

WAGO-Software

Manual

e!COCKPIT



2759-0101
e!COCKPIT

Version 1.0.1

© 2015 by WAGO Kontakttechnik GmbH & Co. KG
All rights reserved.

WAGO Kontakttechnik GmbH & Co. KG

Hansastraße 27
D-32423 Minden

Phone: +49 (0) 571/8 87 – 0
Fax: +49 (0) 571/8 87 – 1 69

E-Mail: info@wago.com

Web: <http://www.wago.com>

Technical Support

Phone: +49 (0) 571/8 87 – 5 55
Fax: +49 (0) 571/8 87 – 85 55

E-Mail: support@wago.com

Every conceivable measure has been taken to ensure the accuracy and completeness of this documentation. However, as errors can never be fully excluded, we always appreciate any information or suggestions for improving the documentation.

E-Mail: documentation@wago.com

We wish to point out that the software and hardware terms as well as the trademarks of companies used and/or mentioned in the present manual are generally protected by trademark or patent.

Table of Contents

1	Notes about this Documentation.....	7
1.1	Scope of Validity.....	7
1.2	Copyright.....	7
1.3	Symbols.....	8
1.4	Number Notation.....	10
1.5	Font Conventions	10
2	Important Notes	11
2.1	Legal Bases	11
2.1.1	Subject to Changes	11
2.1.2	Personnel Qualification	11
2.2	Safety Advice (Precautions).....	12
3	Overview	13
4	Requirements.....	15
4.1	System Requirements.....	15
5	Licenses	16
6	Installation.....	17
7	Starting.....	18
7.1	Purchasing a License.....	19
7.2	Activating the License.....	19
7.3	Activating Automatic Updates.....	21
8	Graphical User Interface.....	22
8.1	Start View (Projects and Templates).....	22
8.2	Main View (Configuring, Parameterization, Programming)	24
8.2.1	Quick Access Toolbar	25
8.2.2	Menu Ribbon	26
8.2.2.1	“START” Tab	26
8.2.2.2	“VIEW” Tab	30
8.2.2.3	“NETWORK” Tab.....	32
8.2.2.4	“DEVICE” Tab	34
8.2.2.5	“PROGRAM” Tab.....	36
8.2.2.6	“DEBUG” Tab.....	42
8.2.2.7	“/O CHECK” Tab.....	46
8.2.2.8	Other Tabs	48
8.2.3	Workspace	49
8.2.3.1	General Operating Elements.....	50
8.2.3.2	General Status Display for Connections.....	50
8.2.3.3	“Network/Devices” Workspace.....	53
8.2.3.3.1	Network View	54
8.2.3.3.1.1	Tiles.....	55
8.2.3.3.1.2	Connectors.....	57
8.2.3.3.1.3	Network/Bus Connections.....	59
8.2.3.3.2	Fieldbus-Specific Configurators	60
8.2.3.3.2.1	MODBUS Configurator	61

8.2.3.3.2.2	CANopen Configurator	66
8.2.3.3.2.3	PROFIBUS Configurator	80
8.2.3.3.3	Device Detail View	84
8.2.3.3.3.1	I/O Mapping	87
8.2.3.4	“Programming” Workspace	89
8.2.3.4.1	Program Editor	90
8.2.3.4.2	Library Manager	91
8.2.3.4.3	Task Configuration	93
8.2.3.4.4	“PLC_Task” Main Task	94
8.2.3.4.5	Breakpoints	97
8.2.3.4.6	Call Stack	97
8.2.3.4.7	Cross Reference List	98
8.2.3.5	User-Defined Workspaces	99
8.2.3.6	Panels	100
8.2.3.6.1	Device Structure	101
8.2.3.6.2	Program Structure	102
8.2.3.6.3	Product Catalog	105
8.2.3.6.4	Settings	108
8.2.3.6.5	Message Bar and Message Window	110
8.2.3.6.6	Other Panels	111
8.2.3.6.7	Status Bar	111
8.3	Backstage View (Files and Settings)	112
8.3.1	“New” Page	113
8.3.2	“Open” Page	114
8.3.3	“Save” and “Save As” Page	115
8.3.4	“Print” Page	116
8.3.5	“Import/Export” Page	117
8.3.6	“Project Settings” Page	119
8.3.7	“Product Catalog” Page	120
8.3.8	“Options” Page	122
8.3.9	“Help” Page	124
9	Operating	126
9.1	Working with the User Interface	126
9.1.1	Using Online Help	126
9.1.1.1	Calling Up Online Help	127
9.1.1.2	Search	127
9.1.2	Using Panels	128
9.1.2.1	Opening and Closing Panels	128
9.1.2.2	Docking, Floating and Grouping Panels	128
9.1.2.3	Docking and Floating the Main View	130
9.1.2.4	Using Multiple Selection	130
9.1.3	Using Workspaces	133
9.1.3.1	Opening the “Network/Devices” Workspace	133
9.1.3.2	Opening the “Programming” Workspace	136
9.1.3.3	Using User-Defined Workspaces	137
9.1.3.3.1	Creating a Workspace	137
9.1.3.3.2	Editing a Workspace	138
9.1.3.3.3	Deleting a Workspace	138
9.1.3.3.4	Restoring a Workspace	139

9.1.4	Using Key Combinations	140
9.1.4.1	General.....	140
9.1.4.2	“START” Tab.....	140
9.1.4.3	“PROGRAM” Tab.....	140
9.1.4.4	“DEBUG” Tab.....	141
9.2	Using Projects	142
9.2.1	Creating a New Project.....	142
9.2.2	Opening an Existing Project.....	142
9.2.2.1	Opening the Project with an Obsolete Device Description	143
9.2.3	Saving a Project.....	145
9.2.3.1	Saving Projects Automatically	145
9.2.3.2	Restoring Projects after a Software/Hardware Crash	146
9.3	Configuration and Parameterization	147
9.3.1	Offline Configuration	147
9.3.1.1	Adding Devices in the Project	147
9.3.2	Online Configuration.....	148
9.3.2.1	Scanning the Network and Devices	148
9.3.2.2	Scanning the Modules of Several Devices	153
9.3.2.3	Connecting and Disconnecting Devices	154
9.3.3	Parameterizing Devices	155
9.3.3.1	Parameterizing Devices via the “Settings” Panel	156
9.3.3.2	Using Device-Specific Configurators.....	157
9.3.3.3	Uploading and Downloading Parameters	158
9.3.4	Configuring Networks	159
9.3.4.1	Creating a Simple MODBUS Network	159
9.3.4.2	Creating a Simple CANopen Network	166
9.3.4.3	Creating a Simple PROFIBUS Network	171
9.4	Testing.....	177
9.4.1	Performing an Input/Output Test for Digital Input/Output Modules	177
9.4.2	Performing a Systematic Input/Output Test (I/O CHECK)	177
9.5	Programming.....	181
9.5.1	Creating Libraries.....	184
9.5.2	Adding Libraries.....	185
9.5.3	Creating Programs	187
9.5.4	Performing I/O Mapping	188
9.5.5	Compiling an Application	190
9.5.6	Excluding Programs for Compiling.....	191
9.5.7	Debugging an Application.....	192
9.5.8	Simulating an Application.....	193
9.5.9	Creating Visualizations	194
9.5.10	Connecting.....	197
9.5.11	Downloading Applications to the Device.....	198
9.5.12	Importing CODESYS 2 and CODESYS 3 Projects.....	199
9.5.13	Importing/Exporting Program Elements	202
9.6	Setting and Managing	204
9.6.1	Documenting and Printing a Project.....	204
9.6.2	Managing Devices	205
9.6.2.1	Adding Devices	205
9.6.2.2	Deleting Devices.....	206

9.6.2.3	Displaying Device Type Versions/Device Descriptions of a Device	207
9.6.2.4	Showing Device Type Versions/Device Descriptions of Several Devices	207
9.6.2.5	Replacing Device Type Versions/Device Descriptions.....	208
9.6.2.6	Replacing Older Versions Automatically	210
9.6.3	Performing Updates.....	211
9.6.4	Archiving.....	213
9.6.4.1	Creating an Archive	213
9.6.4.2	Opening an Existing Archive.....	214
9.6.5	Deinstalling.....	215
9.7	Interfaces	216
9.7.1	<i>smartDESIGNER</i>	216
9.7.1.1	Importing a Project from <i>smartDESIGNER</i>	217
9.7.1.2	Exporting a Project for <i>smartDESIGNER</i>	217
	List of Figures	218
	List of Tables.....	223

1 Notes about this Documentation

Note



Always retain this documentation!

This documentation is part of the product. Therefore, retain the documentation during the entire service life of the product. Pass on the documentation to any subsequent user. In addition, ensure that any supplement to this documentation is included, if necessary.

Note



Note about working with this documentation!

Please read the “Overview” section, it is an introduction to the software and provides a general description of functions. The following sections describe program installation and launch. Next, familiarize yourself with the *e!COCKPIT* graphical user interface. The subsequent sections contain operating instructions for using the software.

1.1 Scope of Validity

This documentation applies to the “*e!COCKPIT*” software.

1.2 Copyright

This Manual, including all figures and illustrations, is copyright-protected. Any further use of this Manual by third parties that violate pertinent copyright provisions is prohibited. Reproduction, translation, electronic and phototechnical filing/archiving (e.g., photocopying) as well as any amendments require the written consent of WAGO Kontakttechnik GmbH & Co. KG, Minden, Germany. Non-observance will involve the right to assert damage claims.

1.3 Symbols

DANGER

Personal Injury!

Indicates a high-risk, imminently hazardous situation which, if not avoided, will result in death or serious injury.

DANGER



Personal Injury Caused by Electric Current!

Indicates a high-risk, imminently hazardous situation which, if not avoided, will result in death or serious injury.

WARNING

Personal Injury!

Indicates a moderate-risk, potentially hazardous situation which, if not avoided, could result in death or serious injury.

CAUTION

Personal Injury!

Indicates a low-risk, potentially hazardous situation which, if not avoided, may result in minor or moderate injury.

NOTICE

Damage to Property!

Indicates a potentially hazardous situation which, if not avoided, may result in damage to property.

NOTICE



Damage to Property Caused by Electrostatic Discharge (ESD)!

Indicates a potentially hazardous situation which, if not avoided, may result in damage to property.

Note



Important Note!

Indicates a potential malfunction which, if not avoided, however, will not result in damage to property.



Information

Additional Information:

Refers to additional information which is not an integral part of this documentation (e.g., the Internet).

1.4 Number Notation

Table 1: Number Notation

Number Code	Example	Note
Decimal	100	Normal notation
Hexadecimal	0x64	C notation
Binary	'100' '0110.0100'	In quotation marks, nibble separated with dots (.)

1.5 Font Conventions

Table 2: Font Conventions

Font Type	Indicates
<i>italic</i>	Names of paths and data files are marked in italic-type. e.g.: <i>C:\Program Files\WAGO Software</i>
Menu	Menu items are marked in bold letters. e.g.: Save
>	A greater-than sign between two names means the selection of a menu item from a menu. e.g.: File > New
Input	Designation of input or optional fields are marked in bold letters, e.g.: Start of measurement range
“Value”	Input or selective values are marked in inverted commas. e.g.: Enter the value “4 mA” under Start of measurement range .
[Button]	Pushbuttons in dialog boxes are marked with bold letters in square brackets. e.g.: [Input]
[Key]	Keys are marked with bold letters in square brackets. e.g.: [F5]

2 Important Notes

This section describes the legal principles and system requirements for using the software in compliance with intended purpose, underlying provisions and stated specifications.

2.1 Legal Bases

2.1.1 Subject to Changes

WAGO Kontakttechnik GmbH & Co. KG reserves the right to provide for any alterations or modifications that serve to increase the efficiency of technical progress. WAGO Kontakttechnik GmbH & Co. KG owns all rights arising from the granting of patents or from the legal protection of utility patents. Third-party products are always mentioned without any reference to patent rights. Thus, the existence of such rights cannot be excluded.

2.1.2 Personnel Qualification

Any steps related to the use of WAGO software may only be performed by qualified employees with sufficient knowledge of handling the respective PC system used.

Steps in which files are created or changed on the PC system may only be performed by qualified employees with sufficient knowledge in the administration of the PC system used in addition to the aforementioned.

Steps in which the behavior of the PC system in a network is changed may only be performed by qualified employees with sufficient knowledge in the administration of the network used in addition to the aforementioned.

2.2 Safety Advice (Precautions)

Note



Use up-to-date security software!

Secure operation of the PC system can be at risk as a result of malware such as viruses and Trojans, as well as related threats such as denial-of-service attacks. Therefore, make sure that the latest security software and definitions are always installed on the PC system.

Note



Disable or uninstall software that is no longer required!

The vulnerability of a PC system against malware and related threats increases with the number of installed or active software components (applications and services). Therefore, uninstall or disable software components that are not needed for the purpose at hand.

Note



Check the runtime system if access problems occur!

To access devices from *e!COCKPIT*, *e!RUNTIME* must be set in your device as the runtime system. In the event of access problems, check the device settings with your tool (hardware dependent) or from the Web-based management system.

Note



Adjust the screen resolution if necessary!

The CODESYS programming environment in *e!COCKPIT* is optimized for a screen resolution of 96 DPI. If editors are displayed out of focus, you may be using a different resolution. To adjust the resolution:

1. Right-click on the desktop.
 2. Click **Customize**.
 3. Click **Display**.
 4. Select **“100%.”**
-

3 Overview

e!COCKPIT is an integrated development environment with seamless data retention for all automation tasks:

- **Hardware configuration and parameterization**

Hardware component configuration is an essential element of automation applications. All devices are configured so that the controller software produces optimum results during runtime. Controllers, fieldbus couplers/controllers, modules, input and output devices, as well as their communication relations can be configured.

Integrated configurators assist with device and network configuration using standard operating procedures: For example, it is possible to arrange devices in the project via Drag & Drop, use Copy & Paste to quickly multiply individual devices or entire network branches, or set parameter values for multiple devices simultaneously.

Besides the organization of devices in a tree structure, *e!COCKPIT* also enables the graphical display of a network topology – in the Network view – as a display form. This makes it possible to monitor complex interrelationships between devices and their current statuses. The network topology is likewise used for configuring different communication protocols. *e!COCKPIT* simplifies connecting controllers to fieldbuses.

Fieldbus-specific device description files (e.g., EDS), device drivers and libraries allow systems from other suppliers to be integrated into the topology along with WAGO devices. As such, the software has information on device specifications, i.e., device data or supported functions.

- **Programming with integrated *e!RUNTIME***

e!COCKPIT integrates the *e!RUNTIME* programming software that is based on CODESYS 3. This enables software development in the standard IEC 61131-3 programming languages: Structured text (ST), ladder diagram (LD), function block diagram (FBD), instruction list (IL), sequential function chart (SFC) and “Continuous Function Chart” (CFC). For flexibility, all programming languages can be combined with one another. Created programs can be checked easily via simulations on the development PC. Existing programs can be reused and further developed.

- **Visualization (operation and monitoring)**

e!COCKPIT uses Drag & Drop to streamline user interfaces for the operation and visualization of a plant. The integrated visualization editor offers direct access to the program's variables in order to simulate the human machine interface (HMI) and PLC program on the PC.

Using Unicode and the latest standards, such as HTML 5 or CSS, also prevents dependency on particular languages and target systems.

- **Diagnostics for target-oriented development and commissioning**

Whether in the office for development or at the machine for commissioning: Knowing the current, detailed status of the automation network is vital in order to obtain, audit and enable rapid fault localization and debugging. *e!COCKPIT* offers powerful diagnostic options for this purpose: Error messages are displayed immediately. Through the structured wiring test function, erroneous wiring can be systematically identified.

- **Other useful functions**

e!COCKPIT comes with an extensive range of IEC libraries. The software also offers several convenience functions, such as automatic updates, context-sensitive menus or user-defined workspaces.

Note



References to the *e!RUNTIME* programming environment!

Setting options in *e!COCKPIT* resulting from the integrated *e!RUNTIME* and CODESYS programming environment are described in the online help feature. Opening the online help in *e!COCKPIT* will also display the CODESYS documentation in the tree view next to the *e!COCKPIT* documentation. Individual setting windows or names within the CODESYS documentation may differ from how they are shown in *e!COCKPIT*.

4 Requirements

4.1 System Requirements

Table 3: System Requirements

Components	Requirements
Operating System	Windows 7/Windows 8
Memory	Min. 2 GB
Free hard disk storage	850 MB (x86) or 2 GB (x64) for .Net 4.5 and 3 GB for <i>e!COCKPIT</i>
Processor	Core2 Duo CPU or newer
Screen resolution	1,366 x 768 pixels
Miscellaneous	Network card, standard web browser with Java support, keyboard and mouse or touch screen

5 Licenses

e!COCKPIT software is protected by licensing mechanisms.

For productive, unlimited software usage, a license is required, and is available by email or phone from WAGO Kontakttechnik GmbH & Co. KG after purchasing the software.

The software can also be used fully without a license key for 30 days. This trial period only includes the days of actual use. When loading the software, a prompt appears with the number of days remaining. Access without a license key is not possible after the trial period.

Depending on the type of software license, an Internet connection may be required for activation. Please refer to your license certificate for the corresponding information.

Activating a license key is described in section “Starting.”

6 Installation

e!COCKPIT software is installed from a setup file. This contains a CODESYS programming environment, the .NET Framework and communication drivers for WAGO 750 Series devices. The device description files (Device Type Packages) for WAGO 750 Series devices are also automatically imported.

1. Open the [e!COCKPIT Website](#).
2. To receive a download link for the *e!COCKPIT* software, click the **[Registration]** link and complete the form.
3. Download the software.
4. Start the installation process by double-clicking the setup file and follow the steps described in the Installation Wizard:
 - Select the installation language.
 - Accept the license agreement.
 - Select the target directory for installation.
5. Click **[Install]** to start the installation.

If an older version is already installed, a dialog appears in which you can choose to accept already installed devices.

6. Click **[Finish]** to complete installation.

7 Starting

1. Launch *e!COCKPIT* via **Start > Programs > “WAGO Software” > “e!COCKPIT”** or via the link on your desktop.

A start screen will be displayed while the software is loaded.

If you have already purchased and activated a license, the lower area of this screen will display the name under which your *e!COCKPIT* application is registered.

If you have not yet acquired a license, you can see the remaining time on the start screen.



Figure 1: Display of the Start Screen during Loading

This also opens a dialog via which you can purchase/activate a license (see the following section).

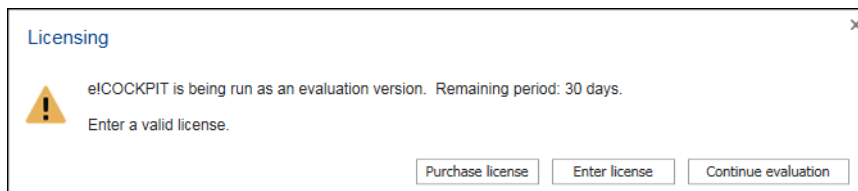


Figure 2: Display of the License Status

2. To test *e!COCKPIT* without entering a license key, click [**Continue evaluation**].

7.1 Purchasing a License

1. If you wish to purchase a license, click [**Purchase license**] after launching the software.

You will be taken to a Website where you can place the order (Online catalogue/eShop).

7.2 Activating the License

When purchasing the software, access data is transmitted by email or telephone.

1. To enter a purchased license key and activate the license, click [**Enter license**].
2. Enter your license key and customer name in the dialog.

Note



Pay attention to exact spelling!

Your access is personalized. Ensure that you enter your customer name exactly as it is written in the email that you received when purchasing the software.

Enter licenses

License key 123456789

Customer name MyCompany

I accept the [WAGO SOFTWARE LICENSE AGREEMENT](#)

I hereby confirm that I am a commercial user

Add licenses Cancel

Figure 3: Entering the License Key

3. Accept the license conditions and confirm that you are a commercial user.
4. To activate the license, click [**Add licenses**].

If the software has already been launched, open this dialog in the Backstage view: “Help” page, [**Product Licensing**], [**Enter License(s)**].

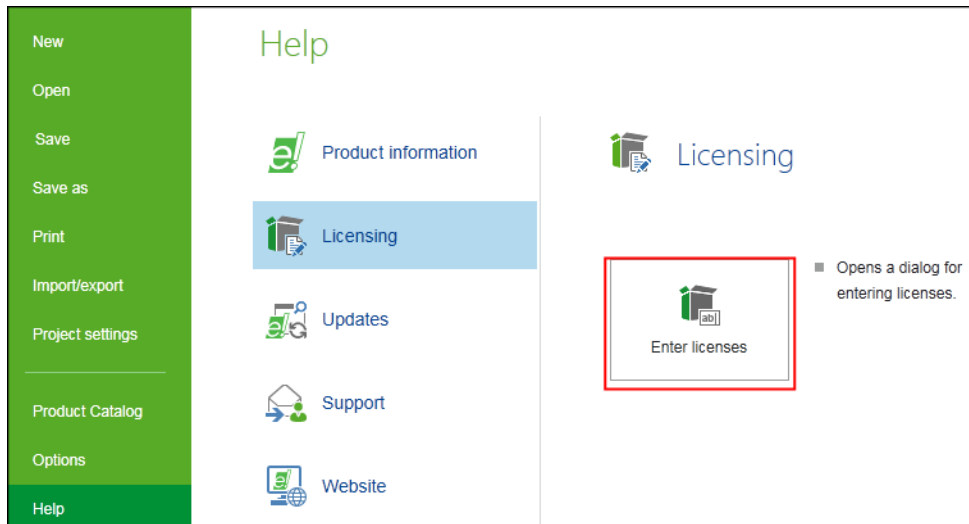


Figure 4: Entering Licenses

The license is created in your name and displayed on the “Help” page of the Backstage view.

7.3 Activating Automatic Updates

After the program is launched, the “Automatic updates” dialog is displayed.

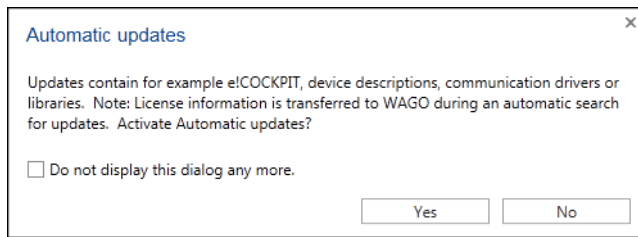


Figure 5: Activating Automatic Updates

1. To automatically search for updates when the program launches, click **[Yes]**.

If you choose **[No]**, you can enable the Update function later: In the Backstage view (“FILE” tab) under **[Help] > [Updates]**, tick the “Check for update and license information when starting *e!COCKPIT*” checkbox (Internet connection required on starting the program).

The update process compares your license information with the records of WAGO Kontakttechnik GmbH & Co. KG.

For additional information on performing updates, see the Section “Operating” > “Setting and Managing” > “Performing Updates.”

8 Graphical User Interface

The graphical user interface consists of three views:

- **Start View**
For launching the program and using templates.
- **Main View**
For configuring, parameterization and programming devices.
- **Backstage View**
For setting general options and file handling settings.

These views are described in the following sections. Start view is shown first when the software is launched.

8.1 Start View (Projects and Templates)

Start view enables you to create new projects or use existing templates. If projects were previously created, the most recently used projects can be accessed quickly from this page.

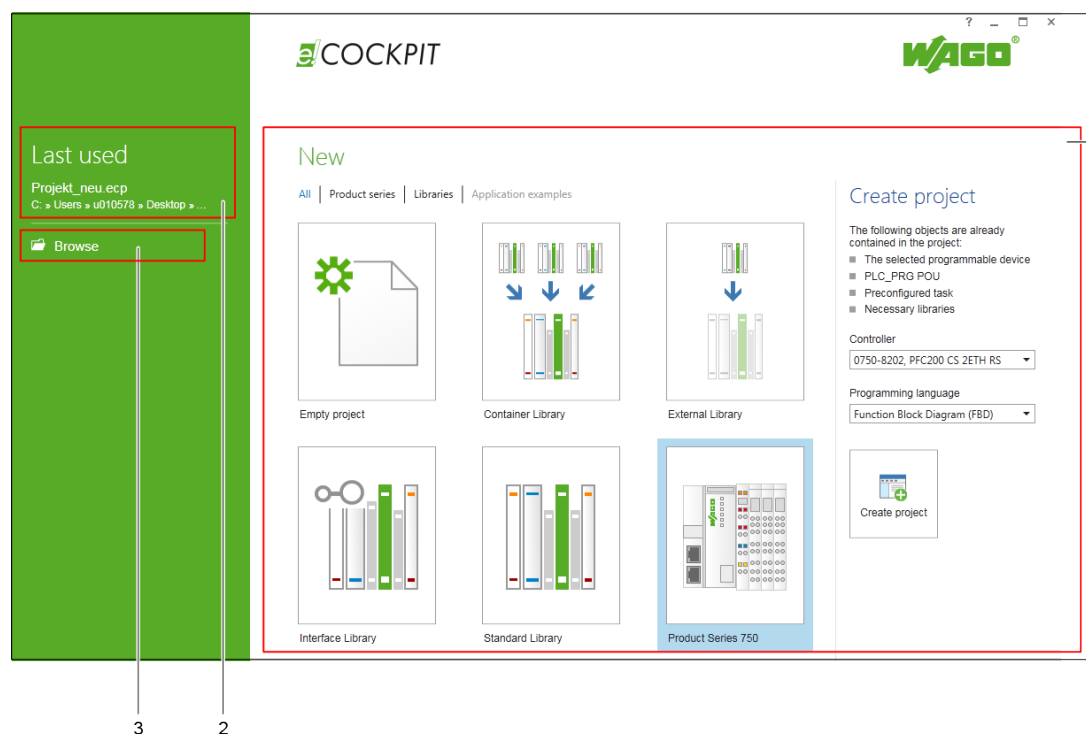


Figure 6: Start View

Table 4: Key for Figure “Start View”

No.	Description
1	New Displays templates that can be used for new projects
2	Last used Displays previous projects with the relevant path
3	Browse Switches to Backstage view, e.g., to open projects from other sources

8.2 Main View (Configuring, Parameterization, Programming)

The main view is divided into a quick access ribbon, menu ribbon, workspace and status bar. The dockable panels enable the workspace to be adjusted to suit the application.

The panels within the main view can be adjusted to create a suitable layout for the relevant task (e.g., creating networks, device configuration or programming). Besides fixed areas offering basic functions, other panels can be arranged as dockable windows depending on the project's needs.

