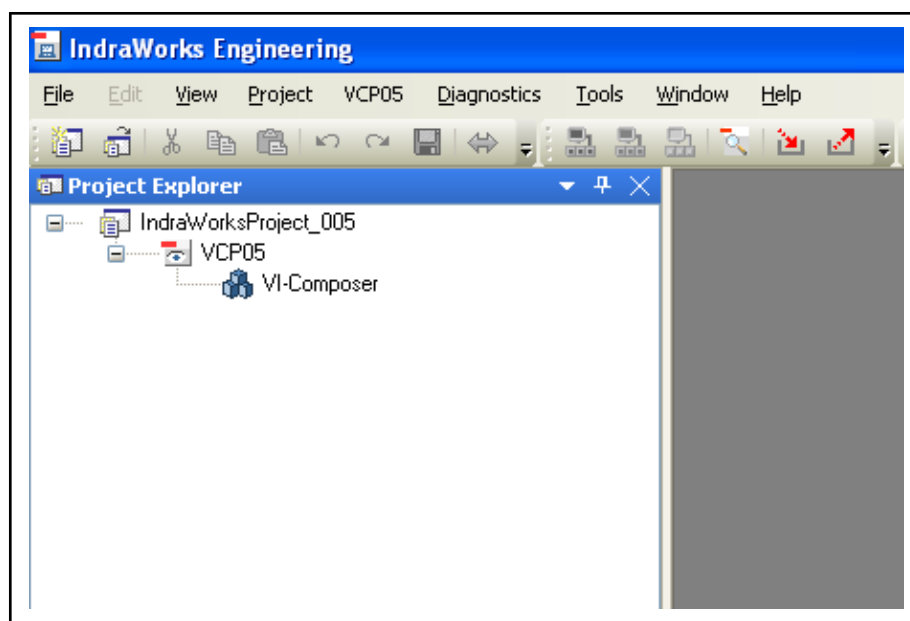


Fig. 5-148: VI-Composer in the Project Explorer

5.17.4 Transferring project files

The context menu of the VCP device can be used to transfer (FTP download) the previously compiled project files to the device.

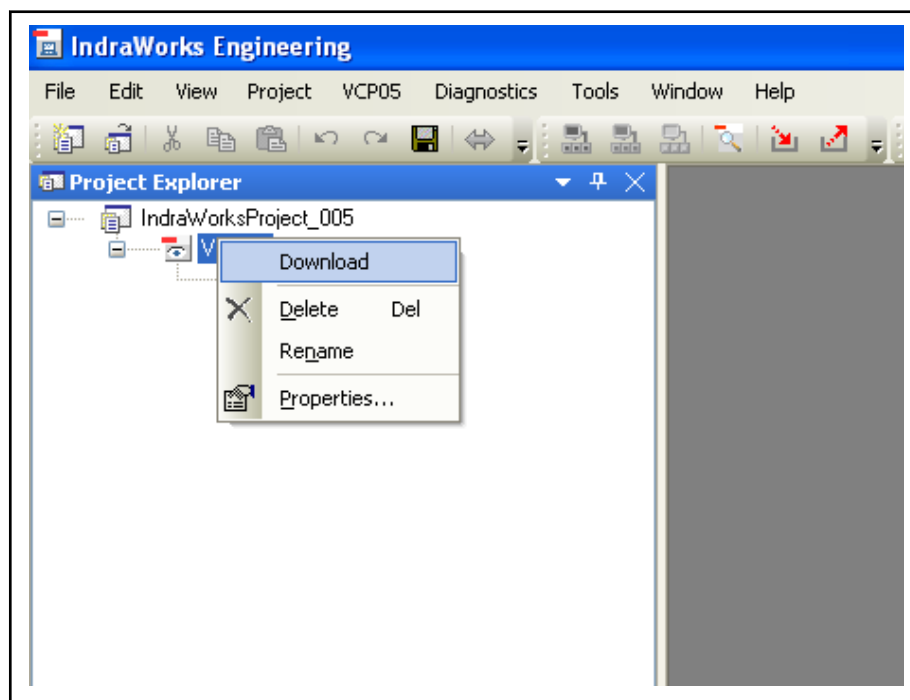


Fig. 5-149: FTP download



Archived project files can also be downloaded without VI-Composer installed software.

5.18 Configuring the Header Display

5.18.1 Overview

IndraWorks HMI provides a preconfigured header display.

The header provides the following information:

- A ProVi error message is pending
- A ProVi warning message is pending
- A ProVi notice is pending
- Control name
- Wind showing/hidden
- Control connection status
- Network status

In addition, the header provides two text fields and six icon fields for application-specific information which can be configured as desired.

The content and layout of the header are contained in the "HMI.Header.ML" file. If a visualization device is added to an IndraWorks project, this file is generated and saved to the project directory with a default configuration. Edit the configuration file and adapt it to your requirements using a text editor. The configuration file is contained in the following folder: <Project directory>\<Name of visualization device>\user\config\HMI.Header.xml.

Some of the visualization devices differ in their screen resolution and, thus, in the available header size. This is taken into account in different configuration files.

IndraWorks projects, which have been created with version 05 or lower, already contain a configuration file which provides the full functionality described above. If necessary, the existing configuration file has to be replaced by the latest version. This version is located in the following folder

IndraWorks\Library\Devices\Visualization\.ApplicationTemplates\<Device Type>\HMI.