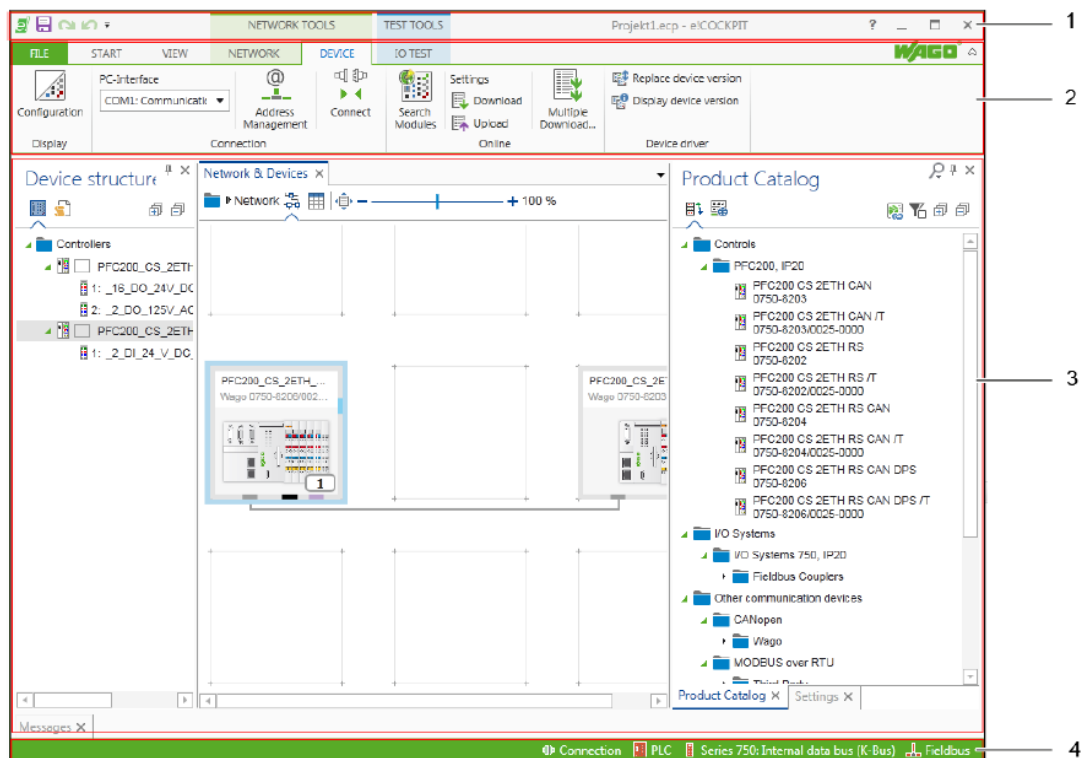


Figure 7: Main View

Table 5: Key for Figure “Main View”

No.	Description
1	Quick access toolbar
2	Menu ribbon
3	Workspace
4	Status Bar

8.2.1 Quick Access Toolbar

The quick access toolbar is located in the upper area of the window. The quick access toolbar is a toolbar that is independent of the currently selected tab and is always visible, making it ideal for frequently used commands.



Figure 8: Quick Access Toolbar (Standard View)



It contains the following buttons by default:

Table 6: Buttons on the Quick Access Toolbar

Symbol	Function	Description
	Save	Saves the project
	Undo	Undoes the last action
	Repeat	Repeats the last action
	Customize	Opens the menu for customizing the quick access toolbar
	Save	Adds or removes the [Save] button (Save project) to or from the quick access toolbar
	Undo	Undoes the last action
	Repeat	Repeats the last action
	Additional Functions...	Currently not available
	Display Above/Below Menu Ribbon	Displays the quick access toolbar above/below the menu ribbon
	Minimize Menu Ribbon	Minimizes/maximizes the menu ribbon.
	Display e!COCKPIT Help	Opens the e!COCKPIT online help The CODESYS help function is displayed in the tree structure underneath the e!COCKPIT help
	Minimize	Minimizes the window
	Maximize	Maximizes the window
	Close	Closes e!COCKPIT (Alt + F4)

Figure 9: Customizing the Quick Access Toolbar

Table 6: Buttons on the Quick Access Toolbar

Symbol	Function	Description
	Minimize Menu Ribbon	Displays only the tab names in the menu ribbon. The menu ribbon opens if a tab is clicked. It is then minimized afterward.*
	Extend Menu Ribbon	Displays the menu ribbon permanently*

*Toggling between a Minimized or Extended ribbon is also possible by double-clicking a tab name.

8.2.2 Menu Ribbon

Tabs are arranged under the quick access toolbar. Selecting a tab opens the particular tab ribbon with relevant command buttons based on task and group.

Relevant tabs are displayed according to the workspace. For example, the “PROGRAM” and “DEBUG” tabs are only displayed in the “Programming” workspace.

The buttons on the tabs are context-sensitive, i.e., only the functions that can actually be used at the moment are displayed and activated. Functions that cannot be executed within the relevant context are shown in gray.

The individual tabs are described below.

8.2.2.1 “START” Tab

The “START” tab contains buttons for general project editing and setting the user interface.

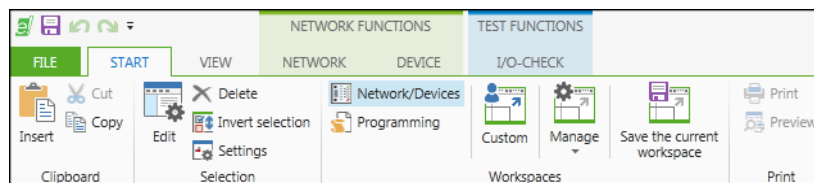


Figure 10: Menu Ribbon (Standard View)

Table 7: “START” Tab Operating Elements




Symbol	Function	Description
“Clipboard” Group		
	Insert	Inserts copied project element Only active if the previously copied or cut element can be inserted at the currently selected position
	Cut	Cut project element
	Copy	Copy project element

Table 7: “START” Tab Operating Elements









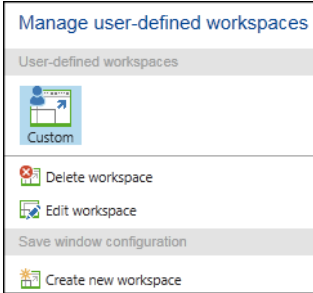



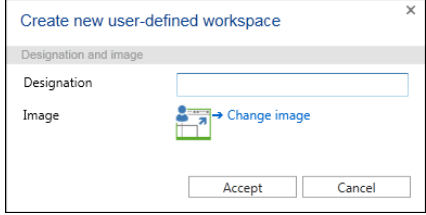



Symbol	Function	Description
“Edit” Group		
	Edit	<p>Opens one or several project elements for editing</p> <p>Double-clicking a project element triggers the same function.</p> <p>“Network/Devices” Workspace: Moves to the Device Detail view of the selected device and displays the “Settings” panel to edit device parameters.</p> <p>“Programming” Workspace: Depending on the selection in the Program Structure, displays the CODESYS program, project information or library managers</p>
	Delete	<p>Deletes one or several project elements</p> <p>“Network/Devices” workspace: Removes the selected devices from the project</p> <p>“Programming” Workspace: Depending on the selection in the Program Structure, deletes the CODESYS program, project information or library managers</p>
	Invert Selection	Inverts the devices/elements selection
	Settings	Opens the “Settings” panel for the selected object (device, program etc.), see Section “Workspace” > “Panels” > “Settings”
“Workspaces” Group		
	Network/Devices	Displays the preconfigured “Network/Devices” workspace (see Section “Network/Devices workspace”)
	Programming	Displays the preconfigured “Programming” workspace (see Section “Programming workspace”)
	<Workspace>	Opens the user-defined workspace (see Section “User-Defined Workspaces”)
	Manage	<p>Opens a context menu for managing workspaces</p>  <p>Figure 11: Managing workspaces</p>

Table 7: “START” Tab Operating Elements

Symbol	Function	Description
		User-defined workspaces
	<Workspace>	Selects the relevant workspace that is targeted for deletion or editing
	[Delete workspace]	Deletes the selected workspace The button of the user-defined workspace is removed from the menu ribbon
	[Edit workspace]	Used for customizing the workspace name and icon
		Save window configuration
	[Create new workspace]	Opens a dialog for creating a new workspace from the current window constellation
		 <p>Figure 12: Creating a New Workspace</p>
		Designation Field for entering the workspace name
		[Change image] opens an Explorer window for selecting a pictogram (JPG, PNG or BMP).
		[Accept] accepts the name and pictogram of the new workspace as a button in the menu ribbon.
		[Cancel] closes the dialog without saving the workspace.
	Save the current workspace	Saves the changed workspace
“Print” Group		
	Print	Prints the content of the active editor in the “Programming” workspace (available in the “Programming” workspace)
	Preview	Creates a print preview of the content for the active editor (available in the “Programming” workspace)

Note



Restore standard workspaces!

If one of the three predefined workspaces has been changed, you can restore it via **[Options] > [Standard workspaces] > [Reset standard workspaces]** in the “FILE” tab.

8.2.2.2 “VIEW” Tab

The “VIEW” tab enables you to display or close workspaces and panels. When a panel is displayed, the relevant buttons are highlighted in blue in the menu ribbon.

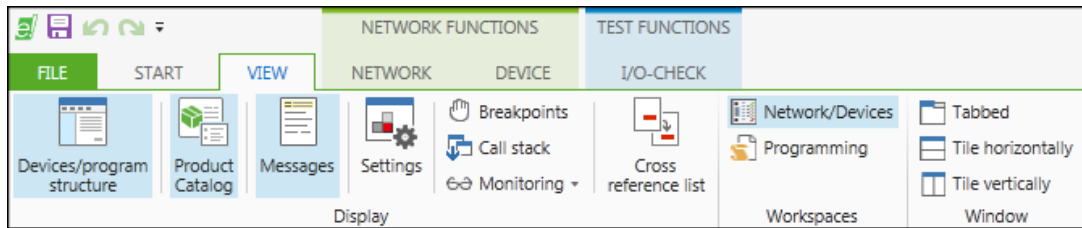







Figure 13: “VIEW” Tab

The “VIEW” tab contains the following operating elements.

Table 8: “VIEW” Tab Operating Elements

Symbol	Function	Description
“Show” Group		
	Devices/Program structure	Opens the “Device Structure” or “Program Structure” depending on which workspace is opened (see Section “Panels” > “Device Structure” or “Program Structure”)
	Product Catalog	Opens the “Product Catalog” panel (see Section “Workspaces” > “Product Catalog”)
	Messages	Opens the message list (see Section “Panels” > “Message Bar and Message Window”)
	Settings	Opens the “Settings” panel (see Section > “Panels” > “Settings”)
	Breakpoints	Opens the Breakpoints list (see Section “Programming” > “Breakpoints” workspace)
	Call Stack	Opens the call stack (see Section “Programming” > “Call Stack” workspace)
	Monitoring	Opens a selection field with the commands “Monitoring 1, 2, 3, 4” or “View all forced values”, which display the corresponding watch list “View all forced values” is a special view of all currently forced values. It enables “Program Organization Units” (POUs – program organization units of type program, function or function block) and a user-defined list of expressions/variables to be monitored.
	Cross reference list	Opens the cross reference list with the references of a variable in the project (see Section “Programming Workspace” > “Cross Reference List”)

Table 8: “VIEW” Tab Operating Elements

Symbol	Function	Description
“Workspaces” Group		
	Network/Devices	Displays the preconfigured “Network/Devices” workspace (see Section “Network/Devices workspace”)
	Programming	Displays the preconfigured “Programming” workspace (see Section “Programming workspace”)
“Windows” Group		
	Tabbed	Arranges the windows of the main view in individual tabs
	Tile horizontally	Arranges the windows of the main view vertically (tiled horizontally)
	Tile vertically	Arranges the windows of the main view horizontally (tiled vertically)

8.2.2.3 “NETWORK” Tab

The “NETWORK” tab contains functions that specifically relate to the display and scanning of devices in the network. The tab is displayed in the “Network/Devices” workspace.

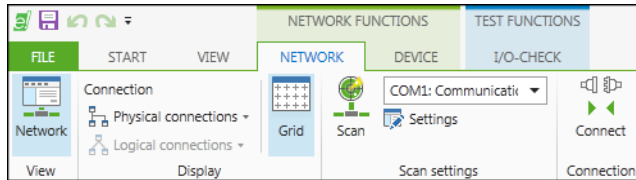











Figure 14: “NETWORK” Tab

The “NETWORK” tab contains the following operating elements:

Table 9: “NETWORK” Tab Operating Elements

Symbol	Function	Description
“View” Group		
	Network	Opens the Network view If the Network view is active, this button is highlighted in blue in the menu ribbon.
“Display” Group		
	Physical connections	Filters the displayed physical connections arranged according to ETHERNET, CANopen or PROFIBUS
	Logical connections	Filters the displayed logical connections according to PROFINET, MODBUS, EtherCAT, Data Server (CODESYS) or network variables (CODESYS)
	Grid	Shows or hides the grid in the Network view
“Scan Settings” Group		
	Scan	Starts the scan operation via the selected PC interface
	Interface	Selects the interface used for the scan (e.g., COM1, ETHERNET)
	Settings	Opens the Settings dialog for the selected interface

“Connection” Group		
	<p>Connect</p>	<p>Opens an online connection to one or several selected devices. The CODESYS application is connected with the target system. An ongoing online simulation is closed. The status of the button depends on the devices selected: “Connect” if no device is connected, “Disconnect” if all selected devices are connected. If the selected devices have different connection states, all devices are disconnected on the first click and connected with a second click. Connection requirements: - error-free program compilation - correctly configured communication device settings Connection to the selected device (login) is performed for all applications.</p>
	<p>Disconnect</p>	<p>Disconnects the development and target systems (controller or simulated device) and returns to offline mode</p>

8.2.2.4 “DEVICE” Tab

The “DEVICE” tab configures devices.

The tab is displayed in the “Network/Devices” workspace.

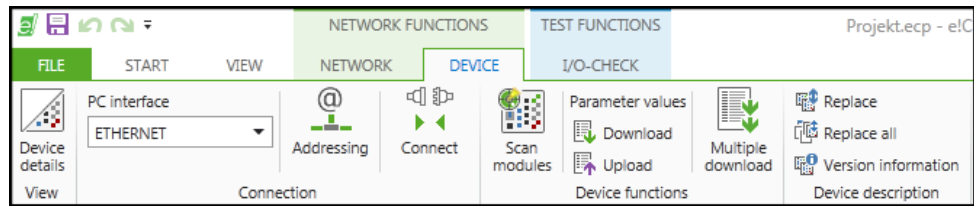


Figure 15: “DEVICE” Tab

The “DEVICE” tab contains the following operating elements.

Table 10: “DEVICE” Tab Operating Elements






Symbol	Function	Description
“View” Group		
	Device details	Opens the Device Detail view If the Device Detail view is active, this button is highlighted in blue in the menu ribbon
“Connection” Group		
	PC interface	Selects the communication interface to be used for the selected device and displays it
	Addressing	Sets the address for the selected device and the relevant interface, such as the IP address
	Connect	Opens an online connection to one or several selected devices. The CODESYS application is connected with the target system. An ongoing online simulation is closed. The status of the button depends on the devices selected: “Connect” if no device is connected, “Disconnect” if all selected devices are connected. If the selected devices have different connection states, all devices are disconnected on the first click and connected with a second click. Connection requirements: - error-free program compilation - correctly configured communication device settings The connection to the selected device (login) is performed for all applications.
	Disconnect	Disconnects the development system and target systems (controller or simulated device) and returns to offline mode

Table 10: “DEVICE” Tab Operating Elements





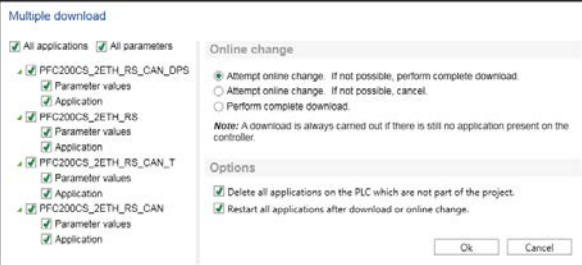



Symbol	Function	Description
“Device Functions” Group		
	Scan modules	Identifies and displays any modules present
	Download	Opens a connection to the selected device and downloads parameters to it
	Upload	Opens a connection to the device, reads device parameters and displays them in the software
	Multiple download	<p>Opens a dialog with a list of all project applications and parameters These can be selected and the selected device(s) can be loaded.</p> 
“Device Description” Group		
	Replace	Opens a dialog to replace the device description of the selected device
	Replace all	Opens a dialog for replacing or updating the device descriptions of all devices contained in the project
	Version information	Opens a dialog with version information for the device description of the selected device

Figure 16: “Multiple download” Dialog

Project devices, applications and parameters that can be selected via checkboxes are displayed in the left-hand area.

“**All applications**” selects all applications of the project.

“**All parameters**” selects all of the project parameters.

In the right-hand area, the standard CODESYS options are listed that refer to the changes, online change and (renewed) download of the application (see CODESYS Online Help, “Multiple download”).

[**OK**] loads the selected applications and parameters to the selected devices.

[**Cancel**] closes the dialog.

The “DEVICE” tab may show additional functions, depending on the selected device. The contents of device-specific tab groups are described in the relevant device documentation.

8.2.2.5 “PROGRAM” Tab

The “PROGRAM” tab contains functions for programming and loading the application onto a device. Actions of this tab always refer to active applications, i.e., applications that are connected or simulated online.

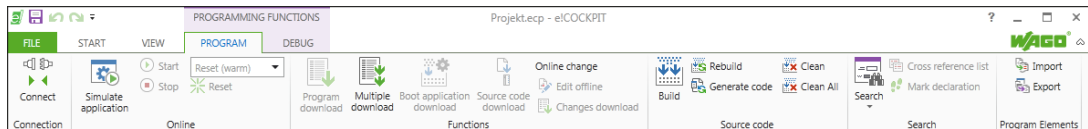


Figure 17: “PROGRAM” Tab

The “PROGRAM” tab contains the following operating elements:

Table 11: “PROGRAM” Tab Operating Elements




Button	Description	
“Debug” Group		
	Syntax check project library	Checks the syntax of the created library The button is visible if a library template was selected in the Start or Backstage view. The library is compiled in the Backstage view > [Import/Export] > [IEC library] .
“Connection” Group		
	Connect	Opens an online connection to one or several selected devices. The CODESYS application is connected with the target system. An ongoing online simulation is closed. The status of the button depends on the devices selected: “Connect” if no device is connected, “Disconnect” if all selected devices are connected. If the selected devices have different connection states, all devices are disconnected on the first click and connected with a second click. Connection requirements: - program compiled error-free - correctly configured communication device settings Connection to the selected device (login) is performed for all applications.
	Disconnect	Disconnects the development and target systems (controller or simulated device) and returns to offline mode

Table 11: “PROGRAM” Tab Operating Elements





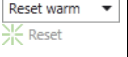

Button	Description	
“Online” Group		
	Simulate application	Can be selected if there is no online connection Starts Simulation mode The mode starts and debugs an application on a simulated target device in order to test the online behavior of an application in advance without requiring the actual device. No communication settings are required for the simulated device. When a function is called for the first time, a prompt asks whether the application is to be created or loaded.
	Close simulation	Closes the simulation mode; the simulation is also closed if the [Connect] button is clicked.
	Start	Starts the application on the controller or in the simulation
	Stop	Stops the application on the controller or in the simulation
	Reset	All Reset commands cause a restart in online mode. The set breakpoints in the project are deactivated. Reset (warm): Resets all variables of the active program to their initial and default values apart from the retentive variables (Retain, Persistent). Behavior is the same as switching the controller off and on while the program is running (e.g., in the event of a power failure). Reset (cold): The same as the “Reset warm” command, but also resets the retentive variables of the active application to their initialization values and default values (behavior is same as the start of the program currently loaded onto the controller). Reset (original): Resets the values of all variables of the active application to their initialization values and deletes the program on the controller.
“Functions” Group		
	Program download	Builds and compiles the active application in online mode, i.e., a syntax check is executed, application code is generated and then downloaded to the controller The compilation log is created in the project directory. All variables are reinitialized apart from the persistent variables.

Table 11: “PROGRAM” Tab Operating Elements


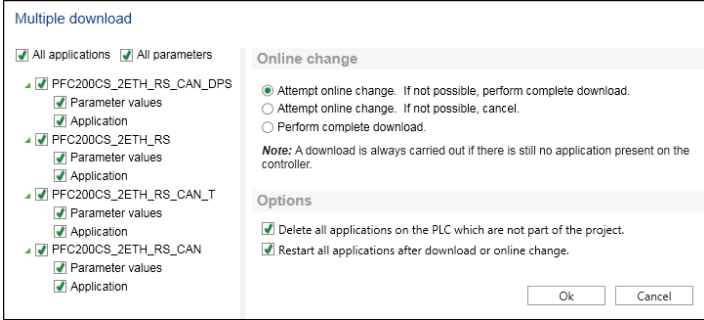





Button	Description	
	<p>Multiple download</p>	<p>Opens a dialog with a list of all project applications and parameters These can be selected and the selected device(s) can be loaded.</p>  <p>Figure 18: Multiple Download</p> <p>Project devices, applications and parameters that can be selected via checkboxes are displayed in the left-hand area. “All applications” selects all applications of the project. “All parameters” selects all of the project parameters. In the right-hand area, the standard CODESYS options are listed that refer to the changes, online change and (renewed) download of the application (see CODESYS Online Help, “Multiple Download”). [OK] loads the selected applications and parameters to the selected devices. [Cancel] closes the dialog.</p>
	<p>Boot application download</p>	<p>Saves the boot application under the name <Application_name>.app on the target device. This automatically begins when the controller is started. If the controller is in offline mode, the current application is first stored as a boot application in the project directory in order to download it later to a target system.</p>
	<p>Source code download</p>	<p>Downloads the current project as an “archive.prj” archive file to the currently connected controller.</p>
	<p>Edit offline</p>	<p>Enables the editing of a currently connected application The changes can be downloaded later to the controller via “Download Changes.”</p>
	<p>Changes download</p>	<p>Compiles changes to the active application carried out offline and downloads them to the controller</p>
“Source code” Group		
	<p>Build</p>	<p>Checks the source code of the active application for errors</p>

Table 11: “PROGRAM” Tab Operating Elements







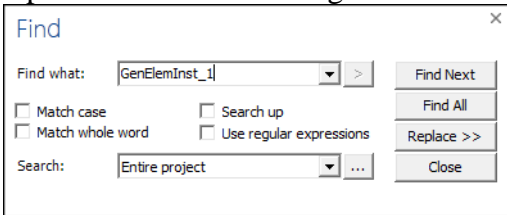
Button		Description								
	Rebuild	Checks the source code again								
	Generate Code	Compiles the source code of the active application. The code is not loaded onto the PLC.								
	Clean	<p>This cleans the compiler information for the currently active application on the development system. The compiler information was created the last time the application was downloaded to the target system and saved in a “*.compileinfo” file on the development system.</p> <div style="text-align: center; background-color: #e0e0e0; padding: 5px;">Note</div> <p> For online change after cleaning, reload the application! Reload the program to the controller after cleaning if you wish to use Online Change, otherwise it is not possible to execute Online Change.</p>								
	Clean All	<p>Deletes the compilation information for all applications. This was created the last time the applications were downloaded to the target systems and saved in a “*.compileinfo” file. After cleaning, an online change is no longer possible for the affected application. In this case, the program must be completely reloaded onto the controllers. Internal information on persistent variables is also deleted, so this must be re-installed after a download.</p>								
“Search” Group										
	Search	<p>Opens the “Search” dialog</p> <div style="border: 1px solid gray; padding: 5px; width: fit-content; margin: 5px auto;">  </div> <p>Figure 19: “Search” Dialog</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 20%;">Find what:</td> <td>Searches for the expression entered or selected from the drop-down list</td> </tr> <tr> <td>Match case</td> <td>Distinguishes between upper/lower case</td> </tr> <tr> <td>Search up</td> <td>When selected, searches for previous occurrences of the expression</td> </tr> <tr> <td>Match whole word</td> <td>When selected, searches for an exact term without additional characters</td> </tr> </table>	Find what:	Searches for the expression entered or selected from the drop-down list	Match case	Distinguishes between upper/lower case	Search up	When selected, searches for previous occurrences of the expression	Match whole word	When selected, searches for an exact term without additional characters
Find what:	Searches for the expression entered or selected from the drop-down list									
Match case	Distinguishes between upper/lower case									
Search up	When selected, searches for previous occurrences of the expression									
Match whole word	When selected, searches for an exact term without additional characters									

Table 11: “PROGRAM” Tab Operating Elements





Button		Description	
		Use regular expressions	When selected, allows the use of syntax rules for evaluating the expression, e.g., characters “*,” “+” or “?”
		[Find Next]	Searches for the next expression
		[Find All]	Shows all occurrences of the expression in the message window Double-clicking on this entry enables you to move in the editor to the appropriate position
		[Replace >>]	Opens a “Replace” dialog This dialog contains the following additional options: [Replace] replaces the next instance of the expression with the expression entered in the “Replace with” field. [Replace All] replaces all occurrences of the expression searched for.
		Search:	Selects the location to be searched: - Active editor - All opened editors - Entire project [...] opens a dialog with the additional option to search in objects and store this search as a search scheme.
		[Close]	Closes the “Search” dialog.
	Suchen ▼	Opens the quick access list for the search.	
		Search	Opens the “Search” dialog (Ctrl + F)
		Replace	Opens the “Replace” dialog (Ctrl + H)
		Next	Shows the next occurrence of the search expression (F3)
		Next (selection)	Displays the next occurrence of the search expression (Ctrl + F3) selected in the editor
		Search previous	Displays the previous occurrence of the search expression (Shift + F3)
		Search previous (selection)	Displays the previous occurrence of the search expression (Ctrl + Shift + F3) selected in the editor
		Cross reference list	Opens the cross reference list with the references of a variable in the project (see Section “Programming Workspace” > “Cross Reference List”)

Table 11: “PROGRAM” Tab Operating Elements

Button		Description
	Mark declaration	“Source text navigation”; jumps to the relevant definition/declaration of the identifier that the cursor is positioned on. Function block, program or variable is opened in the appropriate editor. If the definition is located in a “compiled” library, the corresponding function block is opened in the Library Manager.
	Import	Imports program elements of an export file
	Export	Save program elements in an export

8.2.2.6 “DEBUG” Tab

The “DEBUG” tab contains functions for executing and checking the source code.

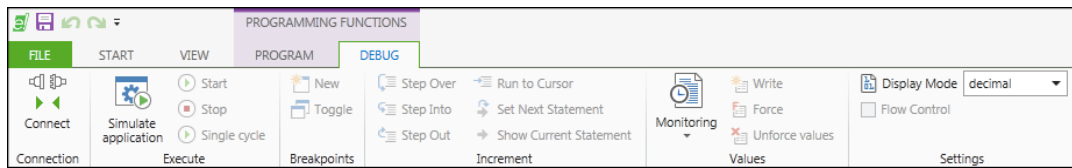


Figure 20: “DEBUG” Tab

The “DEBUG” tab contains the following operating elements:

Table 12: “DEBUG” Operating Elements

Button	Description	
“Connection” Group		
	Connect	<p>Opens an online connection to one or several selected devices.</p> <p>The CODESYS application is connected with the target system. An ongoing online simulation is closed.</p> <p>The status of the button depends on the devices selected: “Connect” if no device is connected, “Disconnect” if all selected devices are connected. If the selected devices have different connection states, all devices are disconnected on the first click and connected with a second click.</p> <p>Connection requirements:</p> <ul style="list-style-type: none"> - error-free program compilation - correctly configured communication device settings <p>Connection to the selected device (login) is performed for all applications.</p>
	Disconnect	<p>Disconnects the development and target systems (controller or simulated device) and returns to offline mode</p>
“Execute” Group		
	Simulate application	<p>Can be selected if there is no online connection; starts Simulation mode.</p> <p>The mode starts and debugs an application on a simulated target device in order to test the online behavior of an application in advance without requiring the actual device. No communication settings are required for the simulated device.</p> <p>When a function is called for the first time, a prompt asks whether the application is to be created or loaded.</p>

Table 12: “DEBUG” Operating Elements









Button		Description
	Close simulation	Closes the simulation mode; the simulation is also closed if the [Connect] button is clicked.
	Start	Starts the application on the controller or in the simulation
	Stop	Stops the application on the controller or in the simulation
	Single cycle	Performs one cycle of the active application
“Breakpoints” Group		
	New	Sets a breakpoint at the current position within the active code editor (see Section “Programming Workspace” > “Breakpoints”)
	Toggle	Toggles between the active and inactive state of a breakpoint. This also causes the setting of a new breakpoint (if not yet set at the current breakpoint position). Active breakpoints are deactivated after online mode is closed or after logging in again (see Section “Programming Workspace > “Breakpoints”).
“Increment” Group		
	Step Over	Executes a program in online mode in defined steps, e.g., for debugging. With instructions on one level, this corresponds to “stepping” with the “Step Into” command (see next command). When a function block call is reached, the entire function block is performed in a single step. A complete action is processed in a sequential function chart.
	Single Into	Executes a program in online mode in defined steps, e.g., for debugging. A single step is performed. If necessary, the program switches to another program organization unit (POU). If the current position is a function call or function block call, the program stops in front of the first instruction of the called function block. In all other situations, the command has the same function as the “Step Over” command. Possible stop positions during step-by-step processing depend on the type of editor. The current position is highlighted in yellow.

Table 12: “DEBUG” Operating Elements











Button	Description	
	Step Out	Steps a program backward In the case of a nested program code, the backward step is made through the calls. This causes a return to the start of the program if the program does not contain any calls. It also causes a return to the calling instruction if the program had jumped previously to a called function block.
	Run to Cursor	Runs the program from the current position to a temporarily definable position For this the cursor is placed at the required stop position before performing the command.
	Set Next Statement	Sets the statement to be executed next; to do this, the cursor is placed in this instruction before executing the command.
	Show Current Statement	This returns the cursor in online mode to the current execution position if the stepwise processing had moved the cursor to another position in the user interface.
“Values” Group		
	Monitoring	Opens a selection field with the commands “Monitoring 1, 2, 3, 4” or “View all forced values”, which display the relevant watch list. “View all forced values” is a special view of all currently forced values. It enables program organization units (POUs – program organization units of type program, function or function block) and a user-defined list of expressions/variables to be monitored.
	Write	Sets the corresponding variable in the controller at the start of the next cycle once to the value defined for it in the programming software. The command is applied to all variables of the active application that are prepared for writing.
	Force	If the controller is connected (online), the Force Values command causes one or several expressions/variables of the active application to be set permanently to defined values, i.e., they are not overwritten while the program is running. This command is available in online mode. [F7]
	Unforced values	Removes the forcing of all variables of the active application in online mode. Forcing is also removed by disconnecting the controller from the programming system. The variables are then assigned their current value from the controller.

Table 12: “DEBUG” Operating Elements

Button	Description
“Settings” Group	
	<p>Display Mode</p> <p>Sets the number format of variable values. The following can be selected: decimal, hexadecimal, binary.</p>
<input type="checkbox"/>	<p>Flow Control</p> <p>Switches the flow control in online mode on or off for the language editors ST, FBD, LD, IL and CFC; an activated flow control enables program processing to be followed in the editor window. Current variables, and thus the results of function calls and operations in the current cycle, are highlighted at the relevant processing position and time.</p> <p>The flow control functions in all currently visible sections of the currently opened editor window.</p> <p>Values cannot be written or forced in the flow control. This can only be performed in the declaration section.</p> <div data-bbox="890 1014 1426 1070" style="background-color: #cccccc; text-align: center;">Note</div> <div data-bbox="799 1070 890 1137" style="background-color: #0056b3; color: white; padding: 5px; display: inline-block;"></div> <p>Function increases run time! Activating the flow control increases the application runtime.</p>

 **WARNING**

Be aware of dangers resulting from writing or forcing values!

Abnormally changing a value in an application running on the controller may cause undesired behavior of the controlled system.

Evaluate any possible dangers resulting from writing or forcing variable values and take appropriate safety precautions. Depending on the system controlled, there is a risk of damage to the system and equipment or danger to people.

8.2.2.7 “I/O CHECK” Tab

This tab enables a systematic input/output test to be performed. It verifies that a controller is wired correctly. Both individual modules and entire nodes can be tested. The “I/O CHECK” tab is displayed if a device is selected or the Device Detail view of a device is opened. If several devices are selected, the “I/O CHECK” tab is hidden.

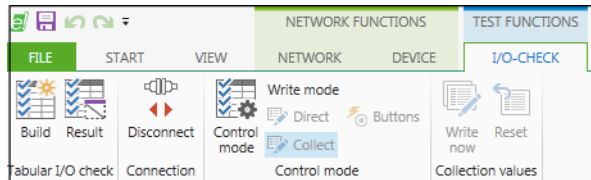








Figure 21: “I/O CHECK” Tab

The “I/O CHECK” tab contains the following operating elements: Active functions are highlighted in blue.

Table 13: “I/O CHECK” Tab Operating Elements

Symbol	Function	Description
“Tabular I/O check” Group		
	Build	Tests the connection (systematic hardware function test)
	Result	Switches the view of the “Status” and “Test Result” columns on/off
“Connection” Group		
	Connect	<p>Opens an online connection to one or several selected devices.</p> <p>The CODESYS application is connected with the target system. An ongoing online simulation is closed.</p> <p>The status of the button depends on the devices selected: “Connect” if no device is connected, “Disconnect” if all selected devices are connected. If the selected devices have different connection states, all devices are disconnected on the first click and connected with a second click.</p> <p>Connection requirements:</p> <ul style="list-style-type: none"> - error-free program compilation - correctly configured communication device settings <p>Connection to the selected device (login) is performed for all applications.</p>
	Disconnect	Disconnects the development and target systems (controller or simulated device) and returns to offline mode

“Control mode” Group		
	Control mode	Activates/deactivates the Control mode: Enables the modules for changes (write). If the control mode is not active and there is an online connection to the device, the monitor mode is automatically activated, i.e., actual values of the I/O module are displayed (read only).
	Direct	Activates/deactivates the mode for direct writing to the I/O module
	Collect (Number of prepared changes)	Writes prepared changes (marked with an asterisk) to the module
	Buttons	Activates/deactivates the push-button mode for digital inputs/outputs
“Collection values” Group		
This tab group is displayed if the “Collect” mode is active (blue background).		
	Write Now	Write collected values to the module. Changed values are indicated with an asterisk.
	Reset	Reset collected values. Changed values are indicated with an asterisk.

Besides this systematic input/output test, a simple test can also be performed for digital input/digital output modules via the Device Detail view (see Section “Workspace” > ... > “Device Detail view”).

8.2.2.8 Other Tabs

Depending on the devices used, other tabs may be displayed that have specific content relating to the particular device driver. The setting options can be found in the relevant product documentation of the device.

8.2.3 Workspace

The workspace consists of the window area underneath the menu ribbon. This is divided into the main workspace and panels that can be arranged freely; these elements can be shown or hidden via the “VIEW” tab to suit the particular application or specific operation. Panels are used to display and/or edit project data or other data.

e!COCKPIT offers two predefined workspaces that are already adapted for typical applications such as network and device configuration, parameterization and programming. The appearance of the user interface changes according to the workspace selected.

User-defined workspaces can be created in addition to the predefined workspaces. The position and size of selected views, tabs and panels are saved here to customize the workspace. These settings are stored separately for each Windows user.

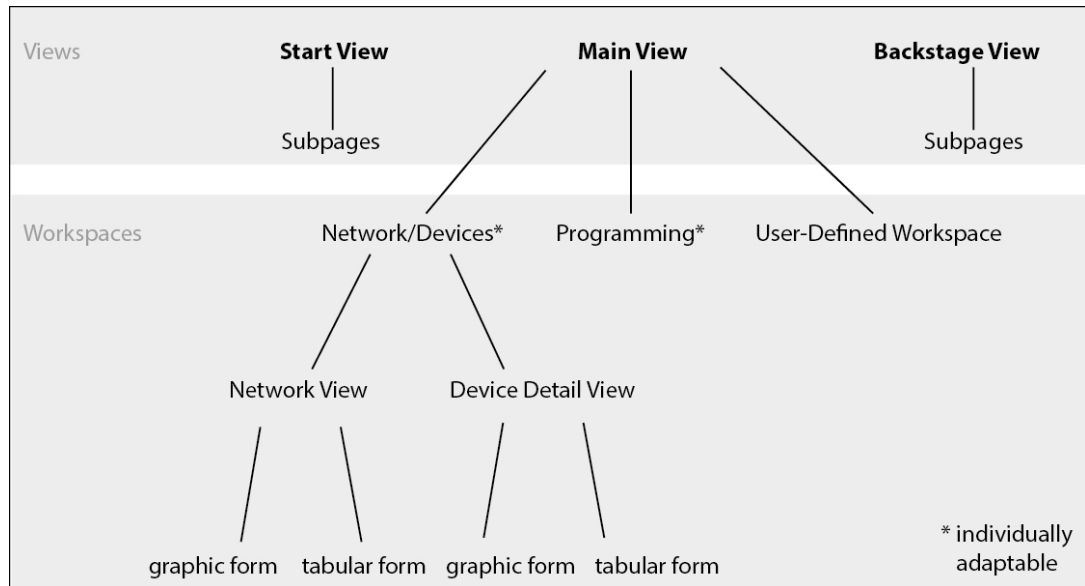


Figure 22: Views and Workspaces









After opening a newly created project in *e!COCKPIT*, the standard view shows the “Device Structure” panel on the left, the Network view in the middle and “Product Catalog” and “Settings” panels on the right.

The configuration, parameterization and programming are performed in the central window area.

8.2.3.1 General Operating Elements

The main workspace and panels have the following operating elements (some of these functions depend on the panel):

Table 14: General Operating Elements

Symbol	Function	Description
	Search 	Shows or hides the search field. (free text search, no wildcards necessary) The search field is permanently displayed in some panels.
	Dock	Hides the panel; Panel appears as a tab on the right of the window or at bottom left.
	Close	Closes the panel
	Zoom	Increases/decreases magnification of the view (Zoom) (Network/Device Detail view)
	Adjust	Adjusts the view to the optimum size or to the window (Network/Device Detail view)
	Expand	Opens all elements of the tree structure
	Collapse	Closes all elements of the tree structure

8.2.3.2 General Status Display for Connections

A device's connection status is shown as a four-section status display in operating fields and workspaces.

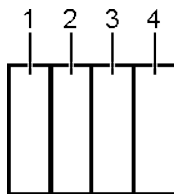






























Figure 23: General Status Display for Connections

Table 15: Key for Figure “General Status Display for Connections”

Item	Explanation
1	Connection status
2	PLC status
3	Status of the internal bus (e.g., internal data bus, 750 Series)
4	Fieldbus status

The status information is likewise displayed in the status bar. The following table provides an overview of the different symbols:

Table 16: General Status Display for Connections

Symbol in Status Bar	Symbol in Workspace	State	Description
Connection			
		Not connected	The selected device is not connected.
		Connected	The selected device is connected.
		Limited Connection	The selected device has a limited connection. This means that the application on the controller is different than the application in the project within <i>e!COCKPIT</i> .
		Control Mode Active	The Control mode is active. Write access to modules is possible. No online connection to the device.
		Simulation Mode Active	The Simulation mode is active.
PLC			
		Stop	The controller is not running (PLC stopped).
		Running	The controller is running.
		Error	Error is present. Note the error messages in the message window.
Internal Bus			
		Not running	No communication via the internal bus.
		Running	Internal bus ready for communication.
		Error	Error is present. Note the error messages in the message window.
Fieldbus			
		Not running	No communication via fieldbus.
		Running	Fieldbus ready for communication.
		Error	Error is present. Note the error messages in the message window.

If a device is connected online, connection status is also displayed in a pop-up display as soon as the mouse is placed over the device. The status information is displayed both for fieldbus couplers/controllers as well as for connected modules.

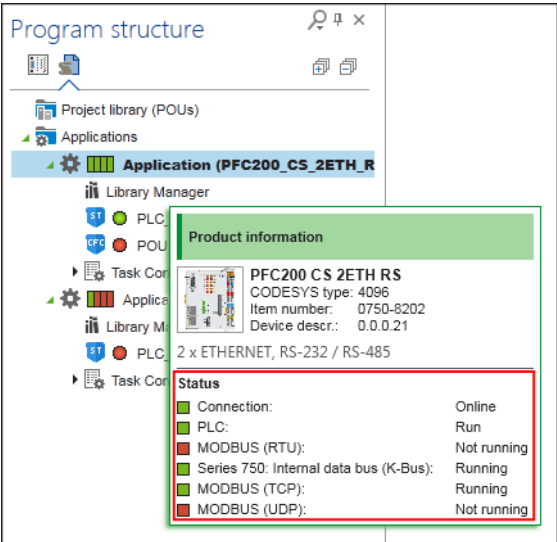


Figure 24: Status Display for Devices (Online)

8.2.3.3 “Network/Devices” Workspace

The “Network/Devices” workspace is used to configure and parameterize the network and devices. The workspace contains the following operating elements:

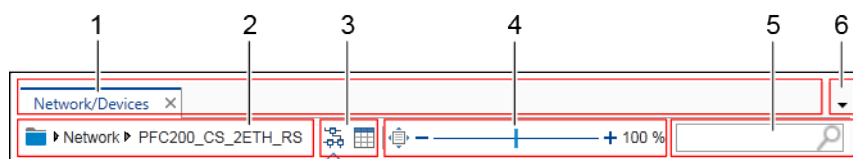





Figure 25: “Network/Devices” Workspace

Table 17: Elements of the Network View

No.	Description
1	Tab “Network/Devices” tab and other tabs, for example, open configurators.
2	Breadcrumb Navigation The path of the view which is currently displayed. The path can be used for navigation by clicking on an entry. Clicking Network for example returns the display from Device Detail view to Network view.
3	Graphic/Table Toggle Button Button is used for toggling between graphical and tabular view.
	 The graphical view is suitable for adding and configuring devices and networks.  The tabular view is primarily designed to provide a better overview during a network or device scan and for comparing configured and found devices.
4	Zoom Function Zoom adjuster that can be clicked and moved to enlarge or decrease the view. “+” or “-” are the buttons that adjust the view in 10 % steps.
	 The button adjusts the view to the current window.
5	Search field Free text search, no wildcards are necessary.
6	Active Tabs Shows active tabs in a selection list.

8.2.3.3.1 Network View

Network view displays configured network connections between devices. It contains the following elements:

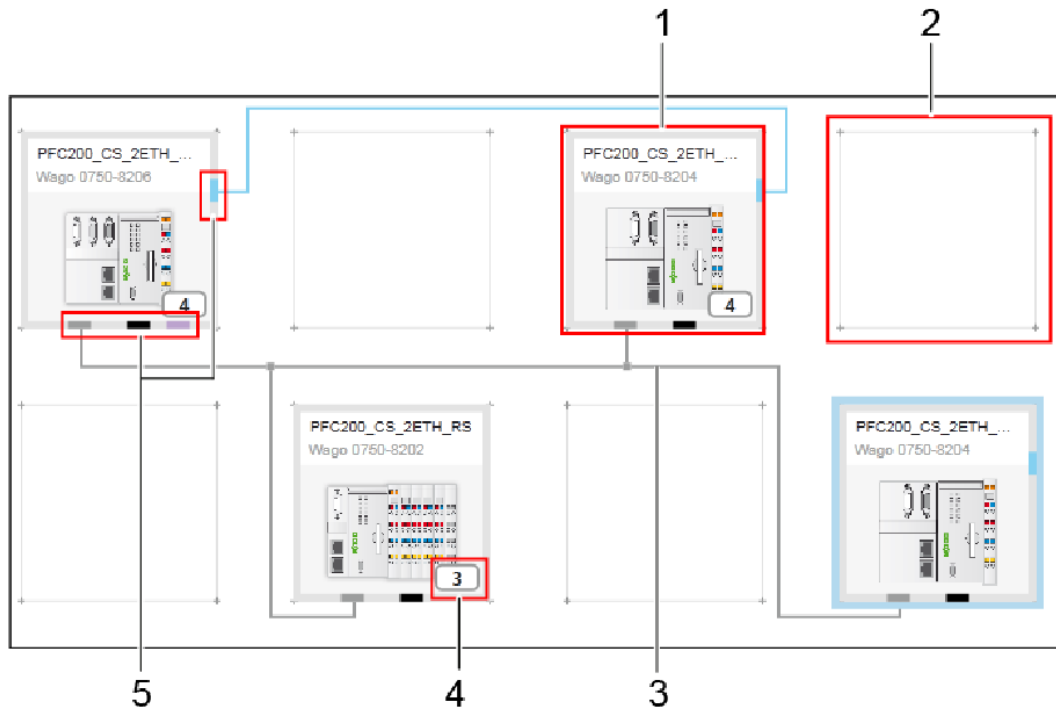


Figure 26: Network View

Table 18: Key for Figure “Network View”

No.	Description
1	Title Displays the included device. To change a name, click the displayed device name to enter a new name.
2	Device Location Empty tile as location for devices.
3	Network/Bus Connections Physical/logical connections between devices are displayed as horizontal/vertical lines. Colors are uniquely assigned to a particular network/bus type.
4	Number of Modules This shows the number of modules connected with this head station and which of these are included in the Device Detail view.
5	Connectors The interfaces via which connections to other devices can be configured.

8.2.3.3.1.1 Tiles

The Network view’s workspace is divided into tiles. Tiles are placeholders for devices that are added from the Product Catalog. The device's status is indicated by its frame:

Table 19: Device Status

Frame Color	Description
Gray	Standard frame of an included device.
Light Blue	Device has been selected/highlighted.
Green	The device is connected online.
Light green	The device has a limited connection (controller application is different to the project application in <i>e!COCKPIT</i>).
Orange	Control mode (“I/O-CHECK” tab) is active.
Blue	Simulation mode is active (“PROGRAM”/“DEBUG” tab).

The device’s status — with regard to the connection, PLC, internal bus and fieldbus — is indicated by a four-section status display in the top right corner of the tile. The individual symbols are explained in the Section “Workspace” > “General Status Display for Connections”.

Right-clicking the tile displays the context menu’s different buttons:

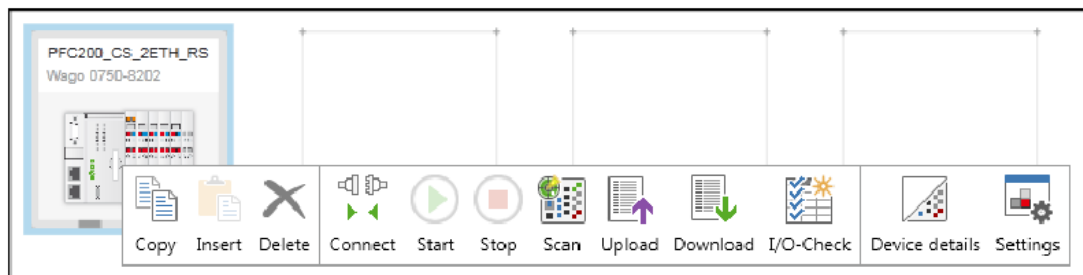















Figure 27: Context Menu of a Device

Table 20: Context Menu of a Device

Symbol		Description
	Copy	Copies selected devices
	Insert	Inserts copied devices
	Delete	Deletes the selected device from the Network view. When deleting the device, the associated program elements are also deleted.
	Connect	Opens an online connection to one or several selected devices
	Disconnect	Removes the connection between the development and target systems (controller or simulated device) and returns to offline mode
	Start	Starts the application on the controller or in the simulation
	Stop	Stops the application on the controller or in the simulation
	Scan	Scans the node for the presence of modules and displays results in a list
	Upload	Reads parameters from the selected device
	Download	Writes parameters to the selected device
	I/O-Check	Opens the “I/O CHECK” tab (see Section “I/O CHECK Tab”)
	Device details	Opens the Device Detail view and displays the I/O mapping
	Settings	Displays device settings in the corresponding panel (see Section “Panels” > “Settings”)

8.2.3.3.1.2 Connectors

Each device supports specific interfaces for communicating with other devices. The interfaces are shown as connection points (connectors) in different colors on the edge of a tile:



Figure 28: Color Coding of Connectors

Table 21: Color Coding of Connectors

Connector Color	Description
Gray	Connector for ETHERNET/MODBUS connections
Black	Connector for serial connections/COM interfaces
Blue	Connector for CANopen connections
Purple	Connector for PROFIBUS connections

Hovering the mouse over a connector displays additional information, such as interface type, the device's IP address or connections to other devices, if present.

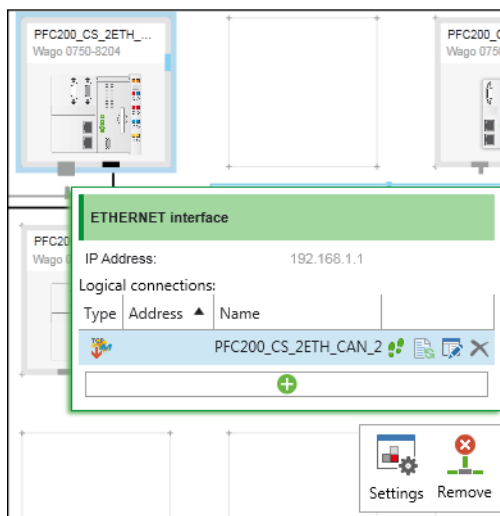










Figure 29: Context Menu of a Connector

Table 22: Operating Elements in the Context Menu of a Connector

Symbol	Description
	Pictogram for the type of logical connection
	Selects the connected device and centers the view
	Updates the configuration The button is available when the configuration of connected devices has been changed. Example: The configuration of a master device is adapted to the configuration of the connected slave. This is indicated by a warning symbol next to the interface: 
	Opens the connection settings (see Section “Fieldbus-Specific Configurators”).
	Removes the logical connection
	Displays device settings in the corresponding panel (see Section “Panels” > “Settings”)
	Deletes all connections for this interface to any number of devices

8.2.3.3.1.3 Network/Bus Connections

The devices can be interconnected in Network view. Connectors are dragged with the mouse between two similar device connectors. The color of a connection is the same as the relevant connector color (see previous Section).

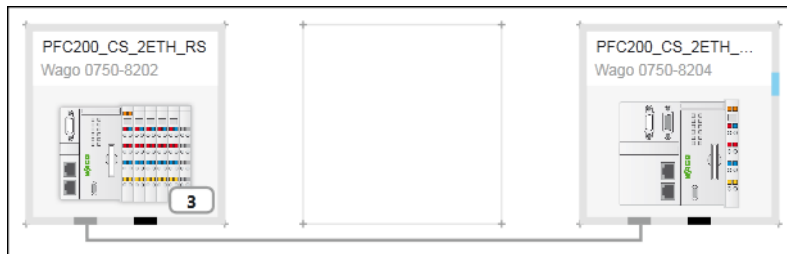


Figure 30: Connection between Two Devices

Connections are either physical or logical connections.

Physical connections correspond to cables between devices that are compatible with their hardware, i.e., their terminations. The direction in which connection lines are dragged with the mouse is not important to physical connections.

Logical connections are only possible where physically connected devices can exchange information with the appropriate protocols. Logical connections are dragged over an already displayed physical connection.

Unlike physical connections, the direction in which devices are connected is relevant for logical connections as the displayed options (e.g., “Modbus TCP master”) refer to the source device.

Example: When selecting “Modbus TCP Master” the source device becomes the master, the target device becomes the slave for the new connection.

With connections via CANopen and PROFIBUS, the physical connection is the same as the logical connection. Only one connection line must be dragged between the devices. Whether it is a master or slave device depends solely on the configuration of the device concerned.

To control data exchange, each logical connection can be further configured via the buttons in the context menu of the connector (see previous Section).

8.2.3.3.2 Fieldbus-Specific Configurators

Fieldbus-specific configurators are used to configure MODBUS, CANopen and PROFIBUS connections. These are accessed via the context menu of a connector and displayed in the main workspace.

The connected devices, as well as the device type, are displayed in the upper area.

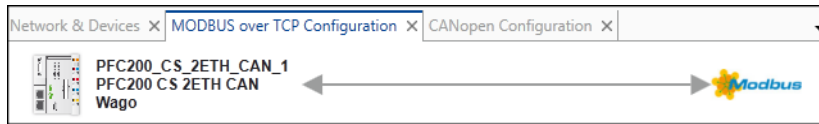


Figure 31: Fieldbus-Specific Configurators

The lower area is structured according to the specific fieldbus, see the following Section.

8.2.3.3.2.1 MODBUS Configurator

e!COCKPIT integrates two MODBUS configurators for configuring the connection between MODBUS master and MODBUS slave.

The data points/variables to be exchanged in a real master-slave connection is set in the master configurator.

The slave configurator sets which data points/variables are available for **all** masters.

MODBUS Master Configurator

The settings are made in two tabs.

The “MODBUS Master-Variables” tab configures data exchange between the master and slave. The left-hand column here displays the variables used by the slave. The right-hand column displays the variables available on the master.

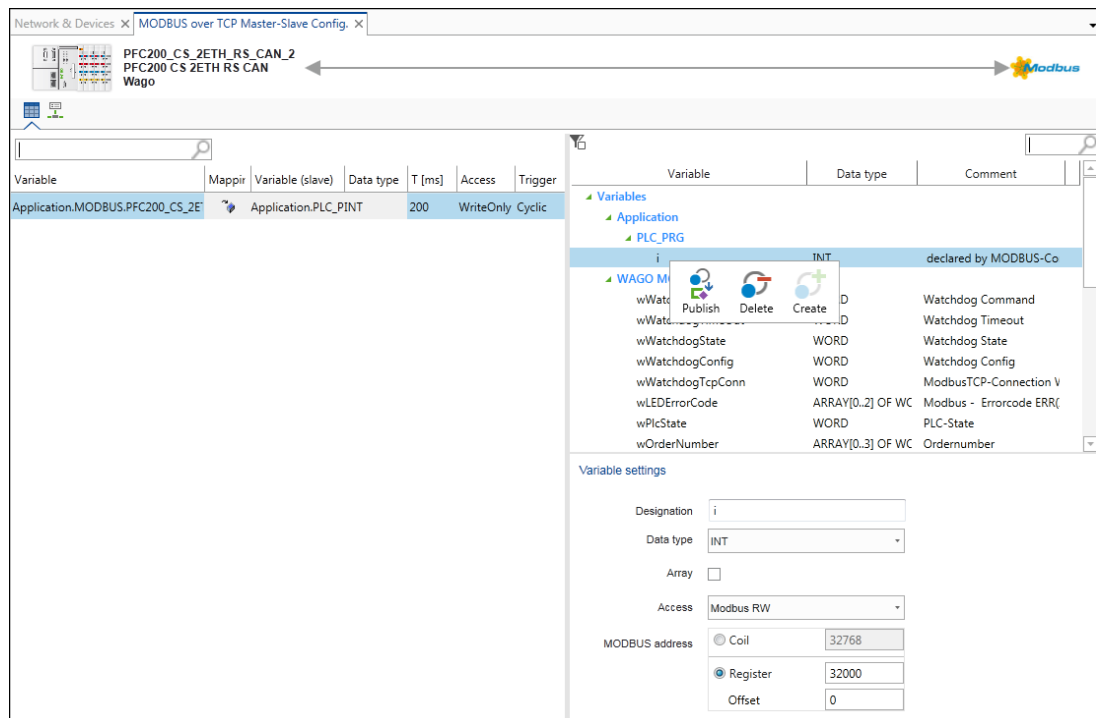


Figure 32: “MODBUS Master Variables” Tab

Table 23: Key for Figure “MODBUS Master Variables” Tab

Parameters	Description
Variables (Slave, left-hand column)	
Variable	Variable name and path on the slave Variants are normally either manual or automatic. Automatic: A variable is created according to the variable path on the master. Manual: The variable is entered by the user. This can be an existing variable or one created by CODESYS.
Mapping	Existing (arrow symbol) variables and variables generated by CODESYS (asterisk)
Variable (slave)	Variable name and path on the master
T [ms]	Cycle time when accessing the variables
Access	Read and write access
Trigger	Triggering of access. Selection depends on the access setting. Read: Cyclic Write: Cyclic/on value change
Variables (Master, right-hand column)	
Variable	Name of variable
Data type	Variable data type
Comment	Comment of the variable
Variable settings	
Designation	Name of variable
Data type	Variable data type
Array	Change data type to array (to read or write access several MODBUS tabs at once). The size of the variables is set via the number of elements in the array.
Access	Access via MODBUS (RO/RW)
MODBUS address	MODBUS address of the variable

Context menu of the slave variables (left-hand column):

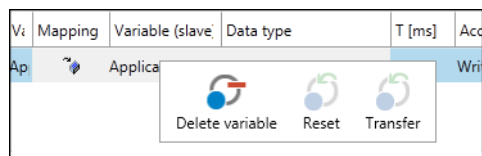





Figure 33: Slave Variables Context Menu

Table 24: Key to Figure “Slave Variables Context Menu”

Symbol	Function	Description
	Delete variable	Delete variable of slave
	Reset	Reset variable to “automatic value”
	Transfer	Accept value of the cell for all selected rows (T [ms], trigger, access)

Context menu of the master variables (right-hand column):

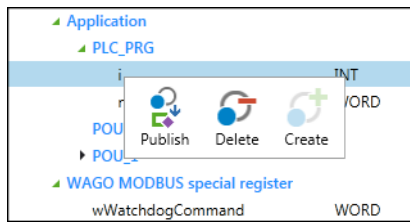





Figure 34: Master Variables Context Menu

Table 25: Key to Figure “Master Variables Context Menu”

Symbol	Function	Description
	Publish	Publish variable via MODBUS and make it available in the slave
	Delete	Delete variable of master
	Create	Create new variable on the master

In the “Connection Settings” tab, all general settings for the MODBUS connection are performed, such as the response timeout of requests and request intervals. Settings for “TCP Keepalive” specifically apply to MODBUS/TCP connections; “Type Of Service (IP)” applies to MODBUS-UDP and TCP connections.