Configuring the user interface

```

<?xml version="1.0" encoding="utf-8"?>
<!-- configuration file for the devices with resolution 1024x768-->
<!-- Please add the configuration only in the Controls with the comment -->
<Header ReferenceWidth = "805">
  <Control Left="585" Top="0" Width="1" Height="75" BackColor="#808080"/>
  <Control Left="0" Top="0" Width="585" Height="37" BackColor="#D3D3D3"/>
  <Control Left="10" Top="6" Width="77" Height="25" Assembly="HMI.Header" Type="HMI.Header.HeaderDiagnosisControl"/>
  <Control Left="98" Top="6" Width="1" Height="25" BackColor="#808080"/>
  <Control Left="102" Top="6" Width="25" Height="25" Assembly="HMI.Header" Type="HMI.Header.HeaderNetworkStatusControl"/>
  <Control Left="128" Top="6" Width="25" Height="25" Assembly="HMI.Header" Type="HMI.Header.HeaderIRemoteStatusControl"/>
  <Control Left="154" Top="6" Width="1" Height="25" BackColor="#808080"/>
  <Control Left="156" Top="6" Width="297" Height="25" Assembly="HMI.Header" Type="HMI.Header.HeaderDeviceControl"/>
  <Control Left="454" Top="6" Width="1" Height="25" BackColor="#808080"/>
  <Control Left="455" Top="6" Width="25" Height="25" Assembly="HMI.Header" Type="HMI.Header.HeaderFreeIconControl">
  <!-- Please add the configuration hier -->
  </Control>
  <Control Left="482" Top="6" Width="25" Height="25" Assembly="HMI.Header" Type="HMI.Header.HeaderFreeIconControl">
  <!-- Please add the configuration hier -->
  </Control>
  <Control Left="508" Top="6" Width="25" Height="25" Assembly="HMI.Header" Type="HMI.Header.HeaderRTProjectStatusControl"/>
  <Control Left="533" Top="6" Width="25" Height="25" Assembly="HMI.Header" Type="HMI.Header.HeaderWinLockStatusControl"/>
  <Control Left="559" Top="6" Width="25" Height="25" Assembly="HMI.Header" Type="HMI.Header.HeaderConnectionControl"/>
  <Control Left="0" Top="38" Width="585" Height="37" BackColor="#A9A9A9"/>
  <Control Left="0" Top="37" Width="585" Height="1" BackColor="#808080"/>
  <Control Left="208" Top="38" Width="271" Height="37" BackColor="#666666"/>
  <Control Left="154" Top="38" Width="53" Height="37" BackColor="#D3D3D3"/>
  <Control Left="0" Top="38" Width="153" Height="37" BackColor="#808080" ForeColor="white" Assembly="HMI.Header" Type="HMI.Header.HeaderFreeTextControl">
  <!-- Please add the configuration hier -->
  </Control>
  <Control Left="155" Top="44" Width="51" Height="25" BackColor="#808080" Assembly="HMI.Header" Type="HMI.Header.HeaderOperationModeControl"/>
  <Control Left="208" Top="38" Width="271" Height="37" BackColor="#808080" ForeColor="white" Assembly="HMI.Header" Type="HMI.Header.HeaderFreeTextControl">
  <!-- Please add the configuration hier -->
  </Control>
  <Control Left="480" Top="38" Width="105" Height="37" BackColor="#D3D3D3"/>
  <Control Left="481" Top="44" Width="25" Height="25" Assembly="HMI.Header" Type="HMI.Header.HeaderFreeIconControl">
  <!-- Please add the configuration hier -->
  </Control>
  <Control Left="507" Top="44" Width="25" Height="25" Assembly="HMI.Header" Type="HMI.Header.HeaderFreeIconControl">
  <!-- Please add the configuration hier -->
  </Control>
  <Control Left="533" Top="44" Width="25" Height="25" Assembly="HMI.Header" Type="HMI.Header.HeaderFreeIconControl">
  <!-- Please add the configuration hier -->
  </Control>
  <Control Left="559" Top="44" Width="25" Height="25" Assembly="HMI.Header" Type="HMI.Header.HeaderFreeIconControl">
  <!-- Please add the configuration hier -->
  </Control>
  <Control Left="0" Top="75" Width="806" Height="1" BackColor="#808080"/>
  <Control Left="0" Top="76" Width="806" Height="37" BackColor="#DCDCDC"/>
  <Control Left="0" Top="76" Width="670" Height="18" Assembly="HMI.Header" Type="HMI.Header.HeaderProViControl">
  <Configuration>
  <Visu ToggleTime="3000" ToggleAll="true"/>
  <Visu Item = "Error" Display = "true" TextColor="#666666" BackColor="#FFB9B9"/>
  <Visu Item = "Warning" Display="false" TextColor="#666666" BackColor="#FFFB99"/>
  <Visu Item = "diagnosis" Display = "false" TextColor="#666666" BackColor="white"/>
  <Visu Item = "Tip" Display="false" TextColor="#666666" BackColor="#B9FFB9"/>
  <Visu Item = "StartPrerequisite" Display="false" TextColor="#666666" BackColor="#ADD8E6"/>
  <Visu Item = "SfcError" Display = "true" TextColor="#666666" BackColor="#FFB9B9"/>
  <Visu Item = "Sfcdiagnosis" Display = "false" TextColor="#666666" BackColor="white"/>
  <Visu Item = "SfcTip" Display="false" TextColor="#666666" BackColor="#B9FFB9"/>
  <Visu Item = "SfcStartPrerequisite" Display="false" TextColor="#666666" BackColor="#ADD8E6"/>
  </Configuration>
  </Control>
  <Control Left="0" Top="94" Width="805" Height="1" BackColor="#808080"/>
  <Control Left="0" Top="95" Width="805" Height="18" Assembly="HMI.Header" Type="HMI.Header.HeaderProViControl">
  <Configuration>
  <Visu ToggleTime="2000" ToggleAll="true"/>
  <Visu Item = "Error" Display = "false" TextColor="#666666" BackColor="#FFB9B9"/>
  <Visu Item = "Warning" Display="true" TextColor="#666666" BackColor="#FFFB99"/>
  <Visu Item = "diagnosis" Display = "true" TextColor="#666666" BackColor="white"/>
  <Visu Item = "Tip" Display="true" TextColor="#666666" BackColor="#B9FFB9"/>
  <Visu Item = "StartPrerequisite" Display="true" TextColor="#666666" BackColor="#ADD8E6"/>
  <Visu Item = "SfcError" Display = "false" TextColor="#666666" BackColor="#FFB9B9"/>
  <Visu Item = "Sfcdiagnosis" Display = "false" TextColor="#666666" BackColor="white"/>
  <Visu Item = "SfcTip" Display="false" TextColor="#666666" BackColor="#B9FFB9"/>
  <Visu Item = "SfcStartPrerequisite" Display="false" TextColor="#666666" BackColor="#ADD8E6"/>
  </Configuration>
  </Control>
  <Control Left="669" Top="76" Width="71" Height="18" BackColor="#DCDCDC" ForeColor="#666666" Type="Date"/>
  <Control Left="741" Top="76" Width="60" Height="18" BackColor="#DCDCDC" ForeColor="#666666" Type="Time" SynchronizeToPfc="False" SyncRate="3600" />
  <Control Left="740" Top="79" Width="1" Height="12" BackColor="#666666"/>
  <Control Left="710" Top="8" Width="74" Height="17" Image="%config%\HMI.Logo.bmp"/>
  <!-- Please add the configuration for Texttokens hier -->
</Header>