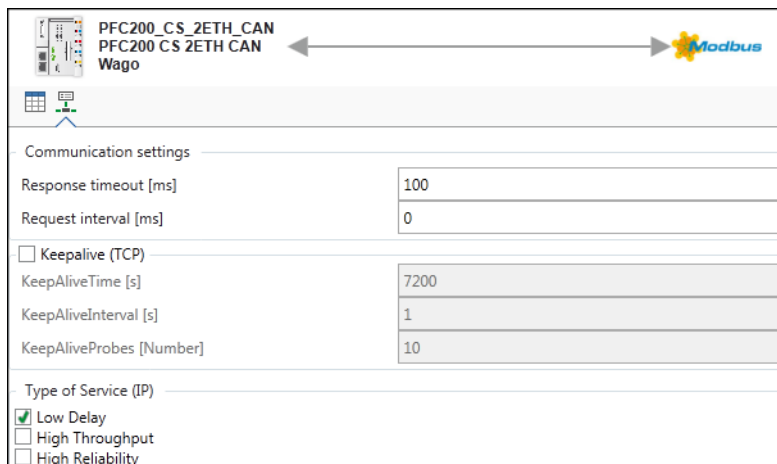


Figure 35: “Connection Settings” Tab

Table 26: Key to “Connection Settings” Tab

Parameter	Description
Communication Settings	
Response timeout [ms]	Wait time in ms until a request is repeated
Request interval [ms]	Time in ms between two successive requests
Keepalive (TCP)	
KeepAliveTime [s]	The time span in seconds (once) between the last data packet and the first KeepAlive packet (probe)
KeepAliveInterval [s]	Interval in seconds between two successive Keepalive packets (Probes)
KeepAliveProbes [Number]	Number of Keepalive packets until the connection is considered lost/removed
Type of Service (IP)	
Low Delay	Sets the “Low Delay” attribute in the IP header
High Throughput	Sets the “High Throughput” attribute in the IP header
High Reliability	Sets the “High Reliability” attribute in the IP header

MODBUS Slave Configurator

The MODBUS Slave Configurator is used to configure the MODBUS-accessible variables.

The left-hand column contains a tabular overview of all variables published via MODBUS. The variables present in the slave application are mapped in the right-hand column.

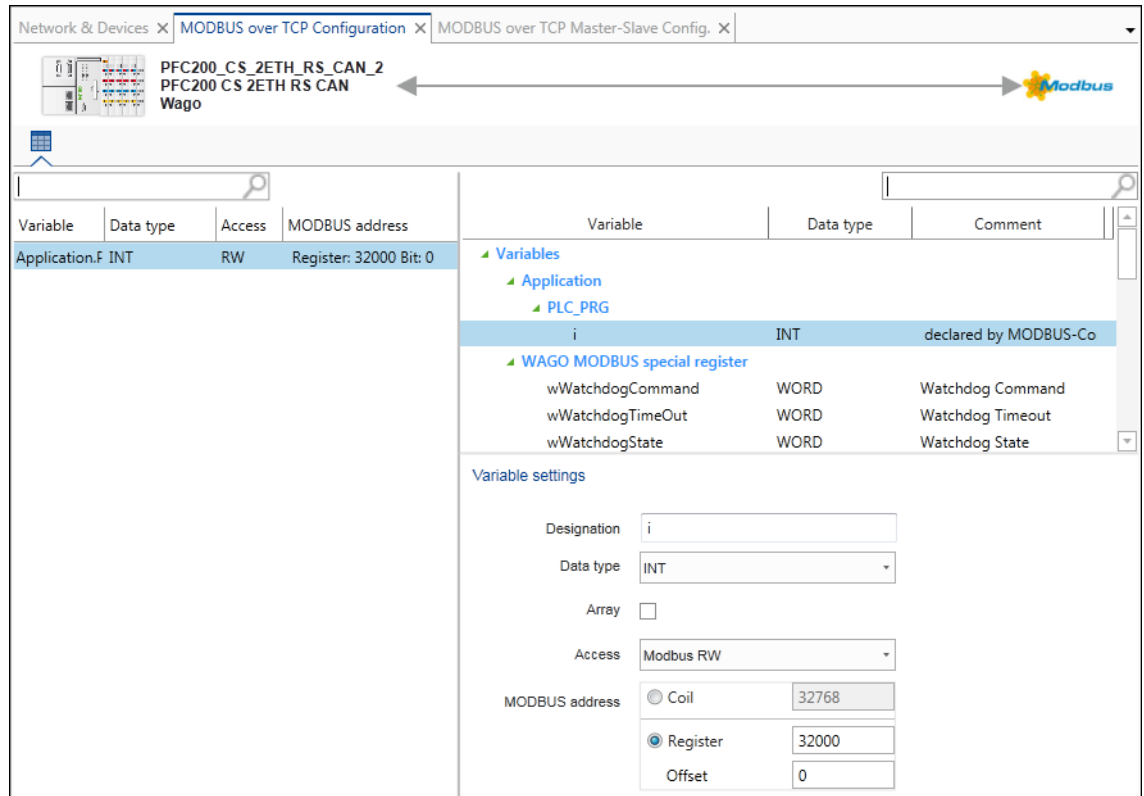


Figure 36: MODBUS Slave Configurator

The function of the right-hand column is identical to the function in the MODBUS master configurator. A variable can be deleted in the left-hand column, otherwise the column is only for informational purposes.

8.2.3.3.2 CANopen Configurator

“CANopen Remote Device” Tab

This tab sets the parameters in order to configure the sending and receiving of messages of other CANopen devices, such as the cyclic polling or monitoring of devices.

The screenshot displays the 'CANopen Remote Device' configuration window. At the top, there are tabs for 'Send PDO Mapping', 'Service Data Object', and 'CANopen I/O Mapping'. Below these are sub-tabs for 'CANopen Remote Device', 'PDO Mapping', and 'Receive PDO Mapping'. The 'General' section includes a 'Node ID' field set to '1', 'SDO Channels (1/1 active)', and the 'CANopen' logo. It also features checkboxes for 'Enable Expert Settings' (checked), 'Optional Device', 'Create all SDOs', 'No initialisation', 'Enable Sync Producing', and a 'Reset Node:' dropdown. The 'Nodeguarding' section has 'Enable Nodeguarding' unchecked, 'Guard Time (ms):' set to '0', and 'Life Time Factor:' set to '0'. The 'Heartbeat' section has 'Enable Heartbeat Producing' unchecked, 'Producer Time (ms):' set to '0', and 'Heartbeat Consuming (0/1 active)'. The 'Emergency' section has 'Enable Emergency' checked and 'COB-ID:' set to '\$NODEID+16#80'. The 'TIME' section has 'Enable TIME Producing' unchecked, 'COB-ID (Hex): 16# 100', and 'Enable TIME Consuming' unchecked. The 'Checks at Startup' section has 'Check Vendor ID' checked, and 'Check Product Number' and 'Check Revision Number' unchecked.

Figure 37: “CANopen Remote Device” Tab (view with “Expert Settings” checkbox enabled)

Table 27: Key for Figure “CANopen Remote Device Tab”

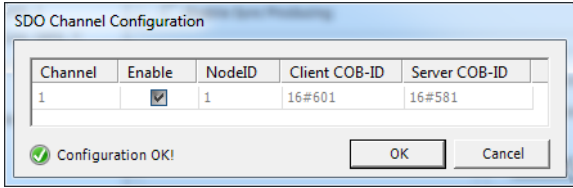
Parameter	Description												
General													
Node ID	Selects the node ID or station address (1 ... 126) that the PFC uses to communicate with the slave in the CAN network.												
[SDO Channels]	<p>Opens the “SDO Channel Configuration” window for configuring the service data objects (SDO). With some devices, this can be used to configure the communication parameters for the SDO transfer.</p>  <p>Figure 38: SD Channel Configuration</p> <table border="1"> <thead> <tr> <th>Column</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>Channel</td> <td>Channel number</td> </tr> <tr> <td>Activate</td> <td>Activate channel</td> </tr> <tr> <td>NodeID</td> <td>Additional node ID for SDO transfers (e.g., a channel for communication with the master and a channel for the communication with another slave)</td> </tr> <tr> <td>Client COB-ID</td> <td>Resulting “Communication Object ID” of the client</td> </tr> <tr> <td>Server COB-ID</td> <td>Resulting “Communication Object ID” of the server</td> </tr> </tbody> </table>	Column	Description	Channel	Channel number	Activate	Activate channel	NodeID	Additional node ID for SDO transfers (e.g., a channel for communication with the master and a channel for the communication with another slave)	Client COB-ID	Resulting “Communication Object ID” of the client	Server COB-ID	Resulting “Communication Object ID” of the server
Column	Description												
Channel	Channel number												
Activate	Activate channel												
NodeID	Additional node ID for SDO transfers (e.g., a channel for communication with the master and a channel for the communication with another slave)												
Client COB-ID	Resulting “Communication Object ID” of the client												
Server COB-ID	Resulting “Communication Object ID” of the server												
Enable Expert Settings	When enabled, the following expert functions described are displayed												
Generate all SDOs	<p>When enabled, SDOs are generated for all objects from the EDS file In addition, the corresponding options must be activated. If writing nodeguarding objects is required, the checkbox for the “Nodeguarding” option must be enabled. When disabled, SDOs are only generated for the objects with default values that deviate from the EDS file.</p>												
Enable Sync Producing	When enabled, synchronization messages are generated for synchronizing the process images between devices via this slave.												
Optional Device	When enabled, the CAN network is also “OPERATIONAL” (ready for communication via PDOs), if the device is not available.												

Table 27: Key for Figure “CANopen Remote Device Tab”

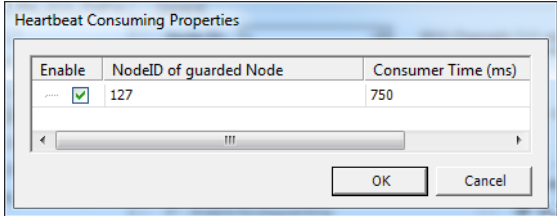
Parameter	Description
No Initialization	When enabled, no parameterization of the slave is performed by the master.
Reset Node	When enabled, the slave is reset by the PFC before the configuration is sent to the slave.
Nodeguarding	
Enable Nodeguarding	When enabled, the slave monitors the PFC for possible interruptions during fieldbus communication.
Guard Time (ms)	Selects the interval in which the PFC expects “Confirmation” from the slave.
Life Time Factor	Sets the multiplication value for the “Guard Time.” If the time resulting from “Guard Time” x “Life Time Factor” (“Node Life Time”) has elapsed, the slave is switched to the predefined state.
Heartbeat	
Enable Heartbeat Production	When enabled, the CAN device produces heartbeats at the ms intervals entered at “Producer Time.”
Producer Time (ms)	
[Heartbeat Consuming]	<p>Opens the “Heartbeat Consuming Properties” dialog.</p> 
	<p>Figure 39: Heartbeat Consuming Properties</p> <p>When enabled, the CAN device monitors the “Heartbeat” of the device selected by Node ID. The Master Heartbeat is entered automatically as standard. Either “Heartbeat” or “Nodeguarding” can be used for monitoring.</p>
Emergency	
Enable Emergency	When enabled, the slave sends error and status messages (Emergency messages). When disabled, the SDO 0x1014 is not transferred to the slave. The standard setting of the slave therefore continues to be valid.
COB-ID	CAN identifier, preset: Node-ID + 0x80

Table 27: Key for Figure “CANopen Remote Device Tab”

Parameter	Description
TIME	
Enable TIME Production	When enabled, the slave sends “Time” messages for synchronizing network devices.
COB-ID (Hex)	CAN identifier, preset: Node-ID + 0x80
Enable TIME Consumption	The slave receives “Time” messages.
Checks at Startup	
Check Vendor ID	Manufacturer ID of the CAN device
Check Product Number	Product number of the CAN device
Check Revision Number	Revision number of the CAN device

“PDO Mapping” Tab

Process data objects (PDO) transport data. These are divided into send and receive objects (TPDO and RPDO). This tab displays and performs the mapping between process data objects (PDO).

The receive PDO is selected in the left-hand area, and the send PDO (from the point of view of the controller) in the right-hand area.

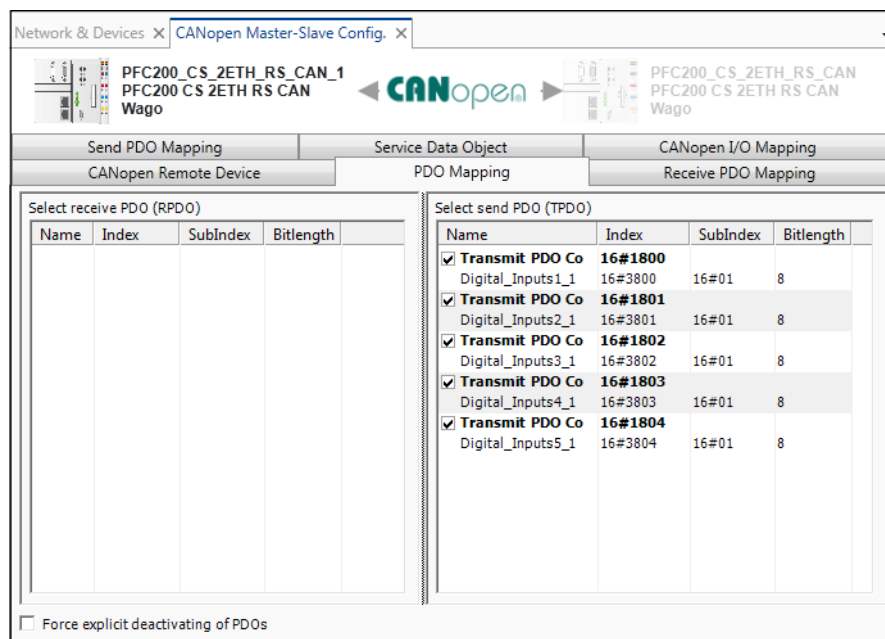


Figure 40: “PDO Mapping Tab”

Table 28: Key for Figure “PDO Mapping Tab”

Column	Description
Name	Name of PDO
Index	Index of PDO
SubIndex	Subindex of PDO
Bit length	Bit length of the PDO
Force explicit deactivating of PDOs	When enabled, the displayed PDOs are no longer used for communication.

“Receive PDO Mapping”/“Send PDO Mapping” Tab

These tabs are displayed if the “Expert Settings” option is enabled in the “CANopen Remote Device” tab.

They display the receive and send PDOs.

“Service Data Object” Tab

This tab displays the Service Data Objects (SDO). Additional SDOs can also be created for parameterization. SDOs transfer parameters to the device configuration.

On initialization of the CAN bus, the set values will be transmitted to the slaves as SDOs.

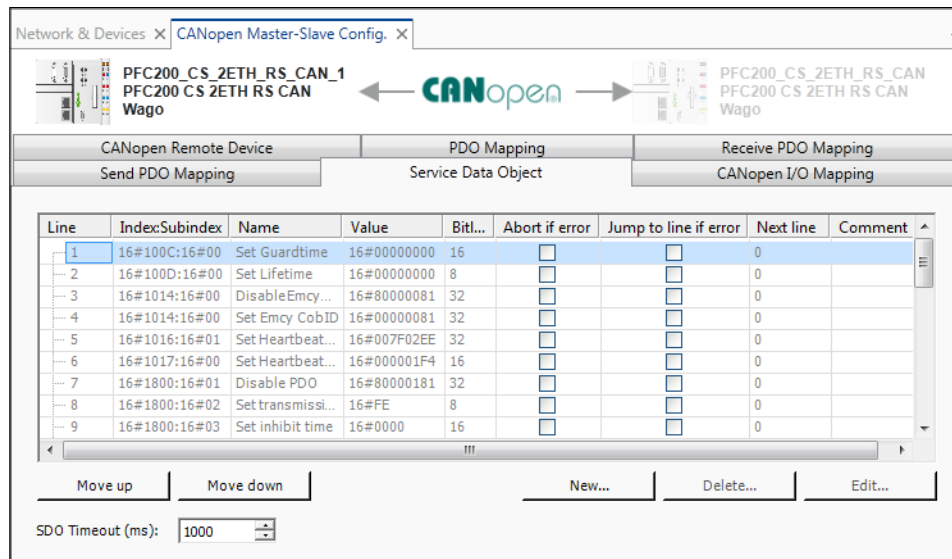


Figure 41: “Service Data Object Tab”

Table 29: Key to Figure “Service Data Object Tab”

Column	Description
Line	Line number
Index:Subindex	Index and subindex of the SDO
Name	Name of the service data object
Value	Value of the service data object

Table 29: Key to Figure “Service Data Object Tab”

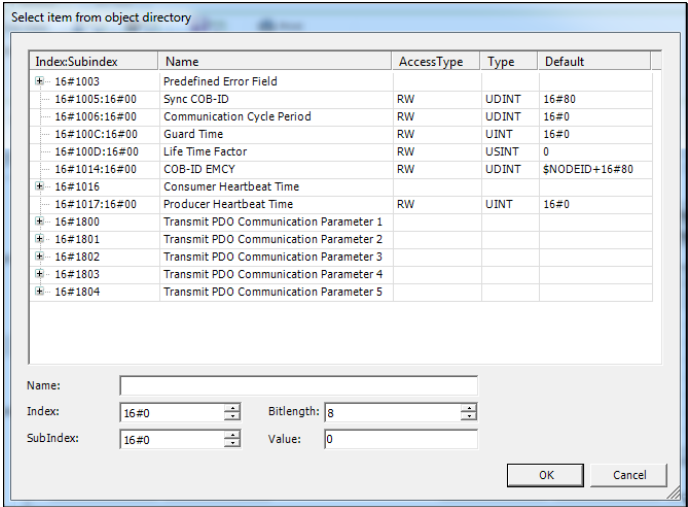
Column	Description
Bit length	Bit length of the service data object
Abort if error	Not used
Jump to line if error	Not used
Next line	Not used
Comment	Displays the comment Can be changed by selecting and overwriting the cell
[Move up]	Moves the entry up
[Move down]	Moves the entry down
[New...]	<p>Opens a dialog window for adding an existing service data object from the list Additional new service data objects can be created or selected service data objects changed in the lower area. Adding/changing is performed by double-clicking an entry or via the [OK] button.</p> 
[Delete...]	Deletes one or several selected service data objects
[Edit...]	Opens a dialog for editing a selected service data object
SDO Timeout (ms)	Not used

Figure 42: Adding Service Data Objects

Parameters	Description
Index:Subindex	Index and subindex of the SDO
Name	Name of SDO
Access Type	Displays which objects can be changed (RW)
Data type	Data type of the SDO
Default	Value from the EDS file

“CANopen I/O Mapping” Tab

The tab displays the mapping of CODESYS IEC objects to CAN variables.

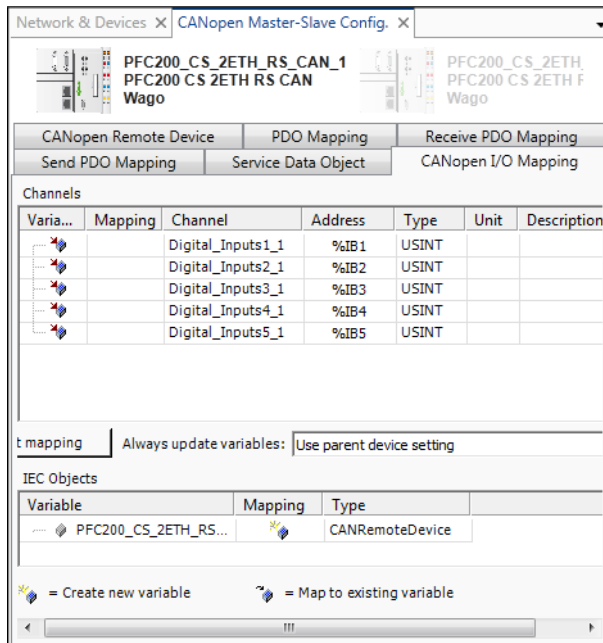




Figure 43: “CANopen I/O Mapping” Tab

Table 30: Key to Figure “CANopen I/O Mapping” Tab

Column/Button	Description
Variable	Project variable
Mapping	Displays which variable is new and which variable already exists:
	 The variable is not yet available, is newly created and can then be used in the entire project.  Uses an already existing variable for mapping
Channel	Channel name
Address	Variable address
Type	Variable data type
Unit	Unit of the variable
Description	Variable description

Configuring the CANopen Slave:

The Configurator is opened after a slave is selected using the appropriate button in the “Settings” panel.

General Configuration

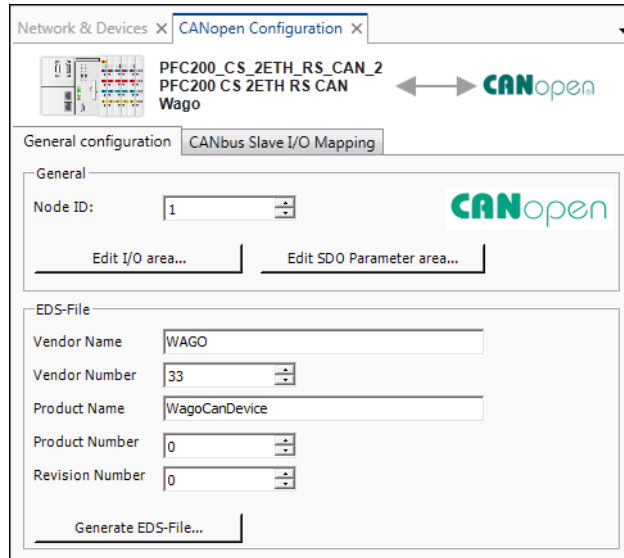


Figure 44: CANopen Slave Configuration

Table 31: General Configuration

Parameter	Description
General	
Node ID	Selects the node ID or station address (1 ... 126) that the PFC uses to communicate with the slave in the CAN network.
[Edit I/O area...]	<p>Opens the dialog for creating I/O areas or process data objects and the corresponding object directory indices (slaves).</p> <p>The possible number and size of the I/O areas depends on the number of unused PDOs. The actual number of Used TxPDOs and Used RxPDOs is displayed below the table (max. 16 PDOs).</p>

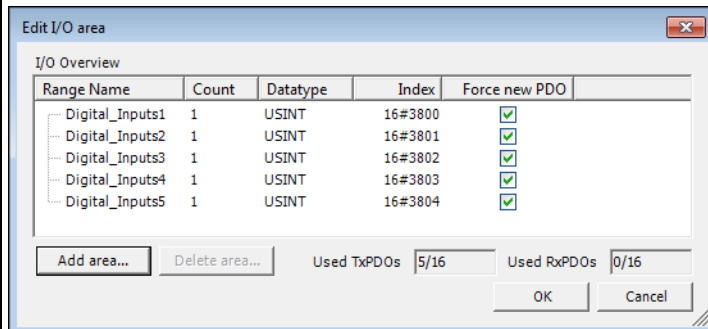


Figure 45: Edit I/O Area

Table 31: General Configuration

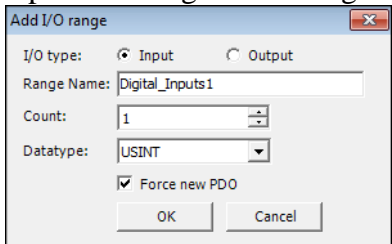
Parameter	Description	
	Range Name	Displays the unique name of the range This is automatically entered but can be changed.
	Count	Displays the number of inputs/outputs. The possible values depend on the data type and the actual number of PDOs.
	Datatype	Displays the data type of the process data object: USINT, UINT, UDINT, ULINT, SINT, INT, DINT, LINT, REAL, LREAL
	Index	Displays the index in the object dictionary of the slave that is assigned to the new object
	Force new PDO	If this option is enabled (default), each new I/O area is mapped to a new PDO — even if the existing PDO mapping is not yet full. Otherwise, the actual PDO mapping is completely filled in before a new PDO is created.
	[Add area...]	<p>Opens a dialog for defining a range</p>  <p>Figure 46: Add I/O Range</p>
	[Delete area...]	Deletes ranges and associated data (e.g., mapping variables, PDOs, PDO mapping) of the selected entries
	Used TxPDOs/Used RxPDOs	Displays the current number or receive/send process data objects The possible number and size of the I/O areas depends on the number of unused PDOs.

Table 31: General Configuration

Parameter	Description												
	<p>[OK]</p> <p>Activates the new I/O range and displays it in the I/O overview table Input/output channels and mapping variables are created. In the device's object dictionary, each I/O range receives its own index (output range: 0x3000 ... 0x30FF, input range: 0x3800 ... 0x38FF) and each individual input/output within the input range of a subindex. When an I/O range is created, PDOs are automatically created (RxPDOs start with 0x1400, TxPDOs with 0x1800). If the “Force New PDO” option is enabled (see above), the inputs/outputs are automatically mapped sequentially to these new PDOs.</p>												
	<p>[Cancel]</p> <p>Closes the dialog and discards any changes</p>												
[Edit SDO Parameter area...]	<p>Opens a dialog for creating configuration data objects The description of the dialog is the same as that of “[Edit I/O area...].”</p> <div data-bbox="630 1041 1300 1422" data-label="Image"> <table border="1"> <thead> <tr> <th>Range Name</th> <th>Count</th> <th>Datatype</th> <th>Index</th> </tr> </thead> <tbody> <tr> <td>ParamRange1</td> <td>1</td> <td>USINT</td> <td>16#5800</td> </tr> <tr> <td>ParamRange2</td> <td>1</td> <td>USINT</td> <td>16#5801</td> </tr> </tbody> </table> </div>	Range Name	Count	Datatype	Index	ParamRange1	1	USINT	16#5800	ParamRange2	1	USINT	16#5801
Range Name	Count	Datatype	Index										
ParamRange1	1	USINT	16#5800										
ParamRange2	1	USINT	16#5801										
EDS File													
Vendor Name	Name of the device manufacturer												
Vendor Number	Number of the device manufacturer												
Product Name	Name of the product												
Product Number	Number of the product												
Revision Number	Revision number of the product												
[Generate EDS File...]	This function enables an EDS file to be generated from the slave configuration.												