```

Fig. 5-150: Configuration file for a resolution of 1024x768 pixels

Configuring the user interface

```
<?xml version="1.0" encoding="utf-8"?>
<!-- configuration file for the devices with resolution 800x600 -->
<!-- Please add the configuration only in the Controls with the comment -->
<Header ReferenceWidth = "618">
  <Control Left="0" Top="0" Width="525" Height="30" BackColor="#D3D3D3"/>
  <Control Left="0" Top="31" Width="525" Height="1" BackColor="#808080"/>
  <Control Left="98" Top="3" Width="1" Height="25" BackColor="#808080"/>
  <Control Left="10" Top="3" Width="77" Height="25" Assembly="HMI.Header" Type="HMI.Header.HeaderDiagnosisControl"/>
  <Control Left="102" Top="3" Width="25" Height="25" Assembly="HMI.Header" Type="HMI.Header.HeaderNetworkStatusControl"/>
  <Control Left="128" Top="3" Width="25" Height="25" Assembly="HMI.Header" Type="HMI.Header.HeaderRemoteStatusControl"/>
  <Control Left="155" Top="3" Width="1" Height="25" BackColor="#808080"/>
  <Control Left="157" Top="3" Width="235" Height="25" Assembly="HMI.Header" Type="HMI.Header.HeaderDeviceControl"/>
  <Control Left="397" Top="3" Width="1" Height="25" BackColor="#808080"/>
  <Control Left="395" Top="3" Width="25" Height="25" Assembly="HMI.Header" Type="HMI.Header.HeaderFreeIconControl"/>
  <!-- Please add the configuration hier -->
</Control>
<Control Left="421" Top="3" Width="25" Height="25" Assembly="HMI.Header" Type="HMI.Header.HeaderFreeIconControl"/>
  <!-- Please add the configuration hier -->
</Control>
<Control Left="447" Top="3" Width="25" Height="25" Assembly="HMI.Header" Type="HMI.Header.HeaderRTProjectStatusControl"/>
<Control Left="472" Top="3" Width="25" Height="25" Assembly="HMI.Header" Type="HMI.Header.HeaderWinLockStatusControl"/>
<Control Left="498" Top="3" Width="25" Height="25" Assembly="HMI.Header" Type="HMI.Header.HeaderConnectionControl"/>
<Control Left="525" Top="0" Width="1" Height="62" BackColor="#808080"/>
<Control Left="0" Top="32" Width="525" Height="30" BackColor="#A9A9A9"/>
<Control Left="0" Top="62" Width="620" Height="1" BackColor="#808080"/>
<Control Left="208" Top="31" Width="210" Height="32" BackColor="#666666"/>
<Control Left="0" Top="32" Width="155" Height="30" BackColor="#808080" Assembly="HMI.Header" Type="HMI.Header.HeaderFreeTextControl"/>
  <!-- Please add the configuration hier -->
</Control>
<Control Left="156" Top="32" Width="52" Height="30" BackColor="#D3D3D3"/>
<Control Left="157" Top="35" Width="51" Height="25" BackColor="#808080" Assembly="HMI.Header" Type="HMI.Header.HeaderOperationModeControl"/>
<Control Left="209" Top="32" Width="209" Height="30" BackColor="#808080" ForeColor="white" Assembly="HMI.Header" Type="HMI.Header.HeaderFreeTextControl"/>
  <!-- Please add the configuration hier -->
</Control>
<Control Left="419" Top="32" Width="106" Height="30" BackColor="#D3D3D3"/>
<Control Left="420" Top="35" Width="25" Height="25" Assembly="HMI.Header" Type="HMI.Header.HeaderFreeIconControl"/>
  <!-- Please add the configuration hier -->
</Control>
<Control Left="446" Top="35" Width="25" Height="25" Assembly="HMI.Header" Type="HMI.Header.HeaderFreeIconControl"/>
  <!-- Please add the configuration hier -->
</Control>
<Control Left="472" Top="35" Width="25" Height="25" Assembly="HMI.Header" Type="HMI.Header.HeaderFreeIconControl"/>
  <!-- Please add the configuration hier -->
</Control>
<Control Left="498" Top="35" Width="25" Height="25" Assembly="HMI.Header" Type="HMI.Header.HeaderFreeIconControl"/>
  <!-- Please add the configuration hier -->
</Control>
<Control Left="0" Top="63" Width="620" Height="35" BackColor="#DCDCDC"/>
<Control Left="0" Top="63" Width="490" Height="17" Assembly="HMI.Header" Type="HMI.Header.HeaderProViControl"/>
<Configuration>
  <Visu ToggleTime="3000" ToggleAll="true"/>
  <Visu Item = "Error" Display = "true" TextColor="#666666" BackColor="#FFB9B9"/>
  <Visu Item = "Warning" Display="false" TextColor="#666666" BackColor="#FFB9B9"/>
  <Visu Item = "diagnosis" Display = "false" TextColor="#666666" BackColor="white"/>
  <Visu Item = "Tip" Display="false" TextColor="#666666" BackColor="#B9FFB9"/>
  <Visu Item = "StartPrerequisite" Display="false" TextColor="#666666" BackColor="#ADD8E6"/>
  <Visu Item = "SfcError" Display = "true" TextColor="#666666" BackColor="#FFB9B9"/>
  <Visu Item = "Sfcdiagnosis" Display = "false" TextColor="#666666" BackColor="white"/>
  <Visu Item = "SfcTip" Display="false" TextColor="#666666" BackColor="#B9FFB9"/>
  <Visu Item = "SfcStartPrerequisite" Display="false" TextColor="#666666" BackColor="#ADD8E6"/>
</Configuration>
</Control>
<Control Left="0" Top="80" Width="618" Height="1" BackColor="#808080"/>
<Control Left="0" Top="81" Width="618" Height="17" Assembly="HMI.Header" Type="HMI.Header.HeaderProViControl"/>
<Configuration>
  <Visu ToggleTime="2000" ToggleAll="true"/>
  <Visu Item = "Error" Display = "false" TextColor="#666666" BackColor="#FFB9B9"/>
  <Visu Item = "Warning" Display="true" TextColor="#666666" BackColor="#FFB9B9"/>
  <Visu Item = "diagnosis" Display = "true" TextColor="#666666" BackColor="white"/>
  <Visu Item = "Tip" Display="true" TextColor="#666666" BackColor="#B9FFB9"/>
  <Visu Item = "StartPrerequisite" Display="true" TextColor="#666666" BackColor="#ADD8E6"/>
  <Visu Item = "SfcError" Display = "false" TextColor="#666666" BackColor="#FFB9B9"/>
  <Visu Item = "Sfcdiagnosis" Display = "false" TextColor="#666666" BackColor="white"/>
  <Visu Item = "SfcTip" Display="false" TextColor="#666666" BackColor="#B9FFB9"/>
  <Visu Item = "SfcStartPrerequisite" Display="false" TextColor="#666666" BackColor="#ADD8E6"/>
</Configuration>
</Control>
<Control Left="488" Top="63" Width="72" Height="17" BackColor="#DCDCDC" ForeColor="#666666" Type="Date"/>
<Control Left="560" Top="63" Width="62" Height="17" BackColor="#DCDCDC" ForeColor="#666666" Type="Time" SynchronizeToPic="False" SyncRate="3600" />
<Control Left="560" Top="65" Width="1" Height="12" BackColor="#666666"/>
<Control Left="540" Top="8" Width="74" Height="17" Image="config\HMI.Logo.bmp"/>
  <!-- Please add the configuration for the Texttokens hier -->
</Header>
```

Fig. 5-151: Configuration file for a resolution of 800x600 pixels

5.18.2 Configuring individual texts and symbols

Texts and icons can be added to the default configuration of the header. These texts and icons can be linked to a PLC variable to show or hide them depending on the state. In addition, configure the user texts language-de-

Configuring the user interface

Configuring user texts

pendently, so that the header displays the texts in the language selected if the language is changed in IndraWorks Operation.

The configuration file is prepared to insert user texts and has to be edited accordingly. Search the following lines in the file:

```
<Control... Type="HMI.Header.HeaderFreeTextControl">
```

```
<!-- "Please add the configuration here" -->
```

```
</Control>
```

Replace the comment line in the middle by the following configuration entries.

```
<Configuration VariableName = "Variable name">
```

```
<Visu Value="Value 1" TextToken="Token 1" blink="Boolean value" TextColor="Color name" />
```

```
<Visu Value="Value n" TextToken="Token n" blink="Boolean value" TextColor="Color name" />
```

```
</Configuration>
```

The placeholder have the following meaning:

- **Variable name**
Name of a PLC variable (type BO or INT) to be used for manipulating the text. The variable refers to the control assigned to the visualization device
- **Value**
Numeric value of the variable. If this value is provided, the "Token" text is displayed with the color "Color name" and flashing if necessary. Insert any number of lines according to this format for different variable values

Example:

```
<Configuration VariableName = ".Test">
```

```
<Visu Value="0" TextToken="MachineOn" blink="TRUE" TextColor="Green" />
```

```
<Visu Value="1" TextToken="MachineOff" blink="FALSE" TextColor="Red" />
```

```
<Visu Value="2" TextToken="MachinePause" blink="FALSE" TextColor="Yellow" />
```

```
</Configuration>
```

Texts for different languages can be implemented by inserting a token name. The respective texts can be configured by inserting the following section:

```
<TextToken Name="token name">
```

```
<Constant Language="Language-culture 1">Text 1</Constant >
```

```
<Constant Language="Language-culture n">Text n</Constant >
```

```
</TextToken>
```

- **Token name**
Name of the token
- **Language culture**
Language and culture to which the text is allocated, e.g.
de-DE for German language, German culture
en-US for English language, American culture

Example:

Configuring the user interface

```
<TextToken Name="MachineOn" >
<Constant Language = "">Open</Constant >
<Constant Language = "de-DE">Open</Constant >
<Constant Language = "en-US">Öffnen</Constant >
</TextToken>
<TextToken Name="MachineOff" >
<Constant Language = "de-DE">Schließen</Constant >
<Constant Language = "en-US">Close</Constant >
</TextToken>
```

Configuring icons

User icons are configured such as user texts. The following sections are provided for the configuration of icons: **"HMI.Header.HeaderFreelconControl"**

```
<Control... Type="HMI.Header.HeaderFreelconControl">
<!-- "Please add the configuration here" -->
</Control>
```

To display an icon in the header, replace the comment line with the following configuration lines:

```
<Configuration VariableName = "Variable name">
<Visu Value="Value 1" blink="Boolean value" BackColor="Color name" Icon-
Name="File name 1" />
<Visu Value="Value n" blink="Boolean value" BackColor="Color name" Icon-
Name="File name n" />
</Configuration>
```

- **Variable name**
Name of a PLC variable (of type BOOL or INT) to control the icon display
- **Value**
Numeric value if the PLC variable applies this value, the following settings are enabled
- **Boolean value**
Defines whether the icon should be blinking
- **Color name**
Background color of the icon
- **File name**
Name of the file containing the icon to be displayed.
The path has to be entered as well. There are the following options:
 - Complete file path
 - Path relative to the IndraWorks directory
 - Use of predefined placeholders
 %configure% corresponds to ..\IndraWorks\configure
 %ActiveHMIConfig% corresponds to <Project directory>\<Visualization device>\user\config

It is recommended to copy your image files to the directory specified by %ActiveHMIConfig%, as this ensures the added files to be saved when the project is archived.

Configuring the user interface

With an IndraWorks project with the name "Project1", located under "My Documents" and a visualization with the name "VPP40", the image files have to be copied to C:\Documents and Settings\<UserName>\My Documents\Project1\VPP40\user\config.

Example:

```
<Control Left="559" Top="44" Width="25" Height="25" Assembly="HMI.Header" Type="HMI.Header.HeaderFreelconControl">
<Configuration VariableName = ".Test">
<Visu Value="0" blink="True" BackColor="Red" IconName="%config%\On.bmp" />
<Visu Value ="1" blink="False" BackColor="Green" IconName ="%config%\Off.bmp" />
</Configuration>
</Control>
```

5.18.3 Configuring for alternating display of messages

There are two lines in the header that are provided to display messages. The latest and existing messages can be alternatively displayed. The switching times, message types, text colors and background colors of the message texts can be configured via the configuration file "HMI.Header.xml".

Search in the configuration file for "HMI.Header.HeaderProViControl". This text exists twice. The first "HeaderProViControl" corresponds to the first line in the header, the second "HeaderProViControl" corresponds to the second line. Already entered configuration texts for one line are structured as follows:

```
<Configuration> <Visu ToggleTime="3000" ToggleAll="True"></Visu>
<Visu Item="Error" Display="True" TextColor="#666666" BackColor="#FFB9B9"></Visu>
<Visu Item="Warning" Display="False" TextColor="#666666" BackColor="#FEFFB9"> </Visu>
<Visu Item="Diagnosis" Display="False" TextColor="#666666" BackColor="White"></Visu>
<Visu Item="Tip" Display="False" TextColor="#666666" BackColor="#B9FFB9"> </Visu>
<Visu Item="StartPrerequisite" Display="False" TextColor="#666666" BackColor="#ADD8E6"></Visu>
<Visu Item="SfcError" Display="True" TextColor="#666666" BackColor="#FFB9B9"></Visu>
<Visu Item="Sfcdiagnosis" Display="False" TextColor="#666666" BackColor="White"></Visu>
<Visu Item="SfcTip" Display="False" TextColor="#666666" BackColor="#B9FFB9"> </Visu>
<Visu Item="SfcStartPrerequisite" Display="False" TextColor="#666666" BackColor="#ADD8E6"></Visu>
</Configuration>
```

- ToggleTime: Switching time in ms, preset for the first line: 3000 ms; preset for the second line: 2000ms.
- ToggleAll: Specifies whether all messages or only the latest message are alternately displayed. If this setting is either TRUE or does not exist, all messages can be displayed. If this setting is FALSE, only the latest messages are displayed

- Configuration for individual message types, such as in line `<Visu Item="Error" Display="True" TextColor="#666666" BackColor="#FFB9B9"></Visu>`
 - Visu item: Message type, 9 message types are available.
These are:
 - ProVi errors (Error)
 - ProVi warnings (Warning)
 - Setup diagnostics (diagnosis)
 - ProVi tips
 - Start prerequisites (StartPrerequisite)
 - SFC error (SfcError)
 - SFC set-up diagnostics (Sfcdiagnosis)
 - SFC information (SfcTip)
 - SFC start prerequisites (SfcStartPrerequisite)
 - Display: Specifies whether this message type is displayed
 - TextColor: Message text color of this message type
 - BackColor: Message background color of this message type

5.19 Export/import

Please refer to:
Rexroth IndraWorks 14VRS Engineering, chapter "Exporting and importing project data" (DOK-IWORKS-ENGINEE*V14-APRS-EN-P).