Figure 47: Edit Parameter Range

CANbus Slave I/O Mapping

The window displays the I/O mapping of the slave side.

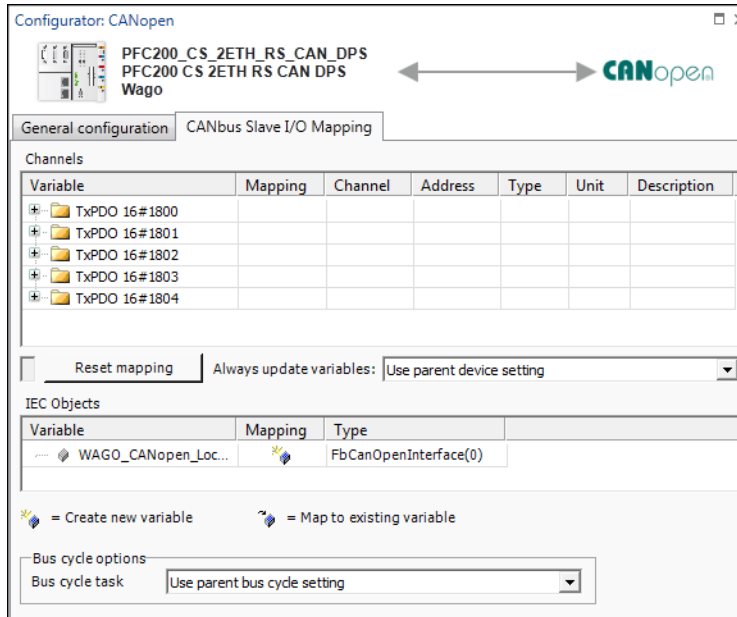




Figure 48: CANbus Slave I/O Mapping

Table 32: CANbus I/O Mapping

Column/Button	Description
Variable	Project variable
Mapping	Displays which variable is new and which variable already exists:
	 The variable is not yet available, is newly created and can then be used in the entire project.  Uses a pre-existing variable for mapping.
Type	Variable data type

Configuring the CANopen Master

The CANopen master configurator opens when the master device is selected and the appropriate button in the “Settings” panel is clicked. It contains two tabs: “CANopen Manager” and “CANopen I/O Mapping.”

CANopen Manager

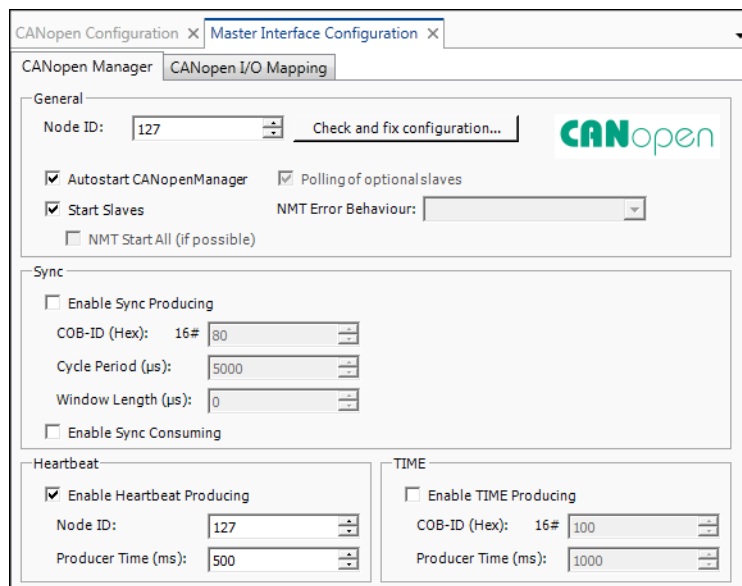


Figure 49: CANopen Manager

Table 33: CANopen Remote Device

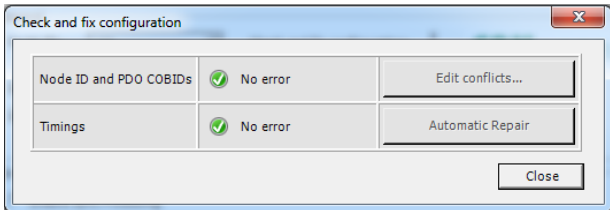
Parameter	Description
General	
Node ID	Selects the node ID or station address (1 ... 126) that the PFC uses to communicate with the slave in the CAN network
[Check and fix configuration...]	Opens a dialog for editing conflicts and for automatic repair 
Autostart CANopenManager	When enabled, starts the CANopen Manager automatically (OPERATIONAL) as soon as all obligatory slaves are ready. Otherwise, the CANopen Manager must be started by the application using the “CiA405 NMT” function block.

Figure 50: Check and Fix Configuration

Table 33: CANopen Remote Device


Parameter	Description
Start Slaves	When enabled, the CANopen Manager is responsible for starting the slaves. Otherwise the slaves must be started by the application using the “CiA405 NMT” function block.
Polling of optional slaves	When enabled, a slave that does not return a response during the boot sequence is polled regularly by the CANopen Manager until it returns a response. This option is currently enabled by default.
NMT Error Behavior	The NMT error behavior is not currently configurable.
NMT Start All (if possible)	The “NMT Start All” option cannot be selected at present.
Sync	
Enable Sync Production	When enabled, (disabled by default), the CANopen Manager transmits synchronization telegrams
	Note
	 Observe note on “Sync Production!” If “Sync Production” is enabled in the CANopen Manager, this function is disabled for all other bus stations.
	The following settings are available:
COB-ID (Hex)	CAN ID of the SYNC telegram Possible value range: [1, 2047]
Cycle Period (µs)	Interval length (in microseconds) after which the synchronization telegram is sent
Window Length (µs)	Not configurable
Enable Sync Consumption	When enabled, (disabled by default), the receipt of synchronization telegrams of other devices by the CANopen Manager is displayed
Heartbeat	
Enable Heartbeat Production	When enabled, the CAN device produces heartbeats at the ms intervals entered at “Producer Time.”
Node-ID	Unique identification (1–27) of the Heartbeat producer on the bus.
Producer Time (ms)	ms intervals in which heartbeats are sent

Table 33: CANopen Remote Device

Parameter	Description
TIME	
Enable TIME Production	When enabled, the slave sends a time stamp
COB-ID (Hex)	“Communication Object Identifier” which identifies the message's time stamp Default values: [0, 2047], default 100
Producer Time (ms)	Interval in milliseconds in which the time stamp is sent This value must be a multiple of the task cycle time.

CANopen I/O Mapping:

The window displays the master's I/O mapping.

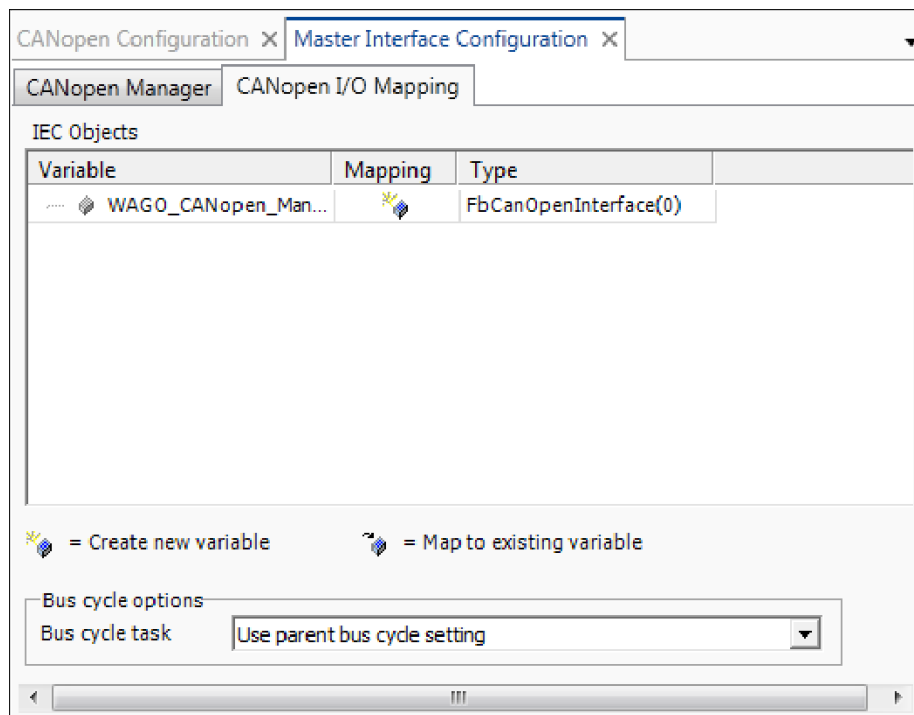


Figure 51: CANopen I/O Mapping

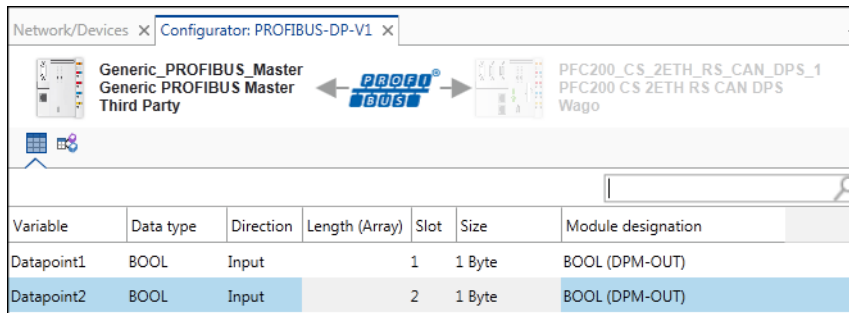
8.2.3.3.2.3 PROFIBUS Configurator

The PROFIBUS Slave Configurator creates data points for communication between the IEC program and PROFIBUS. The following functions are provided:

- Creation of data points
- Deletion of data points
- Mapping of data points to existing IEC variables (CODESYS configuration user interface)
- Creation of data points as new CODESYS variables (CODESYS configuration user interface); these can be addressed directly in the program editor.
- Export of the configuration in the form of a device description file (*.gsd or *.gsg).

The configuration user interface consists of two main areas: “Data Points” tab and “I/O-Mapping” tab, which are arranged beneath the device images.

The “Data Points” tab lists defined data points and creates or deletes new ones. The data points are shown in tabular form. Each column represents a property of the data point. After a data point is created, properties (interdependent) can be changed. The individual table cells serve as entry or selection windows.



Variable	Data type	Direction	Length (Array)	Slot	Size	Module designation
Datapoint1	BOOL	Input	1	1	1 Byte	BOOL (DPM-OUT)
Datapoint2	BOOL	Input	2	1	1 Byte	BOOL (DPM-OUT)

Figure 52: PROFIBUS Connection Configurator

Table 34: PROFIBUS Connection Configurator

Parameter	Description
Variable	Displays the data point name The name is always identical to the name of the IEC variable
Data type	Displays the data type of the data point, as well as the variable per IEC The following data types are supported: <ul style="list-style-type: none">• BOOL• BYTE• WORD• DWORD• SINT• INT• DINT• USINT• UINT• UDINT• REAL• ARRAY OF BOOL• ARRAY OF BYTE• ARRAY OF WORD• ARRAY OF DWORD• ARRAY OF SINT• ARRAY OF INT• ARRAY OF DINT• ARRAY OF USINT• ARRAY OF UINT• ARRAY OF UDINT• ARRAY OF REAL
Direction	Indicates whether it is input data or output data
Length (Array)	Displays the number of elements With simple data types (no arrays), the value is always set to 1. With arrays, the value can be changed.
Slot	Displays the slot's number Each data point is automatically assigned a slot. This cannot be changed. The data of a slot is always transmitted in a PROFIBUS telegram. This guarantees the consistency of the transferred data.
Size	Displays the size of the data point in bytes
Module designation	Displays the module name Each data point can be assigned a module name. If a user-defined name is not assigned, the name is generated from the data type and the number of elements. This is automatically updated with changes. A default name is also generated for deleted module names.

Right-clicking inside the table opens the context menu for adding (**New**), removing (**Delete**) or rearranging (**Up** / **Down**) data points. These commands are

also provided in the menu ribbon. The menu ribbon also provides commands for creating and duplicating several data points.

Variable	Data type	Direction	Length (Array)	Slot
Datapoint1	BOOL	Input		1
Datapoint2				2

Figure 53: Context Menu of a Data Point

The lower area of the PROFIBUS Configurator displays a gray status line. Clicking the text of the status line toggles between two views: “Used” and “Remaining.” The first view shows the slots, input bytes and output bytes already used. The second view shows the remaining bytes.

Variable	Data type	Direction	Length (Array)	Slot	Size	Module designation
Datapoint	BOOL	Input		1	1 Byte	BOOL (DPM-OUT)

Used: Slots: 1 Input: 1 Byte Output: 0 Bytes



Figure 54: Information in the Status Line of the PROFIBUS Configurator

The “I/O Mapping” tab performs mapping to the variables (if required).

Variable	Mapping	Channel	Address	Type	Unit	Description
Datapoint1		Slot1	%IX1.0	BIT		
Datapoint2		Slot2	%IX1.1	BIT		

Figure 55: PROFIBUS Connection Configurator

Table 35: PROFIBUS Configurator

Parameter	Description
Variable	Displays the name of the assigned or created variable. This name is visible in the IEC program.
Mapping	Displays which variable is new and which variable already exists:
	 The variable is not yet available, is newly created and can then be used in the entire project.  Uses an already existing variable for mapping
Channel	Displays the data point's name
Address	Displays the data point's IEC address
Type	Displays the variable's data type
Unit	Not used in this context
Description	Displays the description
[Reset mapping]	Deletes all mapping to existing variables
Always update variables	<p>Defines whether the I/O variables are updated in the bus cycle task. The default value is defined in the device description.</p> <p>Use parent device settings Updates the settings according to the settings of the parent device</p> <p>Activated 1 (use bus cycle task if not used in a task) Updates the I/O variables in the bus cycle task if they are used in no other task</p> <p>Activated 2 (always in the bus cycle task) Updates all variables in each cycle of the bus cycle task, regardless of whether they are used or whether they are mapped to an input or output channel</p>

8.2.3.3.3 Device Detail View

The Device Detail view is opened by double-clicking a device in the Network view. The Device Detail view is designed for each specific product series. It adds modules to a head station (via Drag & Drop from the Product Catalog), for simple cable tests on digital input/output modules (lamp test by clicking LEDs), as well as an entry point for configuring displayed devices.

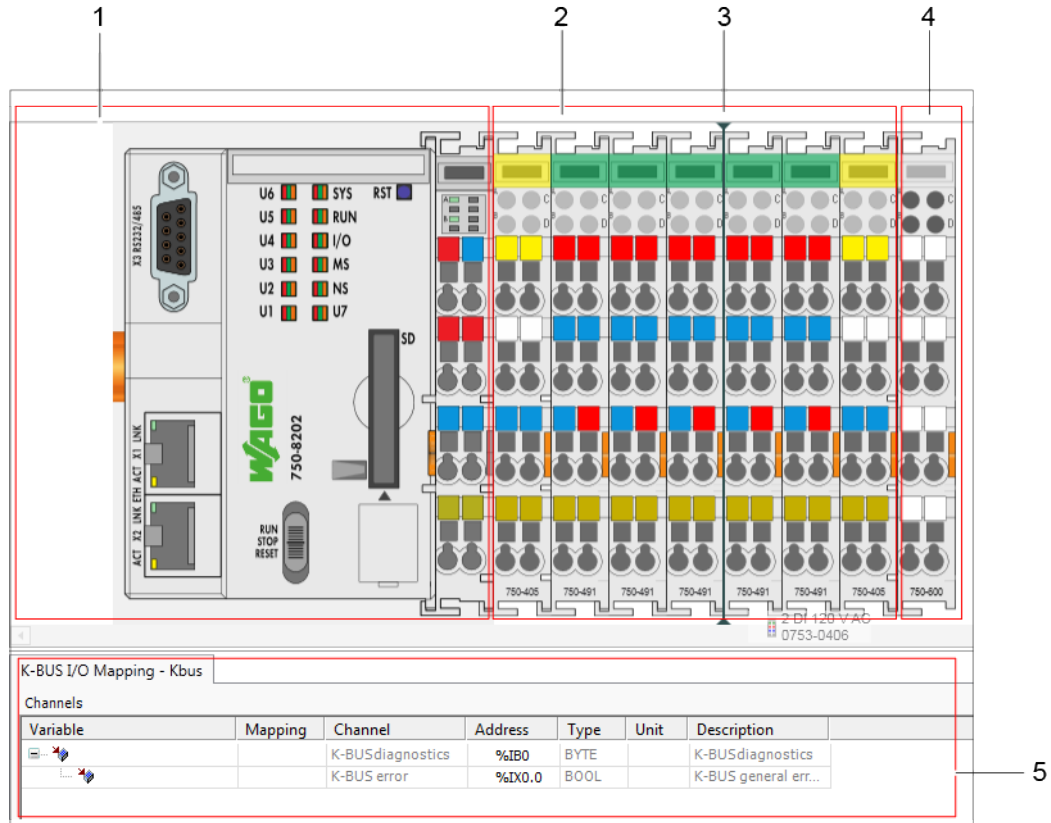


Figure 56: Device View (Example: Fieldbus Controller with Connected Modules)

Table 36: Elements of the Device View

No.	Description
1	Head Station/Controller Detailed view of a head station/controller in the project
2	Arranged Modules Detailed view of arranged modules To check the wiring, the LEDs of digital input/output modules can be switched on and off with a mouse click (if the device is connected online and control mode is active).
3	Positioning Line for Modules Position where new modules are added. Displayed if a module is dragged to the Device Detail view with the mouse.
4	End Module Display of the end module that terminates a node. End modules cannot be removed.
5	I/O Mapping Mapping and assignment of input, output, and memory addresses of the controller to project variables that are used by the application.

Right-clicking a device displays the different buttons of the context menu:

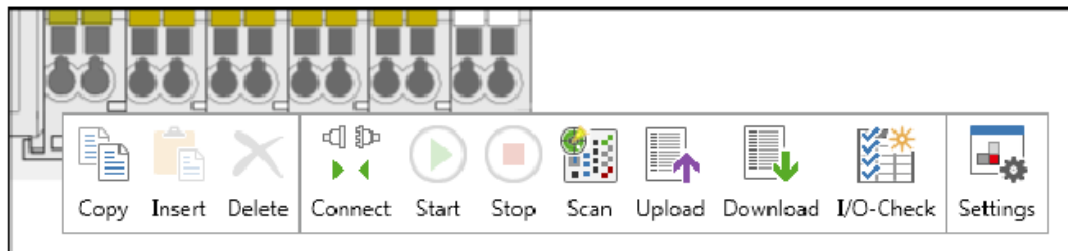














Figure 57: I/O Module Context Menu

Table 37: Context Menu of a Device

Symbol	Function	Description
	Copy	Copies selected modules
	Insert	Inserts copied modules
	Delete	Deletes selected modules from the node
	Connect	Opens an online connection to the node
	Disconnect	Removes the connection between the development and target systems
	Start	Starts the application on the controller or in the simulation
	Stop	Stops the application on the controller or in the simulation
	Scan	Scans the node for the presence of modules and displays results in a list
	Upload	Reads parameters from the displayed devices
	Download	Writes parameters to the displayed device
	I/O-Check	Opens the “I/O CHECK” tab (see Section “I/O CHECK Tab”)
	Settings	Displays device settings in the corresponding panel (see Section “Panels” > “Settings”)

The I/O mapping is displayed below the Device Detail view.

8.2.3.3.1 I/O Mapping

I/O mapping is the assignment of input, output, and memory addresses of the controller to project variables that are used by the application.

The display refers to the selected I/O module in the Device Detail view. If several I/O modules are selected, the I/O mapping of the last selected I/O module is displayed.

Variable	Mapping	Channel	Address	Type	Unit	Description
		_OUT	%QW0	WORD		Output Channels
Application.PLC_PR...		_OUT	%QX0.0	BOOL		Digital output
		_OUT	%QX0.1	BOOL		Digital output
		_OUT	%QX0.2	BOOL		Digital output

Figure 58: I/O Mapping

Table 38: I/O Mapping



Column	Value (example)	Description
Variable	“Input01”	Input field for the name of the variable to be used for mapping the channel in the CODESYS application. Double-clicking the input field displays the [...] button for opening the Input Assistant.
Mapping	 	Symbol which displays a new or existing variable. The variable is not yet available, is newly created and can then be used in the entire project. Uses an already existing variable for mapping
Channel	“Input channel 1”	Displays the symbolic name of the channel.
Address	“%IB3”	Displays the address of the channel in the following format: %<Memory area><Data type><Number(s)> Memory area: I Input/input memory area Q Output/output memory area M Marker memory area Data type: X Single bit None Single bit B Byte (8 Bit) W Word (16 Bit) D Double word (32 Bit) For additional information on I/O mapping and addressing, see the CODESYS Online Help, “I/O Mapping.”

Table 38: I/O Mapping

Column	Value (example)	Description
Type	“BYTE”	Displays the channel data type The table cell remains empty if non-IEC data types are involved.
Unit	“ms”	Displays the unit of the parameter value
Description	“Process data input channel 1”	Writes the parameter Edit the entry by double-clicking the entry field

8.2.3.4 “Programming” Workspace

The “Programming” workspace is similar to the CODESYS 3.0 programming environment.

Different editors are displayed in the middle of the workspace, and are opened by double-clicking on objects in the Project Structure or via the “Edit” button in the menu ribbon. The tab carries the name of the selected object.

The “Programming” workspace also contains its own panels, which can be used for programming (breakpoints, call stack, cross reference list).

8.2.3.4.1 Program Editor

The program editor is displayed in a tab if a program (e.g., PLC_PRG) is double-clicked in the Project Structure or the “Edit” button of the menu ribbon is clicked. For additional information, see the CODESYS online help.

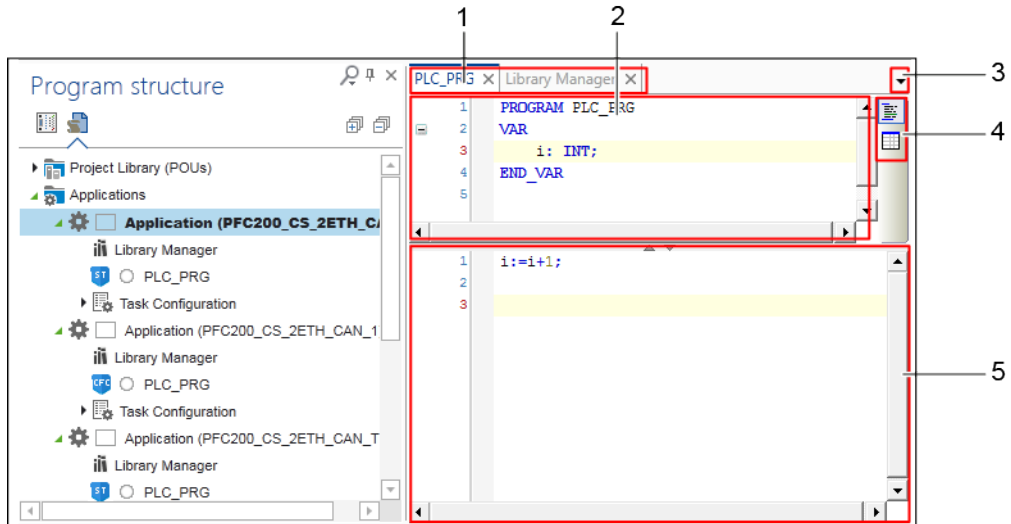




Figure 59: Elements of the Program Editor

Table 39: Elements of the Program Editor

No.	Description
1	<p>Tabs</p> <p>Program editors, library managers, visualizations are displayed based on the selection in the Program Structure (double-clicking). Right-clicking a tab opens a context menu that contains the following functions: “Close,” “Close all windows apart from this one,” “Undock/Dock”, “New horizontal/vertical tab group” and “Move to the next/previous tab group”.</p>
2	<p>Declaration Editor</p> <p>This area normally appears above the program code. It contains the program name, the scope of validity and variable declarations (see CODESYS online help: “Options, Declaration Editor”).</p>
3	<p>Active Tabs</p> <p>Shows active tabs in a selection list.</p>
4	<p>Text/Table Toggle Button</p> <p>Toggles between textual and tabular view (see CODESYS online help: “Options, Declaration Editor”)</p>
	<p> Textual View</p>
	<p> Tabular View</p>
5	<p>Language Editor</p> <p>The program code appears here when using language editors such as for “Structured Text” (ST) or “Continuous Function Chart” (CFC).</p>

8.2.3.4.2 Library Manager

Libraries contain collections of reusable objects such as functions, function blocks or variables that are incorporated into projects. The contents of these libraries can be used here without any new declaration. The library manager is used to manage the libraries of a project.

The following libraries are supported:

- **User-Related Libraries**, which contain ready-to-use software solutions for manufacturing, building and process automation
- **Function-Related Libraries**, which contain technology functions and simple applications, such as for opening a communication connection
- **WAGO and CODESYS System Libraries** which enable complete system access

Note



Only use system libraries from WAGO!

In order to use functions adapted to WAGO devices, only use WAGO's proprietary system libraries. You can recognize these libraries by the prefix “Wago” in the library name.

Information



Additional Information

Further notes on the libraries are provided in the manual “Libraries for *e!COCKPIT* (Overview and Migration Instructions)”.

The manual provides an overview of libraries that can be used in *e!COCKPIT*. It also provides instructions on importing software projects created in the WAGO-I/O-PRO, CODESYS-2 or CODESYS-3 programming environments. The appendix of this manual provides an overview of corresponding functions in libraries for WAGO-I/O-PRO and *e!COCKPIT*.

Additional information on converting CODESYS 2 and CODESYS 3 projects is provided in the CODESYS online help: (“Converting”).

Note



Permanently install libraries!

Note that all libraries must be incorporated permanently. If you simply replace the library files in your local library directory, the libraries are not available in *e!COCKPIT*. They must be installed from the repository.

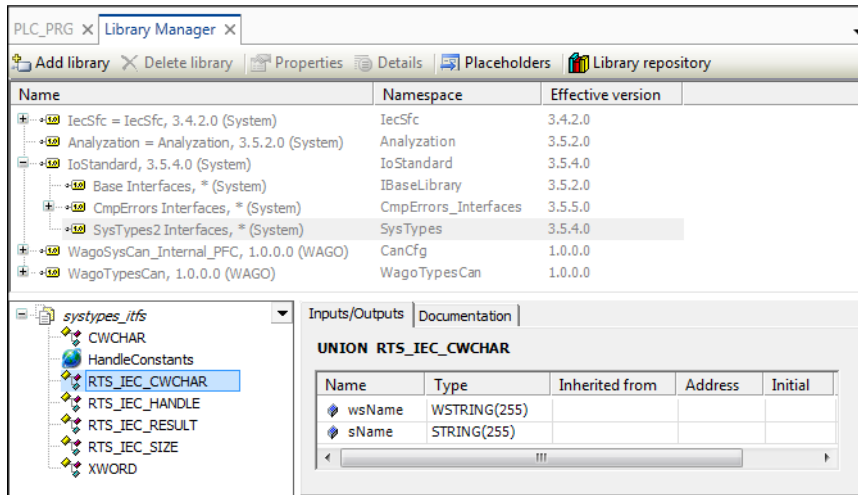


Figure 60: Library Manager

Table 40: Library Manager

Button	Description
[Add library]	Opens a dialog for library selection. The library manager adds to the project a reference to the library already installed in the repository.
[Delete library]	Deletes the reference to the selected library
[Properties]	Displays general properties of the selected library
[Details]	Displays details of the selected library
[Placeholders]	Displays placeholders used and adds target system-independent libraries. Depending on the device, the appropriate library is used; the standard library is used if no devices are available.
[Library repository]	Allows the addition and deletion of libraries from the Library Repository Libraries are permanently installed in <i>e!COCKPIT</i> .

8.2.3.4.3 Task Configuration

A task is a temporal sequence and control unit in an IEC program. It is defined by name, priority, type and a start condition. Either temporal conditions or an internal or external event can be defined for starting the task.

If a device is added to a project, a task configuration with a main task is automatically added for this device in the Program Structure (see the following Section). This call sequence is defined by the combination of priority and condition.

Time monitoring for each task can also be configured (watchdog). The setting options depend on the target system.

The task configuration is opened by double-clicking the appropriate entry in the Program Structure. The “Properties” tab shows the basic settings, such as maximum values for tasks and watchdog parameters.

The “Monitor” tab displays the status and current statistics for cycle times in online mode.

For additional information, see CODESYS online help: “Task Configuration Editor > Dialog, Properties.”

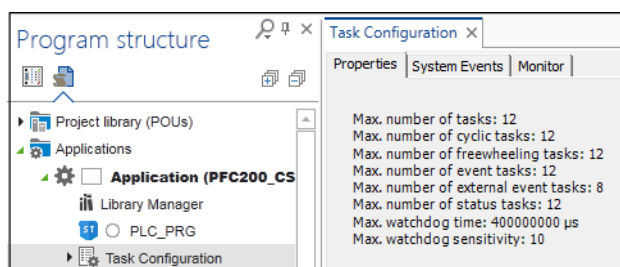


Figure 61: Task Configuration

8.2.3.4.4 “PLC_Task” Main Task

Configured tasks are displayed below the task configuration in the Program Structure. A “PLC_Task” main task is automatically created first for each task configuration. Tasks can be edited or deleted. The number behind the task indicates how many program calls are associated with the task. The task can be double-clicked to change the name. Other tasks can be extended by right-clicking them in the task configuration and choosing [Add New Element] > Task in the context menu.

For additional information, see CODESYS online help: “Task Configuration Editor > Configuration Dialog.”

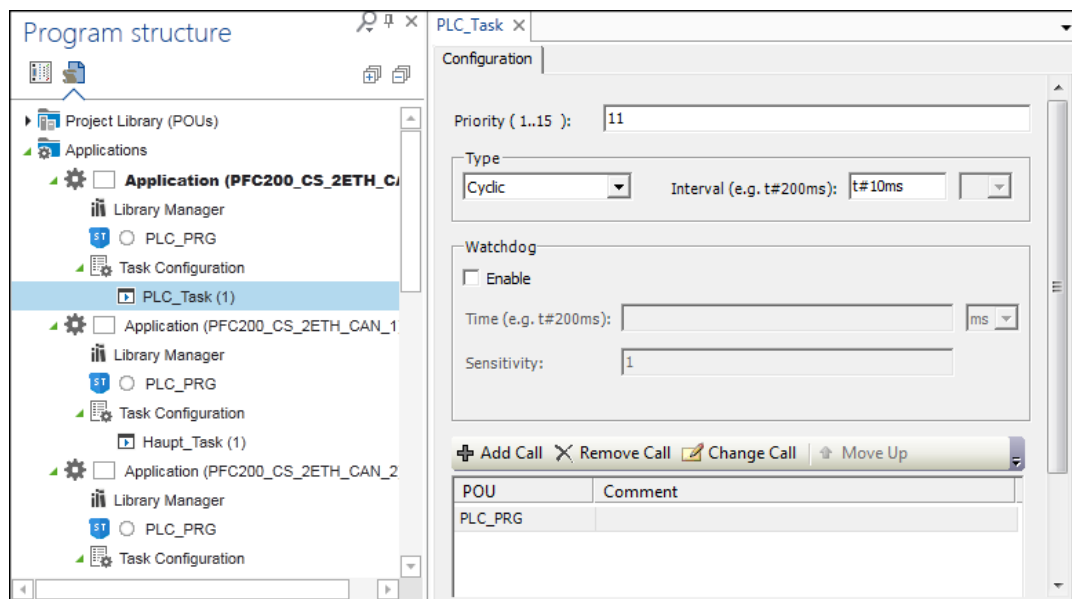


Figure 62: “PLC_Task” Main Task

Table 41: “PLC_Task” Main Task

Parameter	Description
Priority (1..15)	Indicates the task priority on a scale between 1 to 15 1 = highest priority 15 = lowest priority
	<div style="display: flex; align-items: center;"> <div style="margin-right: 10px;"></div> <div style="text-align: center;"> <p>Note</p> <p>Task priority can affect the runtime behavior of the controller!</p> <p>Note that a set task priority in the critical range (1 .. 3) can considerably affect the runtime behavior of the controller.</p> </div> </div>
“Cyclic” Type	Defines the time after which the task will be restarted. The cycle time (e.g., “200”) corresponds to the task cycle time entered in the “Interval” field in milliseconds “ms” or microseconds “μs.”

Table 41: “PLC_Task” Main Task



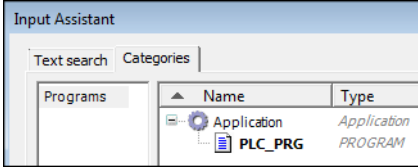
Parameter	Description
	 <p style="text-align: center;">Note</p> <p>Observe requirements for other bus systems! When entering the task cycle time, consider system requirements. For example, in CAN bus systems, this must fit the currently set baud rate and the number of frames used on the bus. To prevent the loss of CAN frames, also enter multiples of the task cycle time for Heartbeat, Nodeguarding and Sync.</p>
“Event” Type	<p>Starts the processing of the task as soon as a rising edge occurs on the variable defined in the “Event” text field The [...] button opens the Input Assistant with a list of all available global variables.</p>  <p style="text-align: center;">Note</p> <p>Use at least on a cyclic task for inputs! If the event that controls a task comes from an input, there must be at least one task that is not event-triggered. Otherwise the inputs and outputs are never updated and the event-controlled task is never started.</p>
“External” Type	Starts task processing as soon as the event defined in the “Event” field occurs (events dependent on the target system)
“Freewheeling” Type	Begins task processing at the start of the program and restarts automatically in a continuous loop at the end of the task (no defined cycle time)
“Status” Type	Starts task processing if the variable of the “Event” field has the value TRUE
Watchdog	Defines watchdog parameters for a task. Depending on the device used and definitions in the respective device description, setting options (upper/lower limits, default values, units, etc.) may be different than the settings shown. The watchdog can be switched off for specific cycles.

Table 41: "PLC_Task" Main Task

Parameter	Description
Enable	<p>If this option is activated, the watchdog is active. The task ends with an error status if the entered cycle time incl. set tolerance (sensitivity) is exceeded.</p> <p>Several successive timeouts: A sensitivity of 0 or 1 generates the exception error in cycle 1, a sensitivity of 2 to n generates the exception error in cycle 2 to n.</p> <p>With Single Timeout: Exception error if the cycle time of the current cycle is longer than (time * sensitivity) Example: Cycle time 10 ms and sensitivity 5 →Exception error as soon as the task (once) is longer than 50 ms.</p> <p>If this option is deactivated, the watchdog is deactivated. There is no task watchdog.</p>
[Add call]	<p>Opens a dialog for adding a program or POU call The dialog can likewise be opened in the Program Structure (right-click on a task > Program Call).</p>  <p>Figure 63: Selecting a Program via the Input Assistant</p> <p>For additional information, see CODESYS online help: "Program Call"</p>
[Remove Call]	Removes the selected program call
[Change Call]	Opens the Input Assistant for selecting another program or POU
[Move up]	Moves the selected program or POU up by one position in the processing sequence
[Move down]	Moves the selected program or POU down by one position in the processing sequence
[Open POU]	Opens the selected program in the programming editor

8.2.3.4.5 Breakpoints

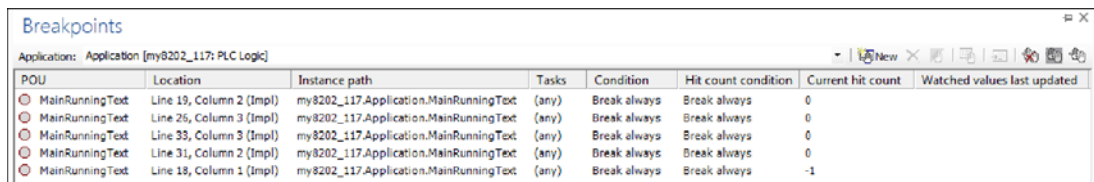
Breakpoints are used for debugging an application program. Breakpoints enable the program to be stopped at a defined position or executed in defined steps.

The breakpoint list is displayed in a window. This is opened via the menu ribbon, “VIEW” tab, [**Breakpoints**] button.

The “DEBUG” tab enables breakpoints to be set at the actual position in the Programming Editor ([**New**] button). It is also possible to toggle between the active and inactive state of a breakpoint ([**Toggle**] button).

Program symbols:

- Breakpoint active
- Breakpoint inactive
- 🚩 Stop at the breakpoint in online mode



POU	Location	Instance path	Tasks	Condition	Hit count condition	Current hit count	Watched values last updated
MainRunningText	Line 19, Column 2 (Impl)	my8202_117.Application.MainRunningText	(any)	Break always	Break always	0	
MainRunningText	Line 26, Column 3 (Impl)	my8202_117.Application.MainRunningText	(any)	Break always	Break always	0	
MainRunningText	Line 33, Column 3 (Impl)	my8202_117.Application.MainRunningText	(any)	Break always	Break always	0	
MainRunningText	Line 31, Column 2 (Impl)	my8202_117.Application.MainRunningText	(any)	Break always	Break always	0	
MainRunningText	Line 18, Column 1 (Impl)	my8202_117.Application.MainRunningText	(any)	Break always	Break always	-1	

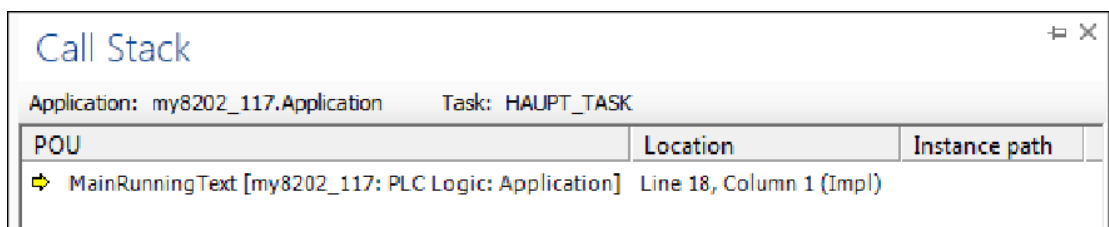
Figure 64: Breakpoints

Further information on creating and editing breakpoints can be found in CODESYS online help: (“Breakpoints”).

8.2.3.4.6 Call Stack

During the step-by-step execution of a program, this always shows the currently reached position in the call stack with a complete call path.

The “Call Stack” panel is opened as a panel via the “View” tab of the menu ribbon.



POU	Location	Instance path
🚩 MainRunningText	[my8202_117: PLC Logic: Application]	Line 18, Column 1 (Impl)

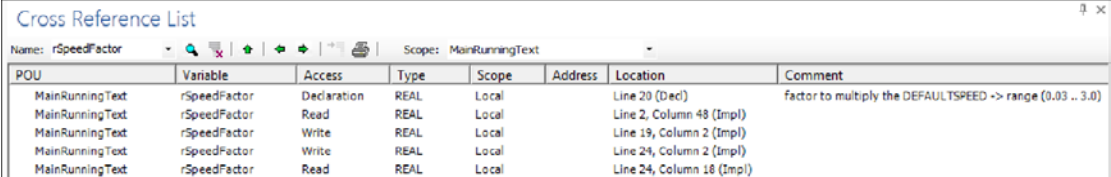
Figure 65: Call Stack

Further information on creating and editing call stacks can be found in the CODESYS online help: (“Call Stack”).

8.2.3.4.7 Cross Reference List

The cross reference list shows all occurrences of a variable in the project or the active editor.

The “Cross Reference List” panel is opened as a panel via the “View” tab of the menu ribbon.



POU	Variable	Access	Type	Scope	Address	Location	Comment
MainRunningText	rSpeedFactor	Declaration	REAL	Local		Line 20 (Decl)	factor to multiply the DEFAULTSPEED -> range (0.03 .. 3.0)
MainRunningText	rSpeedFactor	Read	REAL	Local		Line 2, Column 48 (Impl)	
MainRunningText	rSpeedFactor	Write	REAL	Local		Line 19, Column 2 (Impl)	
MainRunningText	rSpeedFactor	Write	REAL	Local		Line 24, Column 2 (Impl)	
MainRunningText	rSpeedFactor	Read	REAL	Local		Line 24, Column 18 (Impl)	

Figure 66: Cross Reference List

Further information on creating and editing cross reference lists can be found in CODESYS online help: (“Cross Reference List”).

8.2.3.5 User-Defined Workspaces

e!COCKPIT's graphical user interface can be configured as needed. Predefined workspaces are provided for two main tasks in a window layout (“Network/Devices” and “Programming”).

These workspaces can be customized with user-defined settings, additional panels, window divisions, etc. and saved as a user-defined workspace. Use the **[Manage]** button within the “START” tab for this. After saving, a new button is displayed in the “START” tab through which the new workspace can be accessed.

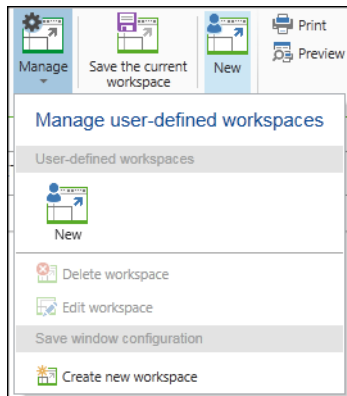


Figure 67: Creating and Managing User-Defined Workspaces

The information for user-defined workspaces is stored for each user in the local Windows properties.

If the default “Network/Devices” or “Programming” workspaces were changed and overwritten with **[Save the current workspace]**, the default settings can be restored at any time in the Backstage view, **[Options] > [Standard workspaces] > [Reset standard workspaces]**.

8.2.3.6 Panels

The workspace in *e!COCKPIT* consists of different windows; within these windows, the panels can be arranged as required.

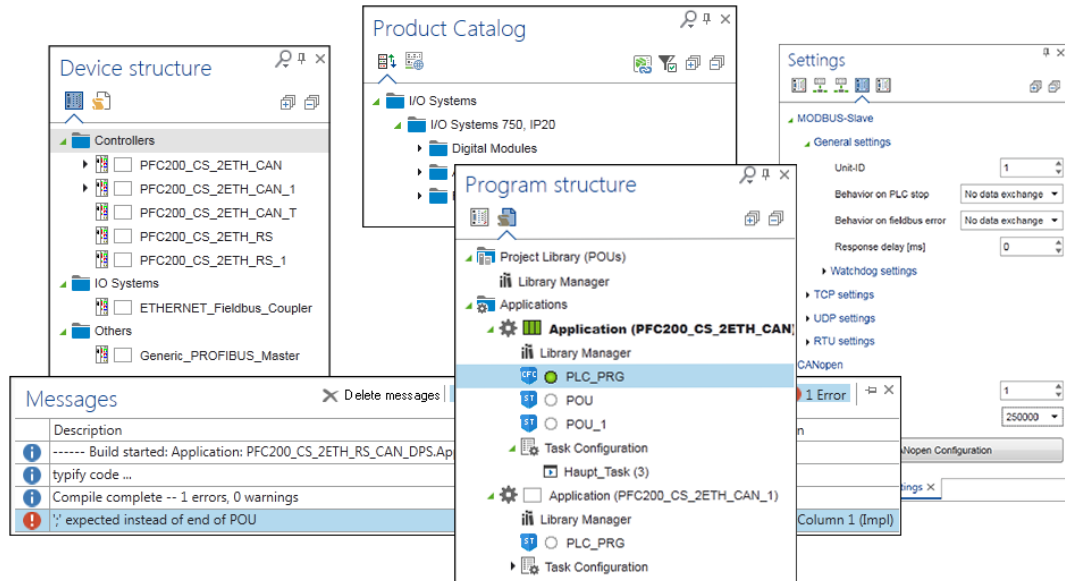


Figure 68: Panels

The “Network/Devices” workspace displays the “Device Structure,” “Product Catalog” and “Settings” panels by default. The “Programming” workspace displays the “Program Structure” panel. This and other panels, such as “Messages” or the panels related to programming such as “Cross Reference Lists,” can be displayed via the “VIEW” tab.

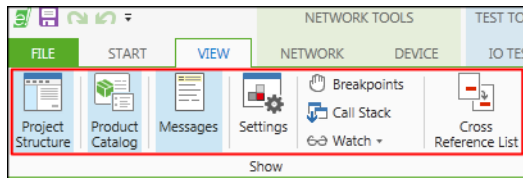


Figure 69: Displaying Panels

Panels can be docked, floated and grouped.

The selection panel synchronizes the selection in other windows or areas of the workspace. The commands and functions of individual tabs depend on the particular elements selected in the panels.

8.2.3.6.1 Device Structure

The Device Structure panel is used for display and providing navigation between the different devices integrated in a project. The panel is part of the “Network/Devices” workspace and is opened via the [Network/Devices] button in the “VIEW” tab.

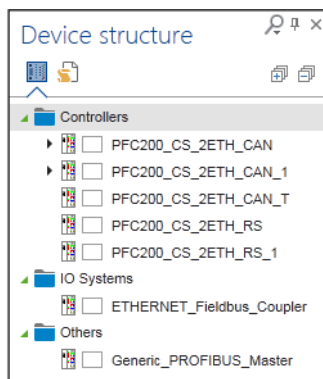


Figure 70: “Device Structure” Panel

The devices are organized in groups that are based on product segments of a Product Catalog, e.g., “Controllers,” or “I/O Systems.”

Device types are represented with different pictograms. A four-section display behind the device pictogram indicates a device’s status for the connection, PLC, internal bus and fieldbus (see Section “Workspaces” > “General Status Display for Connections”). This is followed by the device name based on the device description file. The name can be double-clicked in order to change it.

It is possible to toggle between Device Structure and Program Structure by clicking the tabs in the panel.

Other functions can be accessed via the context menu of a device. These functions are explained in the Section “Tiles.”

8.2.3.6.2 Program Structure

The “Program Structure” is used for display and providing navigation between the different devices integrated in a project. The panel is part of the “Programming” workspace and is opened via the [Programming] button in the “VIEW” tab.

The program elements are displayed by device group (“Application ([Device Name])”).

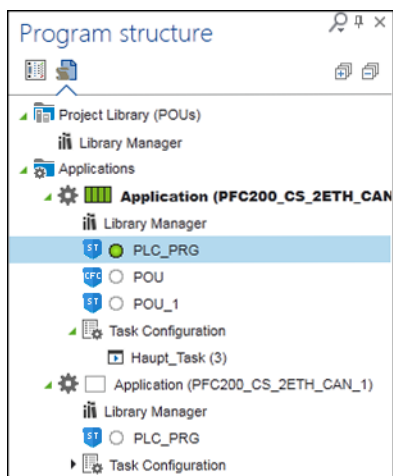


Figure 71: “Program Structure” Panel

A four-section status display provided in front of each application indicates the status with regard to connection, PLC, internal bus and fieldbus (see Section “General Status Display for Connections”).

A running program is indicated by a green circle icon displayed in front of it. Possible error states (such as compiler errors) are indicated with a red underline just as they are in CODESYS. Programs excluded from compiling are indicated in gray.

Right-clicking an element in the list opens a context menu that contains additional functions.

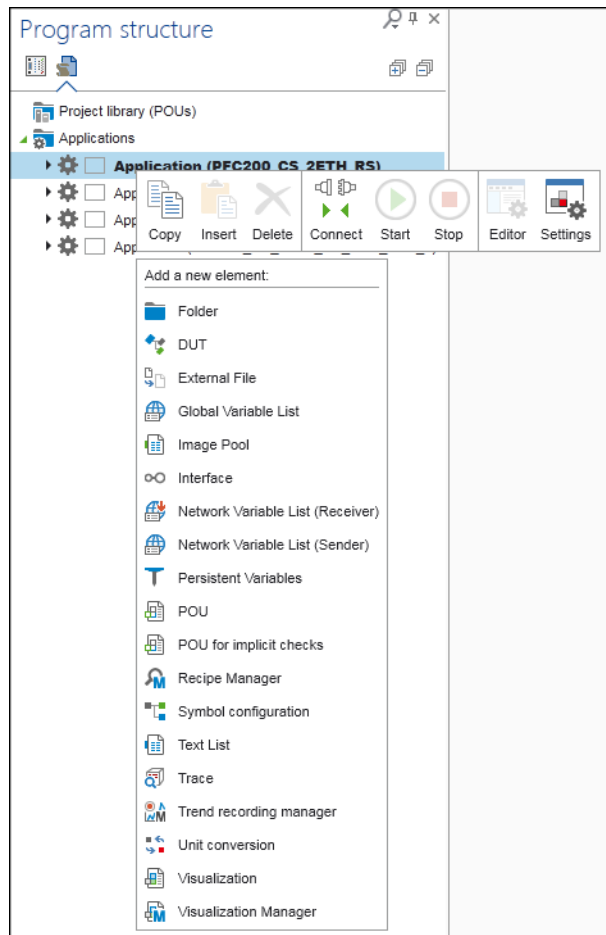











Figure 72: Context Menu of an Application

Table 42: Context Menu of an Application

Symbol	Function	Description
	Copy	Copies the selected element
	Insert	Inserts the copied element
	Delete	Deletes the selected element from the Program Structure
	Connect	Opens an online connection to one or several selected applications
	Disconnect	Removes the connection between the development and target systems (controller or simulated device) and returns to offline mode
	Start	Starts the application on the controller or in the simulation
	Stop	Stops the application on the controller or in the simulation
	Editor	Opens the programming editor for entering program code
	Settings	Opens a dialog in which program properties can be edited (see CODESYS online help: "Properties, General")

Depending on the element that is right-clicked, other CODESYS elements can be appended to the structure, e.g., other program organization units (POUs), visualizations or variable lists. For additional information, see the CODESYS online help: "Adding Objects."

8.2.3.6.3 Product Catalog

The Product Catalog displays all devices that are available in the system and have device drivers that are installed (Backstage view > **[Product Catalog]** > **[Add Devices]**). The devices can be sorted by product segment and protocol type.

Note



If the Network view is activated, only controllers are visible – not modules!
If Network view is active, no modules are shown in the Product Catalog. These are only displayed when Device Detail view is active. No internal WAGO modules (e.g., the internal bus) or end modules are shown.

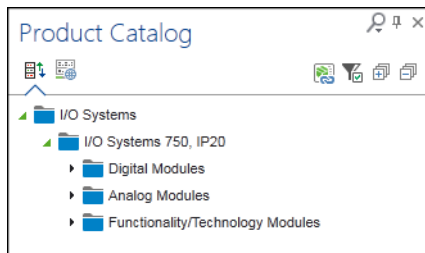


Figure 73: “Product Catalog” Panel

Table 43: Buttons of the “Product Catalog” Panel





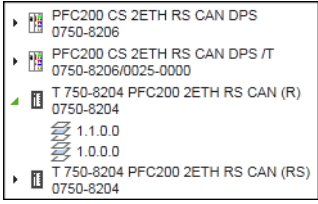


Button	Description
 [Sort by product segment]	Displays the products by segment: <ul style="list-style-type: none"> • Operation and monitoring • Controllers • I/O systems • Accessories • Modules • Basic network functions • Other communication partners
 [Sort by communication protocol]	Sorts the devices by protocol type, such as: <ul style="list-style-type: none"> • MODBUS • PROFIBUS • CANopen • Serial interface
 [Move to the product catalog management]	Moves to the “Product Catalog” page of the Backstage view for incorporating new devices ([Add devices] button)

Table 43: Buttons of the “Product Catalog” Panel

Button	Description
 [Display the available versions]	Displays in the tree structure all available device type versions of a device. This function can be switched on/off for each tab in the Product Catalog.  Figure 74: Displaying Available Versions
 [Expand]	Opens all elements of the tree structure (up to device level)
 [Collapse]	Closes all elements of the tree structure

Functions of the Product Catalog:

- To add a device to a project, individual devices are dragged from the Product Catalog and dropped onto a tile in the Network view.

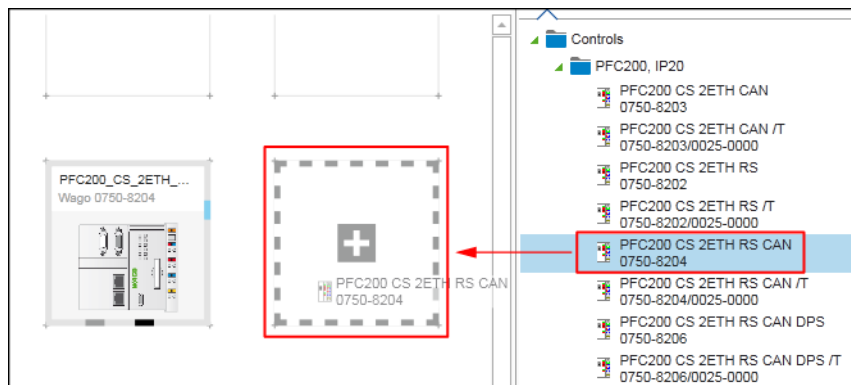


Figure 75: Dragging Devices to the Network View

- Hovering the mouse over a device will display a tooltip showing the product view, device name, type, order number of the device.

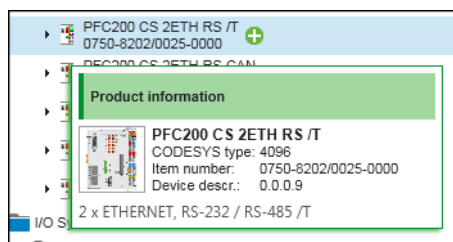


Figure 76: Displaying Product Information

- A green “+” symbol is also displayed. Clicking this symbol adds a device to the next free position in the project.

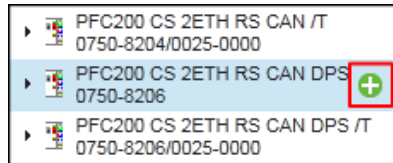


Figure 77: Adding a Device to a Project

8.2.3.6.4 Settings

Device parameters are selected in the “Settings” panel. The parameters can be entered within defined limits. Entries in the “Settings” panel depend on the device selection and are based on the relevant device description of the selected device.

Complex modules or devices with more extensive settings, e.g., those involving CANopen or PROFIBUS parameters, can be set using integrated configurators. The configurators are opened from this panel via the relevant buttons. The setting options for these device configurators, as well as an explanation of individual parameters, can be found in the manual of the respective device.