5.20 Comparing visualization data

Refer to the documentation "IndraWorks 14VRS Engineering", chapter "Comparing project data" (DOK-IWORKS-ENGINEE*V14-APRS-EN-P).



WinStudio and WebComposer project data can currently not yet be compared.

5.21 Validating visualization data

To ensure the expected functioning of a configuration created for a visualization device after transfer to the operating station, it has to be consistent.

Consistent visualization data is present if the following conditions are met:

- All referenced elements exist and are valid
- All PLC variables used are in the symbol configuration of the respective control

To check whether the configuration of the visualization data contains errors, the **Validate** function from the context menu of the visualization device can be executed at any time.

Configuring the user interface

ID	Description	Element	Path	Pfad
HMI-00000100	No control assigned to the device!	VPP40_1		VPP40_1
HMI-00000103	No M-key function block instance specified.	VPP40_1		VPP40_1
HMI-00000002	The "screenActivate" variable is invalid.	Screens		VPP40_1/Screens
HMI-00000002	The "screenNumberFromPlc" variable is invalid.	Screens		VPP40_1/Screens
HMI-00000002	The "screenNumber" variable is invalid.	Screens		VPP40_1/Screens
HMI-00000002	The "ggg" variable is invalid.	vv	M1	VPP40_1/Screen Frame/M-Panels/vv

Fig. 5-152: Outputting messages when validating visualization data

Outputting rule violations in the message window

All problems detected by IndraWorks are output in the message window. It displays the following information in each message:

- **Type:** This information categorizes the message into errors, warnings and information. Errors at the IndraWorks Operation runtime on the operating station can cause incorrect or unforeseeable behavior and should be recovered before transmitting visualization data. Warnings indicate problems that do not affect the runtime of IndraWorks Operation, but might cause incorrect execution of individual functions. Information indicate further inconsistencies without effect on the runtime of IndraWorks Operation
- **ID:** Error number used to see further information on cause and recovery in the table ["Overview on error numbers" on page 194](#)
- **Description:** Brief description of the message
- **Element:** Name of the element containing a problem
- **Position:** Additional information on where the problem occurs in the element (e.g. line number, key labeling)
- **Path:** Storage location of the element in the visualization device

To navigate to the point where the problem occurred, double-click on the message line. A respective editor opens in which the problem can be recovered. In some cases, an editor is not available. In this case, a dialog is displayed with further information on the recovery. After one or several problems have been recovered, call the **Validate** function again to check the correction

Checked criteria

When validating visualization data, the following criteria are checked:

- Selection of a control
- Validity of PLC variables
- Validity of elements (panels and screen content) referenced by screens
- Selection of a start screen in operating areas
- Referencing of screens and panels
- Uniqueness of screen numbers
- Uniqueness of the IP address of the visualization device
- Validity of WinStudio variables
- Validity of the configuration of WinStudio driver sheets
- Existence of unused WinStudio variables

Overview on error numbers

An ID is assigned to each discovered problem during the validation of visualization data. Using this ID, the following table can be used to gather information on cause and its recovery:

Configuring the user interface

ID	Cause	Recovery
HMI-00000001	The selected control is invalid	Open the editor of the incorrect element and select a control
HMI-00000002	The PLC variable used is invalid	Add the variable to the symbol configuration or select a valid PLC variable
HMI-00000003	The function is invalid on one of the F-,M- or OP-keys	Open the editor of the incorrect element and select a function from the function list
HMI-00000004	The stated parameter of a function is invalid	Open the editor of the faulty element and select a valid parameter from the parameter list or edit it
HMI-00000100	No control is selected in the properties of the visualization device	Open the "Properties" dialog of the visualization device and select a control in the "Communication" tab. If required, add a control to the project before
HMI-00000101	The name of the visualization device is invalid or not unique	Open the "Properties" dialog of the visualization device and enter a new, unique name
HMI-00000102	The IP address of the visualization device is invalid or not unique	Open the "Properties" dialog of the visualization device and enter a new, unique IP address
HMI-00000103	No instance of the MKEY function block is entered in the properties of the visualization device. This is required for the communication of the M-keys	Add an instance of the "MKEYS" function block to the PLC program and include it in the symbol configuration. Then, enter the instance name into the properties of the visualization device
HMI-00000104	The option for automatic transmission of diagnostic data is deactivated.	Open the properties dialog of the visualization device and activate the option "Automatically transmit diagnostic data" in the "Visualization data" tab (see "Download options tab" on page 56)
HMI-00000110	Error when transmitting the visualization data	Open the "Properties" dialog of the visualization device and check the IP address
HMI-00000111	The visualization data have been transmitted to the operating station successfully	
HMI-00000112	Error during visualization data upload from the operating station	Open the "Properties" dialog of the visualization device and check the IP address
HMI-00000113	The visualization data upload from the operating station was completed successfully	
HMI-00000200	The screen number of a screen is not unique	Open the screen editor and change the screen number
HMI-00000201	The screen type of the screen is invalid	Open the screen editor and select a screen type from the selection list
HMI-00000202	A referenced element of the screen is invalid	Open the screen editor and check the F- and M-panels used as well as the "Screen Name" entry
HMI-00000203	The screen is not used	If the screen is to be used in IndraWorks Operation, assign the function "Screen change" to an F or M-key and select the name of the screen. If the screen is not called, but should be used later on, the screen can be deleted
HMI-00000204	The list of the screens in the Project Explorer is inconsistent compared to the data storage	Double-click on the message and confirm the prompt on insertion of the screen with Yes

Configuring the user interface

ID	Cause	Recovery
HMI-00000205	No starting area has been selected	Select an operating area as starting area below the "Screens" folder. Otherwise, an empty screen is displayed when enabling the operating area in IndraWorks Operation
HMI-00000206	No operating area is assigned to the screen	Move the screen to an existing operating area folder
HMI-00000300	The operating area is not used	Open the OP-panel editor and assign an OP-key with the "Change in operating area" function and select the operating area. If the operating area is not required, delete the element in the Project Explorer
HMI-00000301	No start screen is used for the operating area	Open the operating area editor and select the screen of the operating area as start screen. Otherwise, an empty screen is displayed when enabling the operating area in IndraWorks Operation
HMI-00000302	The key is not labeled	Open the respective panel editor and enter a label text for the key or select an image. Otherwise, there may be operating errors as an empty key is assigned a function
HMI-00000500	The F-panel is not used	Open a screen and select the F-panel there or assign the function "Following level" and the name of the F-panel to an F-key in another F-panel. If the F-panel is not required, delete it in the Project Explorer
HMI-00000600	The M-panel is not used	Open a screen and select the M-panel there or assign the function "Following level" and the name of the M-panel to an M-key in another M-panel. If the M-panel is not required, delete it in the Project Explorer
HMI-00001001	The WinStudio screen is not referenced.	Create a screen with the screen type "WinStudio screen" and select the WinStudio screen
HMI-00001002	There is an unused WinStudio variable.	Open the tag editor and remove the unused variable
HMI-00001003	The driver sheet item is invalid	Open the WinStudio driver sheet editor and correct the incorrect item. If necessary, delete it and then add it again via the browser window. Otherwise, communication with the control is not possible
HMI-00001004	Unknown OPC server	OPC driver sheets can only be validated for the OPC servers "IndraLogic.OPC.DA" and "OPC-IwSCP.1". Other OPC servers can also be used, however, the item can then not be validated
HMI-00001005	The OPC-UA connection is invalid	The "Connection" field of the OPC-UA driver sheet is empty or contains an invalid entry. Select an existing entry from the list
HMI-00001006	The item is empty	Each variable has to be connected to an item in the driver sheet to read values of a connected control or to write values to an item. Select a variable on the process variable browser and insert this variable in the empty item field in the driver sheet
HMI-00001007	The variable name is empty.	The value read by the connected control is not assigned to any variable and cannot be processed further. Enter the name of an existing or of a new variable
HMI-00001008	A variable is available multiple times in the driver sheet.	A multiple assignment can result in undesired behavior during runtime. Open the driver sheet and use different variables
HMI-00001009	The driver sheet contains too many entries.	Some drivers restrict the number of entries per driver sheet. Open the driver sheet and remove any redundant entries and insert the entries in a new driver sheet, if required