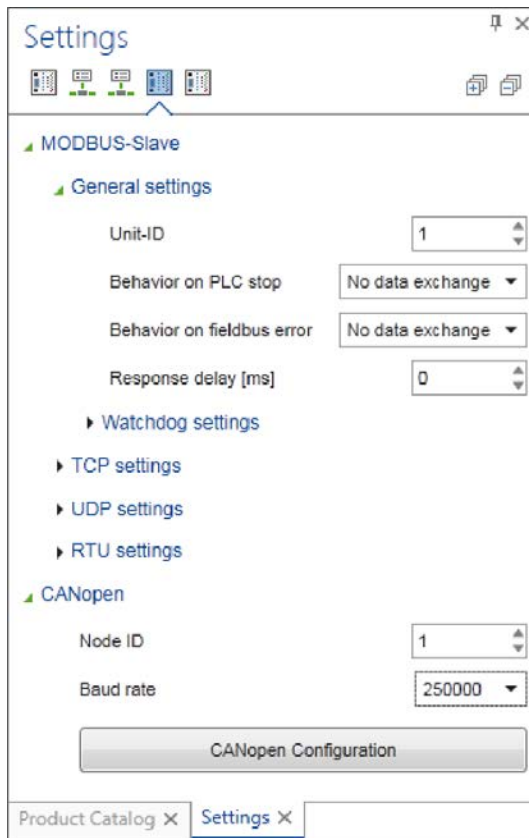


Figure 78: “Settings” Panel with the Button to a Device Configurator (CANopen)

If a head station is selected in Device Detail view or a device in Network view, the “Settings” panel also serves as an entry point for opening network configurators for MODBUS, CANopen and PROFIBUS connections. Network configurators are described in Section “Fieldbus-Specific Configurators”.

Other PLC settings can be carried out via the second tab.

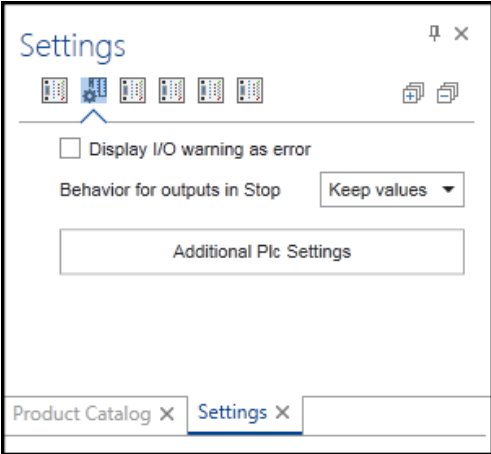


Figure 79: Additional Plc Settings

8.2.3.6.5 Message Bar and Message Window

e!COCKPIT features a message bar below the menu ribbon that displays any warnings, messages or errors during configuration, programming or validation. Up to three messages from different sources can be output. The info symbol on each line shows the meaning of the message: Error, warning or info.

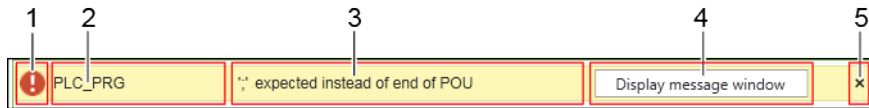


Figure 80: Message Bar

Table 44: Message Bar Elements

No.	Description
1	Message type “Errors”
1	Name of the affected element, e.g., the program “PLC_PRG”
2	Short message description
3	Button for opening the message window
4	Close current message and display next message

Error messages can refer, for example, to validations of the Device Detail or Network view. CODESYS error messages, configuration errors (incompatible I/O module), compiler errors, warnings, etc. can be displayed. The test is performed continuously.

The complete message list is displayed in a message window. This is opened via the **[Display Message Window]** button of the yellow message bar or alternatively via the menu ribbon, “VIEW” tab, **[Messages]** button.

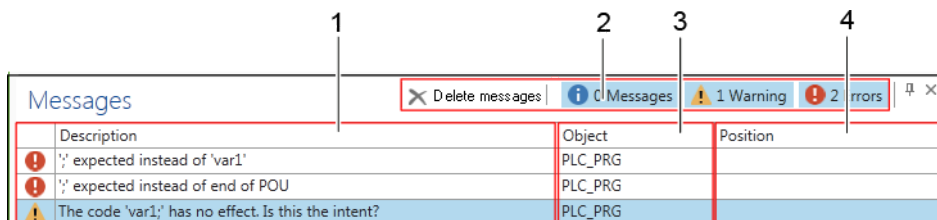


Figure 81: Message Window

Table 45: Elements of the Message Window

No.	Description
1	Description: Description of the message
2	Filter Messages/Warning/Errors filter are checked (blue background) or unchecked. [Delete messages] removes all messages from the list.
3	Object: Name of the project element
4	Position: Line in the source code

8.2.3.6.6 Other Panels

Besides the panels described, other device-specific panels may appear that are not included in this description.

The contents of these tabs result from the particular device description file (DTP) used.

The “Call stack,” “Breakpoints” and “Cross reference list” panels are described in greater detail in the context of programming, see Section “Programming Workspace”.

8.2.3.6.7 Status Bar

The status bar displays all relevant connection information of the currently selected device with regard to the connection, PLC, internal bus and the fieldbus.

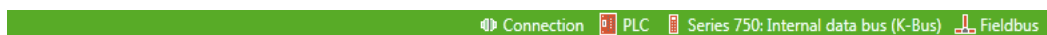


Figure 82: Status Bar

This information applies to one device. No status is displayed if several devices are selected. However, the connection status is displayed in the status bar if several applications of the same device with the same connection properties are selected.

The individual symbols are explained in the Section “Workspace” > “General Status Display for Connections”.

8.3 Backstage View (Files and Settings)

The project file itself, as well as other general basic settings of *e!COCKPIT*, are handled and managed via Backstage View, keeping these tasks separate from the processing of actual project content.

The buttons of the navigation bar in the left-hand window display the relevant pages in the right-hand window area or in a separate dialog. The individual pages are described in the following sections.

Open the Backstage view from the Start view via **[Other Projects]** or via the “FILE” tab of the main view.

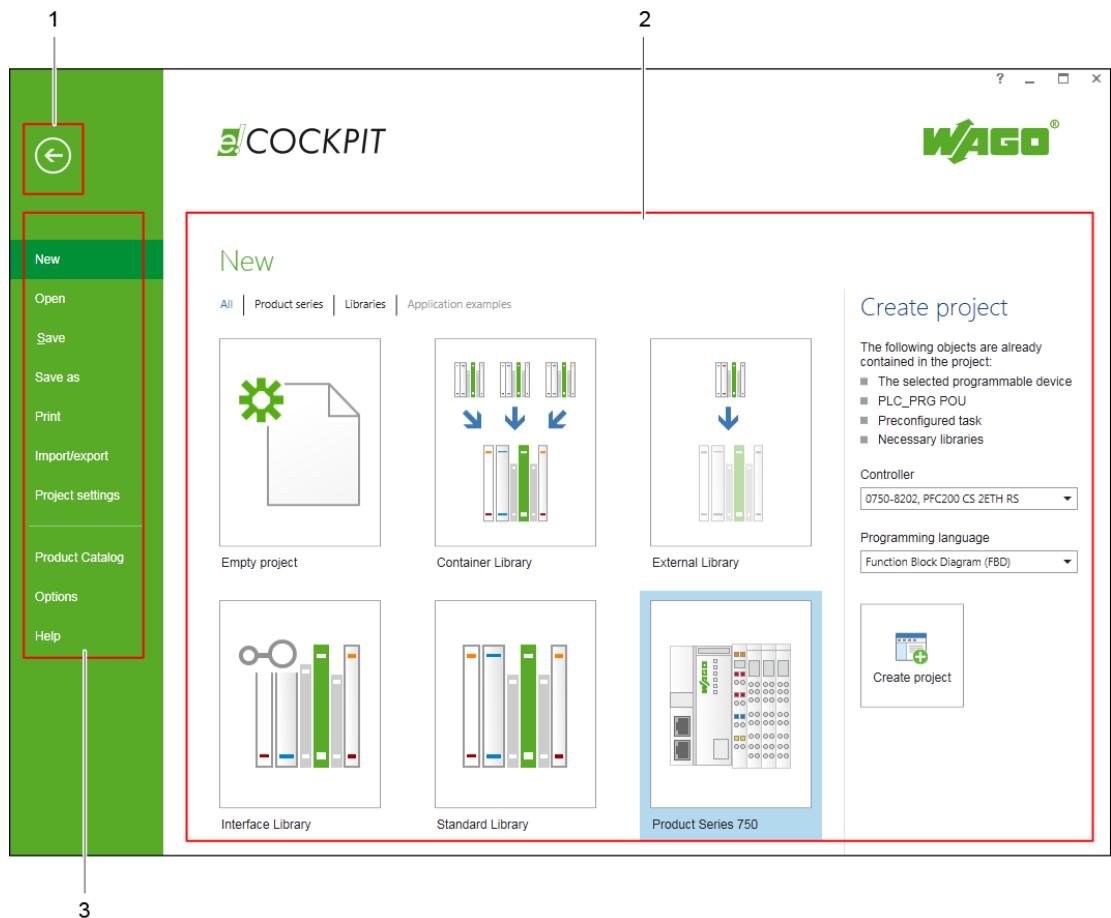


Figure 83: Backstage View

Table 46: Legend for Figure Backstage View

No.	Description
1	Back Returns to the previous view
2	Pages Shows the content for handling and managing files, as well as basic settings for <i>e!COCKPIT</i> (see following Section)
3	Navigation Bar Navigates via buttons to different pages that open in the right-hand window area

8.3.1 “New” Page

The “New” page creates a new project. Project templates provide special presets for creating a project.

Some templates in the right-hand window offer additional setting options, such as for creating a project with the “750 Series” template. This template creates a project with a fieldbus controller and an initial application. Create the project by clicking the [**Create Project**] button. The projects of templates that do not require any additional settings are created immediately when these templates are clicked.

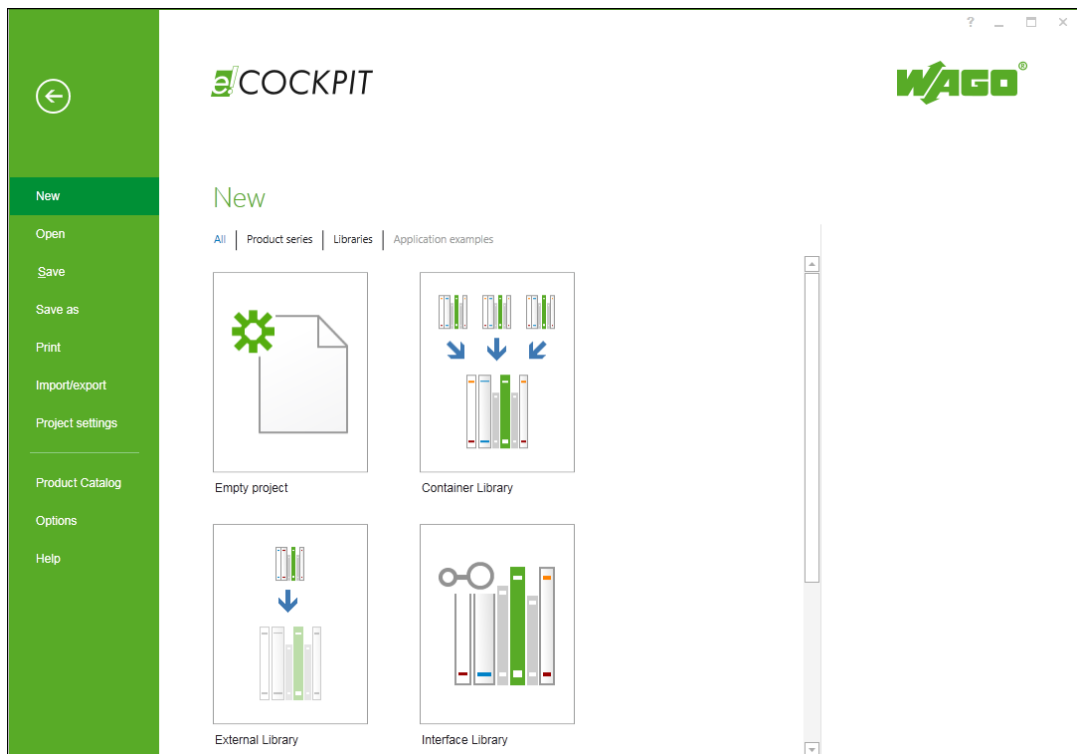


Figure 84: Backstage View, “New” Page

8.3.2 “Open” Page

The “Open” page finds and opens documents.

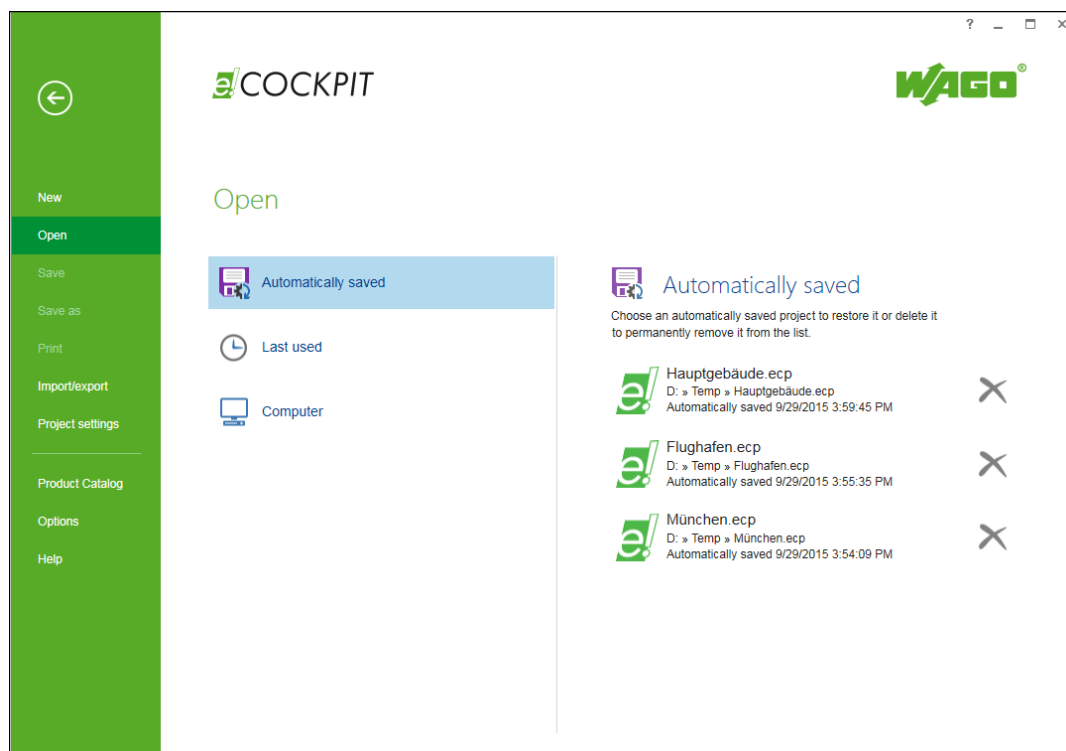


Figure 85: Backstage View, “Open” Page

Table 47: Backstage View Operating Elements, “Open” Page

Button	Description
[Automatically saved]	Displays (if present) automatically saved projects Besides manual saving (“Save”/“Save as” page), the project is also saved automatically by default every 10 minutes. The activation/deactivation of the automatic saving and setting of the save interval are made on the “Options” page.
[Last used]	Displays the last projects used Hovering the mouse over one of these projects will cause a gray pin to appear on the right. Clicking the pin will permanently identify the corresponding project in the list as a favorite.
[Computer]	Shows the last folder selected on the PC. [Browse] opens files in other locations.

8.3.3 “Save” and “Save As” Page

“Save” stores the project under the currently used file name and memory location. The “Save as” page selects either the last folder used or another location via **[Browse]**.

See “Options” for automatic saving of projects.

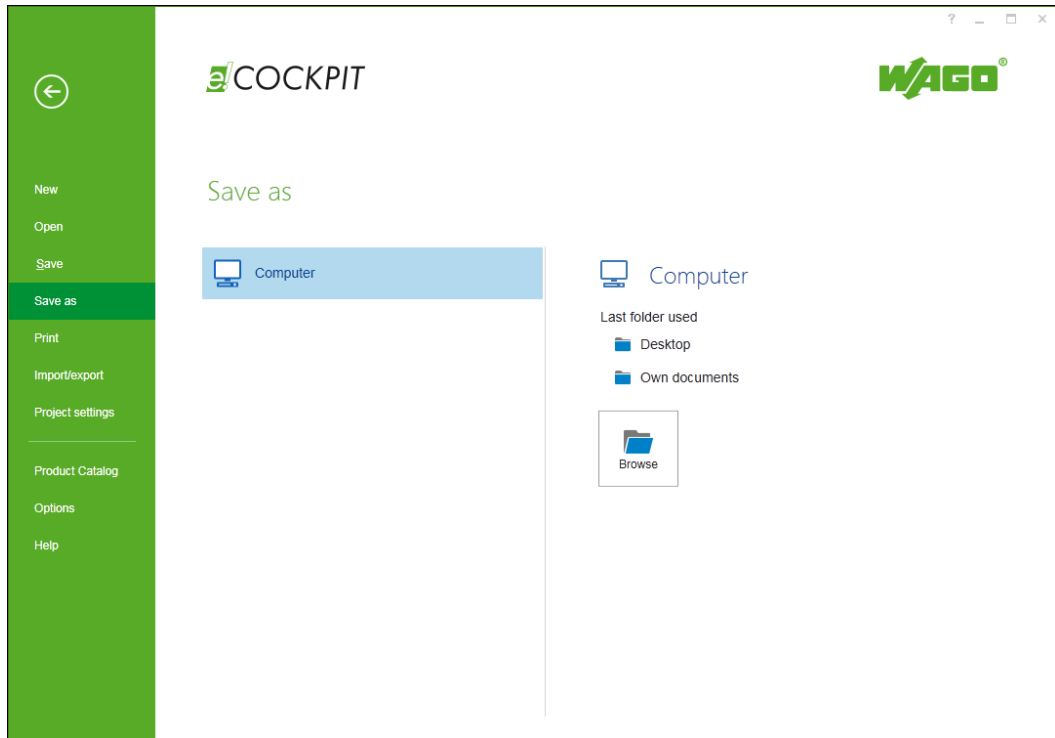


Figure 86: Backstage View, “Save As” Page

8.3.4 “Print” Page

The “Print” page generates a print version of the entire project content or selected objects.

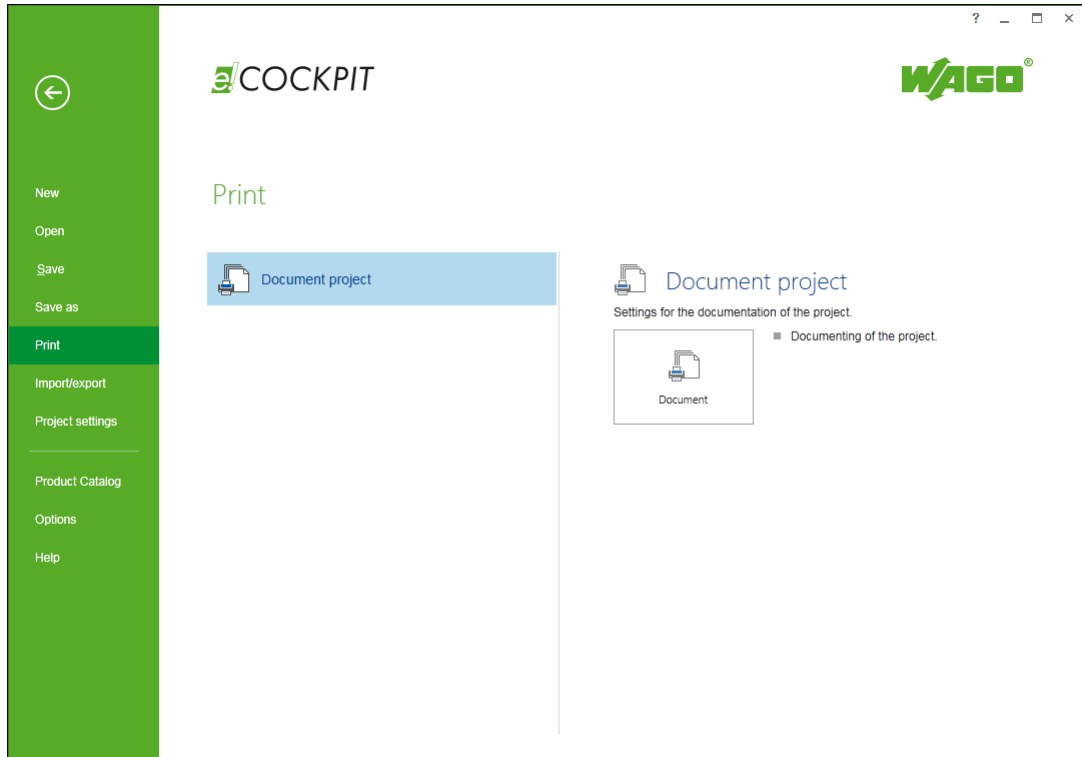


Figure 87: Backstage View, “Print” Page

Table 48: Backstage View Operating Elements, “Print” Page

Button	Description
[Document project]	Opens a dialog for selecting contents of the current project that can be combined and printed

8.3.5 “Import/Export” Page

The “Import/Export” page enables projects from CODESYS 2 and 3 to be imported, as well as *e!COCKPIT* archives generated and opened.

Functions are provided for compiling and saving libraries ([IEC library] button) if a library is selected on the “New” page as a project template.

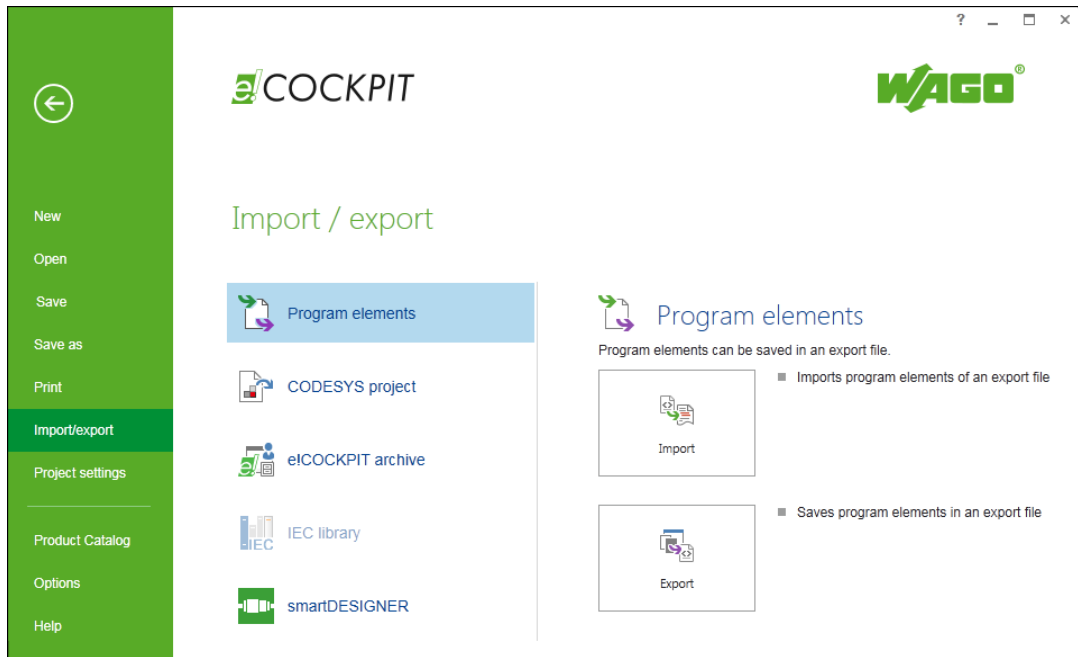


Figure 88: Backstage View, “Import/Export” Page

Table 49: Backstage View Operating Elements, “Import/Export” Page

Button	Description
Program elements	
Import	Imports program elements of an export file
Export	Save program elements in an export
CODESYS project	
[Open CODESYS 2 project]	Imports CODESYS 2 projects (file suffix “.pro”)
[Open CODESYS 3 project]	Imports CODESYS 3 projects (file suffix “.project”)
e!COCKPIT archive	
[Save archive]	Saves the current project on the controller as an archive file. The archive combines project-related files so that they can be forwarded to third parties. An archive combines all the files contained and referenced in the currently opened project into a central “*.projectarchive” archive file.
[Open archive]	Opens a project present on the controller as an archive file
IEC library	
[Compile and save]	Compiles and saves the library with the file suffix “compiled-library”
[Save in library repository]	Saves the library in the system repository
smartDESIGNER	
[Import]	[Import] loads all devices and modules of a project saved as a new project in e!COCKPIT from smartDESIGNER via CAE export.
[Export]	[Export] saves all devices and modules in a file that can be read in from smartDESIGNER via the CAE import and further processed. The [Export] button is inactive if a project is not present or a library project is selected.

8.3.6 “Project Settings” Page

The “Project Settings” page is used to view and modify CODESYS-specific project settings. For additional information, see the CODESYS Help.

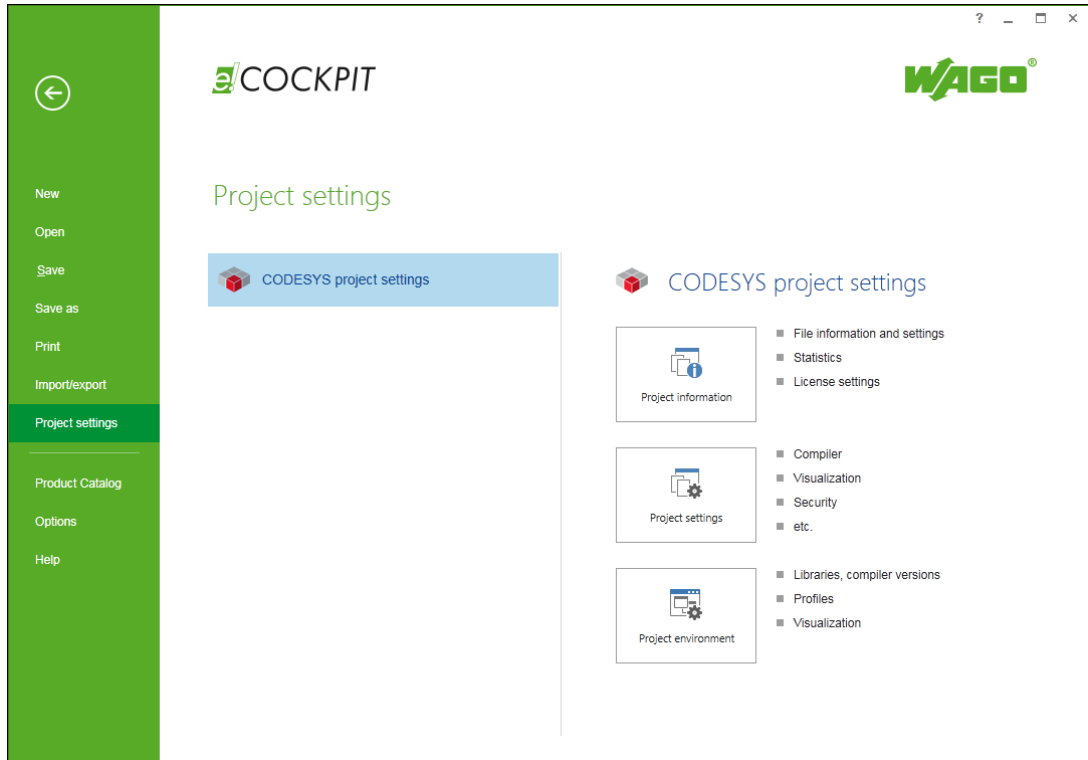


Figure 89: Backstage View, “Project Settings” Page

Table 50: Backstage View Operating Elements, “Project Settings” Page

Button	Description
[Project information]	Opens the “Project Information” dialog to view or enter file information/settings, statistics and license settings (see CODESYS online help: “File, Project information”)
[Project settings]	Opens the “Project Settings” dialog for setting tasks such as compiler options and visualization. Also opens a dialog for security settings including project encryption, user and access rights administration, version handling and page settings for printing. The availability of individual categories partly depends on the installation scope (see CODESYS online help: “File, Project settings”).
[Project environment]	Opens the “Project Settings” dialog for managing versions of embedded libraries, compilers, devices, visualization profiles and styles (see CODESYS online help: “File, Project environment”)

8.3.7 “Product Catalog” Page

The “Product Catalog” page adds and deletes devices, as well as manages installed device description files.