Configuring the user interface

ID	Cause	Recovery
HMI-00001010	The refresh time is invalid	Open the driver sheet and enter a numerical value under "Refresh time"
HMI-00001011	The item is too long.	In BR_WS driver sheets, the item lengths is limited to 960 characters. Open the driver sheet and remove the corrupt item
HMI-00001012	Invalid trigger variables	In the BR_WS driver sheet, a fixed value or a variable has to be entered in at least one field for "Read trigger", "Write trigger", "Read if idle" or "Write upon change"
HMI-00001013	The number of controls supported by the BR_WS driver has been exceeded.	The BR_WS driver can communicate with a maximum of 20 different controls. Open the driver sheets and check the used control. If required, remove the driver sheets for redundant controls
HMI-00001014	The task <Task name> is not enabled.	Each communication drivers requires a certain task for the execution during runtime. Open the task configuration and change the start mode of the specified task to "Automatic"
HMI-00001015	The driver <Driver name> is not enabled.	The project contains driver sheets for the specified driver. The driver is enabled. Open the context menu of the "Communication" folder, select "Add driver" and insert the specified driver
HMI-00001016	The task <Task name> is enabled.	The specified task is enabled in the project but no driver sheets are available. Open the task configuration ("Status") and change the start mode of the specified task to "Manual"
HMI-00001020	The WinStudio element contains an invalid WinStudio variable	Open the WinStudio element in the respective editor and select a valid variable

Tab. 5-9: Overview on error IDs of the validation messages

5.22 Version control of the visualization devices

5.22.1 General information

The visualization groups VEH/VEP and VPx/VSx support the IndraWorks version control with the respective application types "HMI Standard" and "WinStudio".

Configuring the user interface

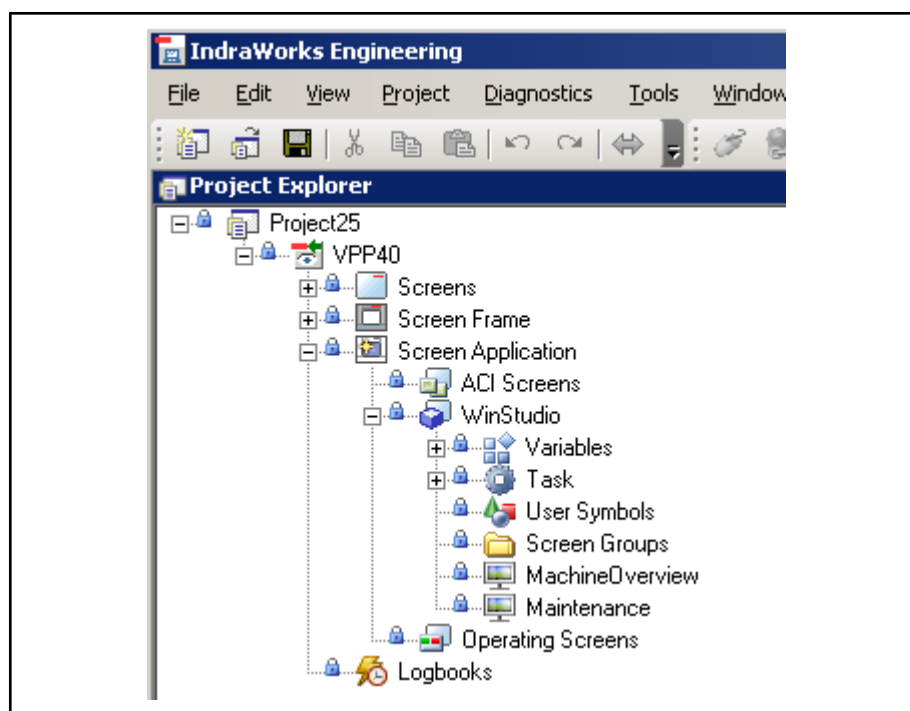


Fig. 5-153: Visualization devices under VCS

5.22.2 Special features

Operating screens: The operating screens used in the M-panel are implicitly checked in and checked out while configuring the respective panel in the operating screen editor.

Internal WinStudio variables: This node is not versioned, since the user cannot change the internal WinStudio variables. The variables generated by the user are managed and versioned from the "User Variables" node.

Comparing project data: Comparing individual HMI elements is currently not supported.

6 IndraWorks process variables

6.1 Tool window for process variables

6.1.1 General information

The tool window for process variables displays elements for process connection, such as PLC variables, OPC items, etc.. Additionally, user-friendly filter and search option are provided. If an editor requires the tool window, it is automatically displayed. Go to **ViewOther windows ► Process variables** to display or hide it independently.

6.1.2 Drag and drop

"Drag and drop" the elements of the tool window for process variables to the provided input fields of the editor windows.



It is possible to edit them manually at any time.

Input fields that may be assigned elements of the tool window for process variables are indicated by the process variable symbol if they are still empty. Otherwise they are colored. During a "Drag and Drop" operation, the input field displays additional tips on the current element.

6.1.3 Filtering

Control filter

If the tool window for process variables displays several controls, display or hide the elements for each control. In order to display elements of all controls, select **Show all**

Text filter

Enter a filter text and select one of the filter criteria provided. All elements that match the filter criterion are displayed. Delete the filter text in order to deactivate the text filter.

6.1.4 Searching

Text search

Enter a search text. During input, the first element containing the search text is already indicated. If an element is already selected, the search starts with this element and then increases the search radius. If no element can be found, the text field is colored red. To abort a started search, click on the **Cancel** button, if available.

Highlight search results

Select or deselect **Highlight search results** in the context menu to highlight the search results.

6.1.5 Additional functions

Views

Select one of the views provided to structure the elements as desired.

Sort

Sort the elements alphabetically in ascending or descending order.



The elements can only be sorted in the "List" view.

Expanding / reducing

Select **Expand** or **Reduce** in the context menu of an element to display or hide sub-elements.

Show element properties

Select or deselect **Display element properties** in the context menu to display or hide details of the selected element.

Get process variables from control

Use **Get process variables from control** to save all process variables of a control provided by the UA server of the control in a process variable file. After

IndraWorks process variables

the operation, the browser automatically switches to import mode. The generated process variable file can be loaded to the browser at any time and returns the process variables in the required syntax depending on the use. Refer to [chapter 6.1 "Tool window for process variables" on page 199](#).



The process can take several minutes and depends on the number of nodes in the address space of the UA server. Thus, it is recommended to only download Logic process variables.



The device name is used at a later point when returning the process variables.



If the process variables are no Logic process variables, the variables are always returned in an OPC UA syntax!

Data source for Logic process variables

Select the source of the Logic process variables from the Logic programming system.

- **Logic process variables of symbol file (Standard)**

The Logic process variables are supplied by the symbol file. Only "selected" Logic symbol variables that were selected explicitly in the Logic symbol configurator are displayed. Manually add a process variable to the Logic symbol configuration in order to display it in the browser. The browser is then automatically updated every time the symbol file is modified. The symbol file is updated by clicking on **Generate ► Generate code**.

- **Logic process variables of programming system**

The Logic process variables are directly supplied by the Logic programming system. All available variables are displayed and updated immediately after declaration and subsequent error-free compilation of the Logic program. If the process variables are "dragged and dropped" successfully and/or selected in the browser dialog, they are added automatically to the Logic symbol configuration. Click on the optional "Display Logic libraries" button in the context menu of this option to display public Logic process variables of integrated libraries as well.



This source requires significantly more resources than the symbol file and should therefore only be used for small to medium-sized Logic projects.

It may take several minutes to determine all Logic process variables if this option is enabled. Enabling the button "Display libraries" may increase this time even further.

Logic process variables have to be deleted from the Logic symbol configuration manually.

6.1.6 Import mode

The tool window for process variables provides an import mode to display elements that are not available in the current project but are needed for the project configuration.

Save a "process variables file" at any time from the tool window for process variables by clicking on the button **Create process variables file...**

Import as many process variable files as desired into the tool window for process variables by clicking on the **Load process variable file...** button.

Then, the tool window for process variables switches to the import mode and displays the content of these files. When at least one export file was loaded to the process variable tool window, use **Import view on/off** to switch between the "Normal mode" and the "Import mode".

Select **Delete** in the context menu of the export file element to delete the export file from the tool window for process variables.

6.2 Process variables dialog

The process variables dialog is displayed if editors can be used to select elements for process connection but do not support the tool window for process variables. The process variables dialog provides the same functionality as the tool window for process variables only with a slightly limited scope.

6.3 Browsing in WinStudio with OPC DA (OPC.IwSCP.1)

6.3.1 Browsing with driver sheets

Create a new WinStudio OPC DA driver sheet in **Screen Application ► WinStudio ► Variables ► Communication ► OPC ► New driver sheet** or open an existing driver sheet.

In the "Server selection" field, choose the IndraWorks OPC server "OPC.IwSCP.1" and save the driver sheet. The tool window for process variables is displayed.

Drag and drop the elements into the "Element" column and assign a variable name.

6.4 Browsing in WinStudio with OPC DA (IndraLogic OPC DA)

6.4.1 Browsing with driver sheets

Configure the IndraLogic OPC DA control name in **Screen application ► WinStudio ► Properties ► Communication ► Variable integration ► Configure ► Device name of the control**



Here, select the IndraLogic OPC DA device name set in the external IndraLogic OPC DA server configurator. Click on **Configurator** to open it.

Create a new WinStudio OPC DA driver sheet in **Screen Application ► WinStudio ► Variables ► Communication ► OPC ► New driver sheet** or open an existing driver sheet.

In the "Server selection" field, select the IndraWorks OPC server "IndraLogic.OPC.DA" and save the driver sheet. The tool window for process variables is displayed.

Drag and drop the elements into the "Element" column and assign a variable name.

6.5 Browsing in WinStudio with OPC UA

6.5.1 Browsing in OPC UA driver sheets

Create a new WinStudio OPC UA driver sheet in **Screen application ► WinStudio ► Variables ► Communication ► OPC UA ► New driver sheet** or open an existing driver sheet.

IndraWorks process variables

Choose an OPC UA connection in the "Connection" field and save the driver sheet. The tool window for process variables is displayed.

Drag and drop the elements into the "Element" column and assign a variable name.



The more stable "BrowsePath" of the element is returned and not the equivalent "NodeID" (that is dynamically generated for some servers).

Creating a new OPC UA connection

Create a new OPC UA connection with **Add**.

Configure the OPC UA connection with the respective parameters.



For IndraMotion controls use the syntax

```
opc.tcp://{IP address}:4840
```

Select **Screen application ► WinStudio ► Variables ► Communication ► OPC-UA ► EditSecurity...** for the OPC UA connection and click on "Trust server certificate".

6.6 Browsing in WinStudio with BR_WS driver

6.6.1 Browsing with driver sheets

Create a new WinStudio BR_WS driver sheet in **Screen application ► WinStudio ► Variables ► Communication ► BRWS ► New driver sheet** or open an existing driver sheet.

In the "Station" field, select the IP address of the control already entered into the driver sheet and save the driver sheet. The tool window for process variables is displayed.

Drag and drop the elements into the "Element" column and assign a variable name.

Configuring the BR_WS driver

Add the "BR_WS" driver in **Screen application ► WinStudio ► Variables ► Communication ► Add/remove driver ► BR_WS**

Configure the driver and set a control IP address in **Screen application ► WinStudio ► Variables ► Communication ► BRWS ► Properties ► IP address**

7 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

A

Abbreviations.....	8
About this documentation.....	5
Information representation.....	7
Validity of the documentation.....	5
ACI screens.....	43, 148
Activate operating area.....	74
Activate window after application start.....	76
Add operating area.....	71
Add or remove screens.....	71
Advanced settings.....	73
Arguments.....	75

B

Background color.....	150
Browse in WinStudio with BR_WS driver	
Browse with driver sheets.....	202
Browse in WinStudio with OPC DA	
(IndraLogic OPC DA)	
Browse with driver sheets.....	201
Browse in WinStudio with OPC DA (OPC.IwSCP.1)	
Browse with driver sheets.....	201
Browse in WinStudio with OPC UA	
Browse in OPC UA driver sheets.....	201

C

Call VI-Composer.....	185
Cause.....	25
Close together with desktop.....	75
Commissioning.....	15
Comparing visualization data.....	193
Complaints.....	9
Configuration.....	49
Configuration files.....	49
Configuration mode.....	43
Configuration of F-keys	
Structure in folders (from 13V06).....	82
Configuration of M-keys	
Structure in folders (from 13V06).....	91
Configuration of the user interface	
IndraWorks Engineering.....	148
User controls.....	165
Configure default structures.....	95
Configure F-keys	
F-panel editor.....	79
Configure for alternating display of messages.....	192
Configure individual texts and symbols.....	189
Configure M-keys	
M-panel editor.....	88
Configure screens.....	66
Configure operating areas.....	71
External applications.....	75
Screen control by the PLC.....	74
Screen editor.....	67

Configure user interface.....	49
Overview.....	49
Configuring F-keys.....	78
Delete F-panel.....	82
Rename F-panel.....	82
Configuring M-keys.....	86
Create M-panel.....	87
Delete M-panel.....	91
Programming M-keys in the PLC.....	91
Programming M-keys in the PLC for Vex devices.....	93
Rename M-panel.....	90
Configuring the F-keys	
Creating an F-panel.....	78
Configuring VCP devices.....	183
Context menu of the visualization device.....	53
Convert configuration data of old IndraWorks versions.....	183
Create a new screen.....	67
Criteria analysis window.....	41
Criticism.....	9
Customer Feedback.....	9

D

Delete operating area.....	72
Delete screen.....	71
Detailed view.....	26
Diagnostic display.....	24
Diagnostics.....	15, 33
Documentation structure.....	5
Download.....	9

E

Edit screen.....	68
Engineering.....	8
Error symbols.....	23
Export/Import.....	193

F

F-key.....	45
F-key operation.....	21
F-keys.....	78
F-panel.....	9, 19
Copy, delete, paste an F-key.....	81
Default.....	81
Details.....	81
Disable.....	81
Display.....	80
Function.....	80
Hide.....	81
Image file.....	80
image formats.....	80
Reset.....	81
Select image.....	80
Text.....	80

Index

Write variable.....	81
Feedback.....	9
Field bus diagnostics.....	33
Filter.....	26
Foil keys.....	22
Function.....	80
Function block.....	87, 91

G

General WinStudio settings and functions.....	117
---	-----

H

Header display.....	19, 20, 187
Helpdesk.....	203
HMI wizard.....	51
HMI_MKeys.lib.....	91
HMICE_MKEYS.....	94
HMICE_Mkeys.lib.....	93
Hotline.....	203
HTML Information for diagnostics.....	28

I

IL_VEXKEYS.....	95
Important instructions on use.....	11
IndraLogic-Target.....	91
IndraWorks.....	8
IndraWorks Engineering.....	148
ACI screen editor.....	149
New screen.....	158
User controls.....	161
IndraWorks HMI.....	13
IndraWorks HMI user interface.....	15
IndraWorks Operation.....	151, 153
IndraWorks process variables.....	199
Browse in WinStudio with BR_WS driver.....	202
Browse in WinStudio with OPC DA (IndraLogic OPC DA).....	201
Browse in WinStudio with OPC DA (OPC.lwSCP.1).....	201
Browse in WinStudio with OPC UA.....	201
Tool window for process variables.....	199
Insert VCP devices.....	184
Instance.....	87
Intended use.....	11

J

Join with running application.....	76
------------------------------------	----

L

Language setting.....	140
Language-dependent user texts.....	112
Layout.....	70
Layout template.....	158
Library.....	91
Library and function blocks.....	93
Lock and hide F- and M-keys by the user management.....	179

Logbook.....	29, 65, 113
--------------	-------------

M

M-key FB.....	55
M-keys.....	86
M-panel.....	9, 19
Control.....	89
Copy, delete, paste an M-key.....	90
Details.....	90
Display.....	89
Function.....	89
Function block instance.....	87
PLC flag.....	90
Preview bar.....	89

M-Panel

Add library.....	91
Machine operation.....	23
Machine overview.....	41
Message list.....	25
MKEYS.....	91
MKEYS_SINGLESEL.....	92
MKEYS_VPP21.....	93
More information.....	
Cause.....	25
Recovery.....	25
Multilingualism.....	140

N

Names and abbreviations.....	8
------------------------------	---

O

OEM data.....	108
OP-keys.....	83
OP-panel.....	9, 19
Copy, paste, delete OP-keys.....	85
Create.....	84
Details.....	85
Disable.....	86
Display.....	84
Function.....	84
Hide.....	86
Image file.....	84
Select image.....	84
Text.....	84
Write variable.....	86
OP9.....	24
Operating screens.....	36
Operation.....	8
Overview screen.....	26

P

Panel.....	9
Panel change.....	80
Pixels.....	80
PLC variable not found.....	23
Process variables tool window Additional functions.....	199

Drag and drop.....	199
Filter.....	199
General information.....	199
Import mode.....	200
Process variables dialog.....	201
Search.....	199
ProVi.....	15
ProVi message.....	9, 27

R

Recovery.....	25
Rename screen.....	71
RIL_VExUtil.lib.....	93

S

Safety instructions.....	7
Save screen.....	71
Screen attributes.....	120
Screen change.....	80
Screen creation.....	66
Screen layout.....	19, 150
Screen management.....	69
Screen name.....	69
Screen number.....	68
Screen parameter.....	69
Screen preview.....	71
Screen size.....	120
Screen type.....	69
ScreenPool.....	66
Screens.....	65
Select F-keys.....	69
Select M-keys.....	69
Service hotline.....	203
Short info.....	25
Start maximized.....	75
Start screen of an operating area.....	72
Starting area.....	73
Suggestions.....	9
Support.....	203
Symbol configuration.....	23
Symbol file.....	23
SYSLIBGETSYMBOLADDRESS.....	94
SysLibSymbols.lib.....	93

T

Touch screen.....	22
Transfer project files.....	186
Translation editor.....	141

U

Unintended use.....	11
User controls.....	165
User interface IndraWorks HMI-CE	
Start IndraLogic V and Operation Desk-	
top on a visualization device.....	15
User rights for screen and function calls.....	178
User screen.....	9, 116

User screens.....	23
User texts.....	25

V

Validate visualization data.....	193
Version control of the visualization devices.....	197
Version display.....	47
Visualization device.....	49
Visualization devices with foil keys.....	22
Visualization devices with touch screen.....	22

W

Warning triangle.....	23
WinStudio.....	9, 23, 65, 116
WinStudio applications	
Create a WinStudio application.....	141
Further information.....	142
Transfer application and activation on the	
target device.....	142
User management.....	141
Web functionality.....	141
WinStudio Applications.....	141
WinStudio licensing.....	138
Effects of a missing or limited license.....	140
Overview on WinStudio license.....	139
Overview on WinStudio licenses.....	139
WinStudio tools.....	133
Working directory.....	75

Notes

Notes

Bosch Rexroth AG

Electric Drives and Controls

P.O. Box 13 57

97803 Lohr, Germany

Bgm.-Dr.-Nebel-Str. 2

97816 Lohr, Germany

Phone +49 9352 18 0

Fax +49 9352 18 8400

www.boschrexroth.com/electrics



R911343569