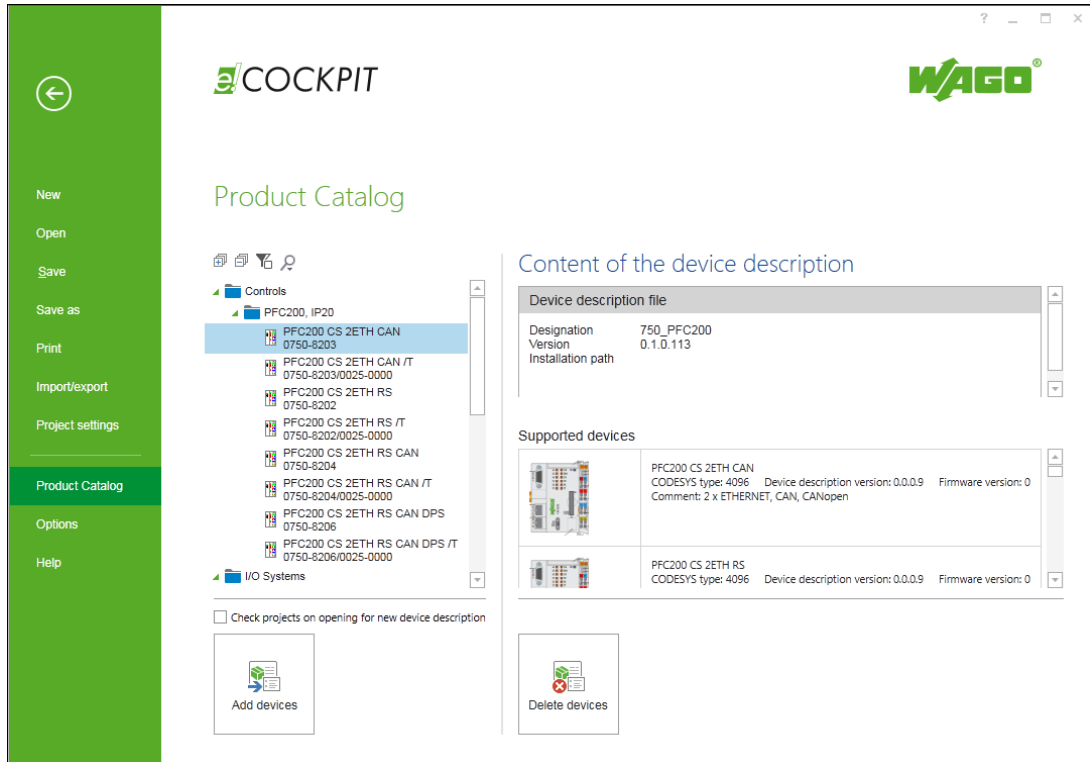


Figure 90: Backstage View, “Product Catalog” Page

The Product Catalog content is shown on the left-hand side of the page (see also Section “Panels” > “Product Catalog”).

When a device is selected, the name, version and path of the installed device description file are shown on the right-hand page (“Content of the device description”).

The lower area lists all devices that belong to this description file and version (“Supported Devices”).

Table 51: Backstage View Operating Elements, “Product Catalog” Page

Symbol/Function	Description
[Add devices]	Imports a device description file. A window opens to select the description files.
[Delete devices]	Deletes the device description file for the device selected in the Product Catalog, as well as for all devices listed under “Supported devices” using the same device description

Table 51: Backstage View Operating Elements, “Product Catalog” Page

Symbol/Function	Description
Check projects on opening for new device descriptions	When a project is opened, this automatically checks for new versions of device description files available for the included devices in the <i>e!COCKPIT</i> Product Catalog If updates are needed, the versions replaced are indicated. The update is performed after confirmation.
	When disabled, there is no check for new device descriptions. The check can be performed manually and individually for each device (“Replace Device Description” function).

8.3.8 “Options” Page

The “Options” page contains general settings such as language selection or resetting of the workspaces, as well as CODESYS-specific settings. For additional information, open CODESYS Help.

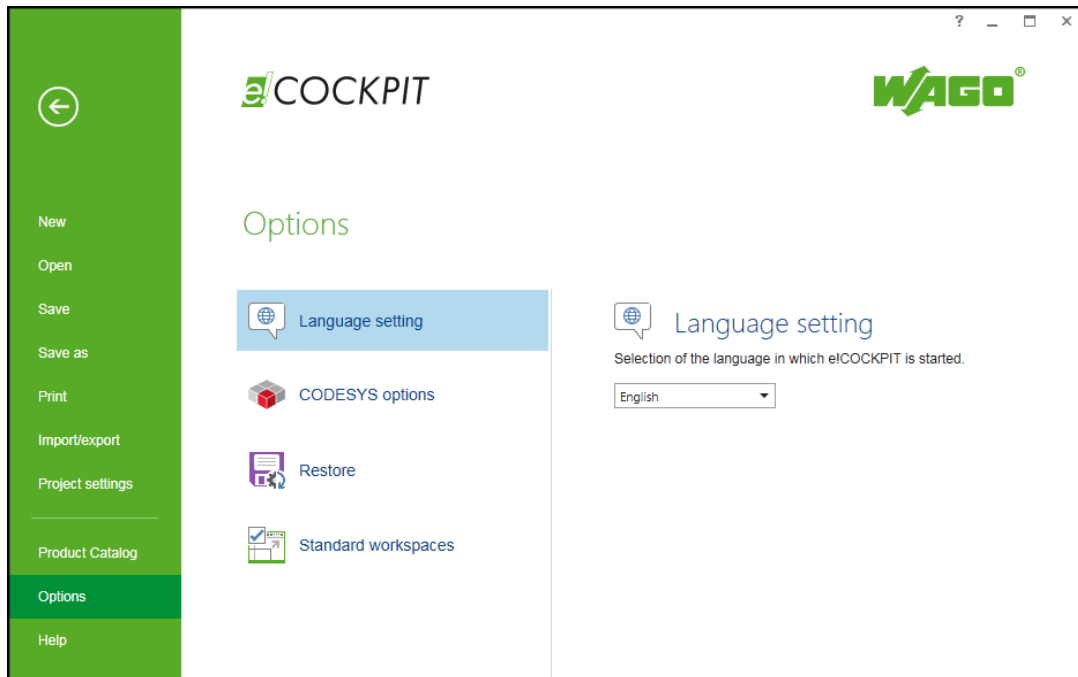


Figure 91: Backstage View, “Options” Page

Table 52: Backstage View Operating Elements, “Options” Page

Button	Description
[Language setting]	Sets the language in which <i>e!COCKPIT</i> is started (German, English) The setting is activated when the software is restarted.
[CODESYS options]	[Options] opens a window with general options that control the appearance of the CODESYS user interfaces embedded in <i>e!COCKPIT</i> (see CODESYS online help: “Options dialog”). [Library/visualization elements/visualization styles repository] opens library repositories, visualization elements and styles that are installed and used in CODESYS projects. The repositories stored there are installed from the specified location permanently in <i>e!COCKPIT</i> . When changing to other libraries, for example, these must not be replaced in the local system folder but must be imported from the repository in <i>e!COCKPIT</i> and installed.
[Restore]	[Restore] saves the current project at regular intervals if changes were carried out. The save interval can be set in minutes (default activated, 10 min).

Table 52: Backstage View Operating Elements, “Options” Page

Button	Description
[Standard workspaces]	[Reset standard workspaces] resets changes to the “Network/Devices” and “Programming” standard workspaces back to the installation settings.

8.3.9 “Help” Page

The “Help” page contains product and license information, product web pages, support addresses and update options.

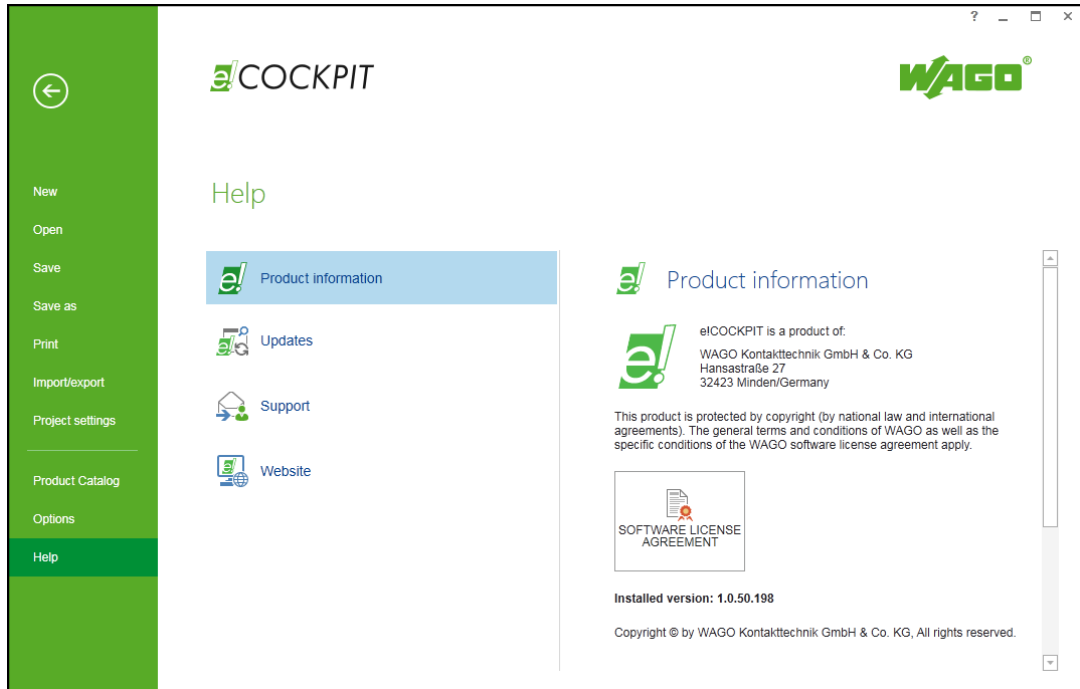


Figure 92: Backstage View, “Help” Page

Table 53: Backstage View Operating Elements, “Help” Page

Button	Description
[Product information]	Displays the <i>e!COCKPIT</i> version. [SOFTWARE LICENSE AGREEMENT] shows <i>e!COCKPIT</i> conditions of use. [Display information] shows the integrated CODESYS version.
[Licensing]	Displays information on the used product license [Enter licenses] opens a dialog for entering access data.
[Updates]	Shows (can be filtered via “Updates”/“New”/“Installed”) all available updates, as well as previously installed versions of the <i>e!COCKPIT</i> software, device description files, communication drivers and IEC libraries [Search for updates] opens a connection with the Internet and searches for available updates. [Install updates] installs selected components. “Check for updates and license information on starting <i>e!COCKPIT</i> ” checks at program start for available updates and displays them. I accept the WAGO SOFTWARE LICENSE AGREEMENT checks license information when installing updates. The license agreement must therefore be accepted to complete the installation.

Table 53: Backstage View Operating Elements, "Help" Page

Button	Description
[Support]	Displays Support contact information Clicking the displayed email address causes a locally installed mail client (if present) to open an email window. Clicking the [Generate Email with Log Files] attaches all <i>e!COCKPIT</i> log files to your email. Click the [Save log files] button to create and save a log file.
[Website]	Displays the link to the <i>e!COCKPIT</i> software website This provides you with general information and additional links.

9 Operating

This section describes software operation through its stages and typical workflows.

9.1 Working with the User Interface

In this section, familiarize yourself first of all with the general settings of the user interface and the interaction between the workspaces and panels.

9.1.1 Using Online Help

The online help feature in *e!COCKPIT* contains the description of the *e!COCKPIT* user interface (1) and help for the integrated CODESYS 3.5.(2) software.

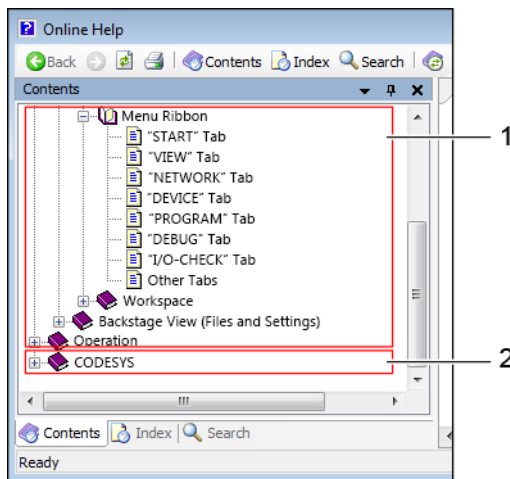


Figure 93: Using Online Help

Please note that some settings, windows, etc. in CODESYS online help may be named differently and that the workflows described may differ slightly from those in *e!COCKPIT*.

9.1.1.1 Calling Up Online Help

The *e!COCKPIT* online help function is context-sensitive – the help pages displayed relate directly to the currently active panel.

- To display all sections of the Online Help, click [?] at the top right of the window.
- To display the online help of a selected/active panel, window or element of the user interface, press the [F1] key.
- Help information on devices are provided in DTP files which are installed during the device installation. You can access the device help files via the Windows Start menu > All Programs > WAGO Software > *e!COCKPIT* > Device Help Files
- Press [F1] if you are in a program editor and require help on elements from libraries.
- In the Library Manager, context-sensitive information on selected elements of WAGO libraries is displayed in the “Documentation” tab. In the Library Manager click a library and an element (e.g., a module) in order to display more information.

Besides online help providing general information on entire windows or dialogs, tooltips are provided for individual buttons.

- To open a tooltip containing a short description of the button function, hover the mouse over a button.

9.1.1.2 Search

Users can perform searches in the online help function to narrow down the results displayed.

1. Click the [Search] button in the online help window.
2. Enter a search term in the text field.
3. Start the search by pressing the [Enter] key or by using the [Search] button.

9.1.2 Using Panels

9.1.2.1 Opening and Closing Panels

The workspace in *e!COCKPIT* consists of different windows; within these windows, the panels can be arranged as required.

1. Open the “VIEW” tab.
2. Click the appropriate button to display the Device Structure, Program Structure, Product Catalog, Messages or Settings panels. The “Settings” panel can be selected as soon as a project contains a device.

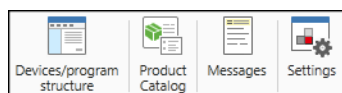


Figure 94: Displaying Panels

The buttons of visible panels are highlighted in blue.

3. To close a panel, click the ‘x’ on the panel itself.



Figure 95: Closing a Panel

9.1.2.2 Docking, Floating and Grouping Panels

To change the layout of the panels:

1. Click the title bar of the appropriate panel.
2. Hold down the mouse button and drag the panel to the required position.

The panel appears as a transparent blue object. Potential destinations are shown for the panel.

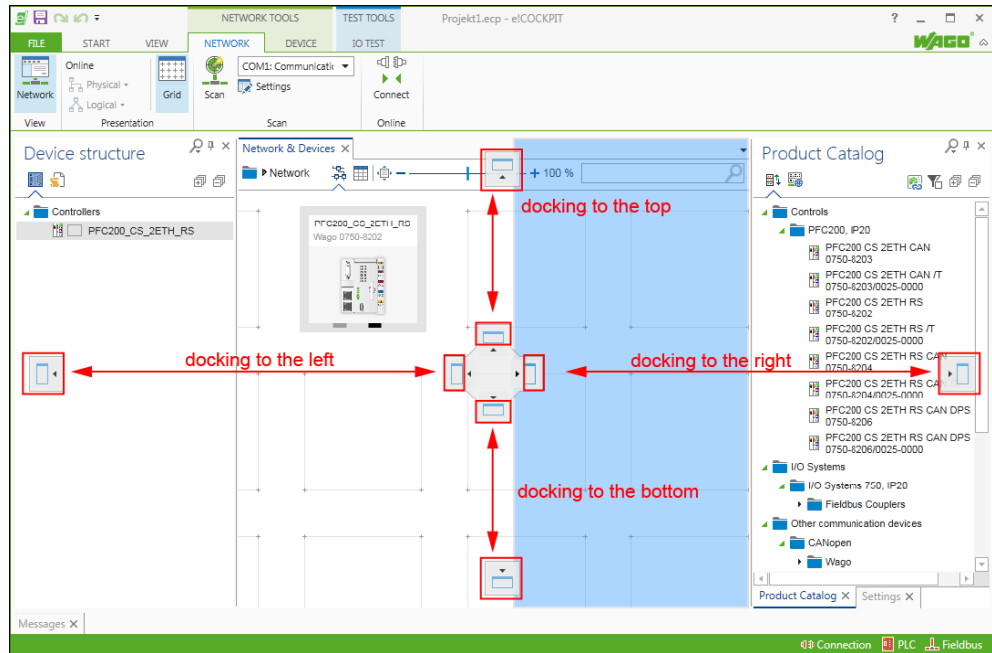


Figure 96: Docking Panels

3. **Docking:**

To dock the panel at this position, release the mouse button precisely ON the appropriate location icon.

Floating:

To float the panel on the workspace, release the mouse button at any other position.

Grouping:

Drag the panel over, underneath or between other panels so that a cross appears. Release the mouse button on the appropriate side of the cross to connect it with the desired panel.

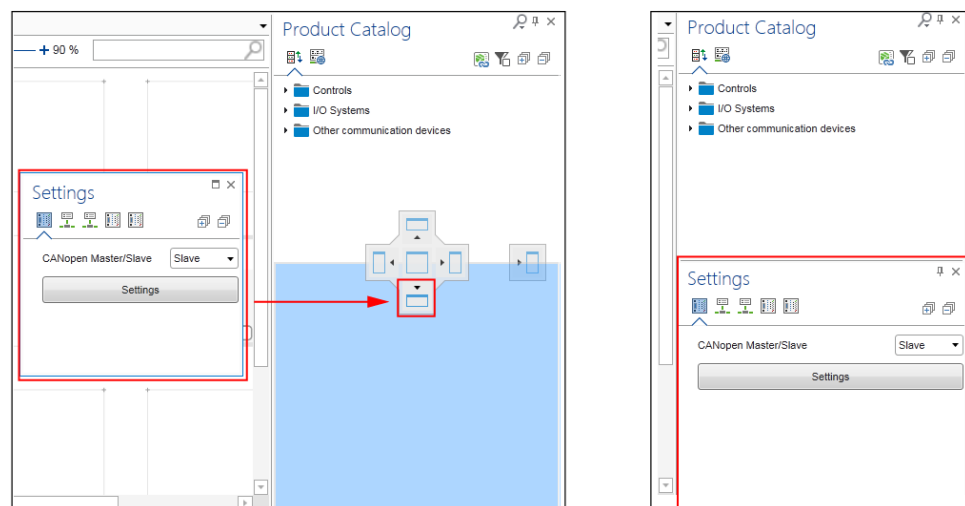


Figure 97: Grouping Panels

To separate panels from each other, drag the particular panel to another area.

9.1.2.3 Docking and Floating the Main View

The tabs of the main view cannot be docked, however they can be floated on the workspace.

1. Click the tab of the main view, hold down the mouse button and drag the view to another position.
2. To centrally dock the main view again, move the view to the docking point at the center of the workspace and release the mouse button.

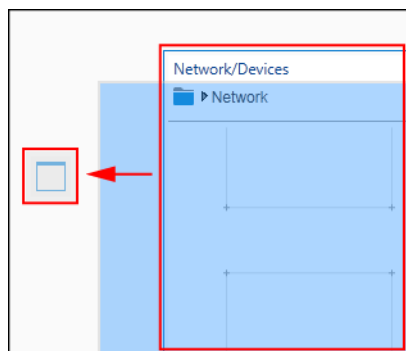


Figure 98: Docking and Floating the Main View

9.1.2.4 Using Multiple Selection

e!COCKPIT supports the multiple selection of elements in order, for example, to move several devices to the Network view or to easily change the settings of multiple devices.

You can always jointly select those elements that belong to the same class. For this reason, I/O modules and head stations cannot be selected together.

Some functions/buttons are deactivated if they cannot be applied to multiple elements simultaneously.

Selecting Multiple Elements with [Ctrl] + Mouse:

1. Click the elements to be selected while holding the Control key [Ctrl] down.

The selection appears in blue.

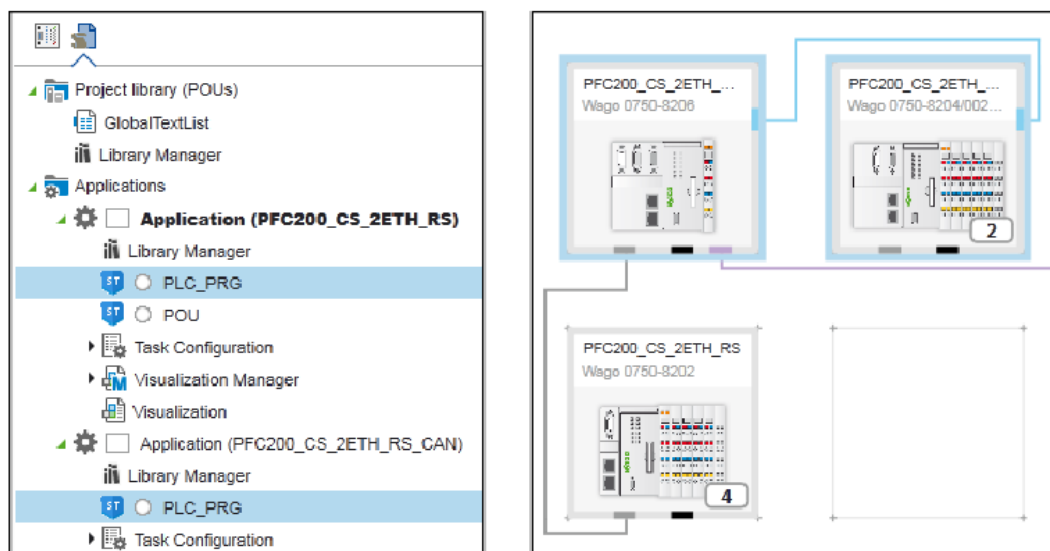


Figure 99: Multiple Selection in Tree Structures or Network View

Selecting Multiple Contiguous Elements with [Shift] + Mouse:

1. To select a contiguous series of elements, select the first element with the mouse and click the last element while holding down the **[Shift]** key.

This selection is also only possible for elements of the same class. Other devices are skipped.

In Network view, click the first and last device to be selected from left to right and from top to bottom with the **[Shift]** key pressed to add the entire group of devices to the selection. This selection will also include devices that are not visible in the current view on the Network view workspace.

Selecting Elements with Frames (Network View):

1. Click the mouse and drag a frame around or across the device to be selected.

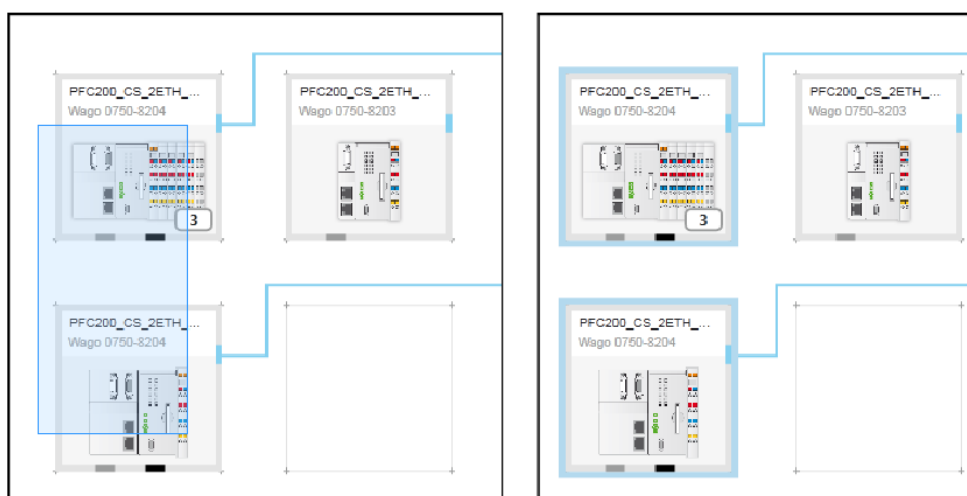


Figure 100: Multiple Selection with Frames

The devices touched by the frame are selected.

2. Move the selected devices to free tiles.

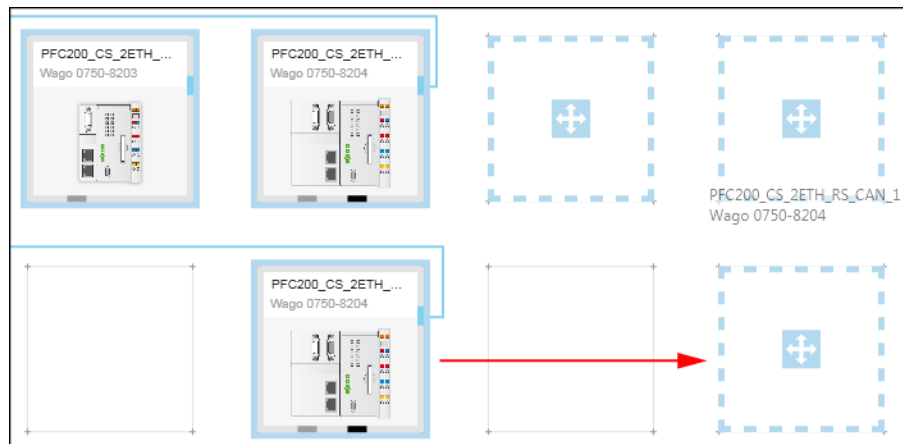


Figure 101: Moving Multiple Selection

Deselecting Individual Elements with the [Ctrl] Key:

1. In order to deselect individual elements from a selection, click them while pressing the [Ctrl] key.

Canceling the Entire Selection:

1. To deactivate the entire selection, press the [Esc] key.

Invert Selection:

1. To invert the selection click the [Invert selection] button in the “START” tab of the menu ribbon.

9.1.3 Using Workspaces

e!COCKPIT's graphical user interface can be configured as needed. Predefined workspaces are provided for two main tasks in a window layout.

The predefined workspaces are provided in the “START” tab or the “VIEW” tab via the menu ribbon. The appropriate window layout is displayed in the workspace after selecting a workspace.

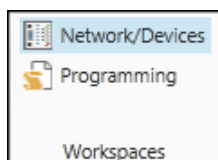


Figure 102: Predefined Workspaces

9.1.3.1 Opening the “Network/Devices” Workspace

1. To open the workspace for configuring the network and setting device parameters, click the [Network/Devices] button in the “START” or “VIEW” tab.

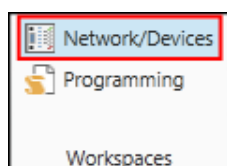


Figure 103: Opening the “Network/Devices” Workspace

Alternatively, click the [Network/Devices] button in the Program Structure.

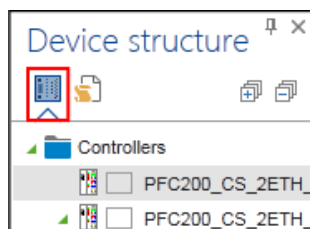


Figure 104: Opening the “Network/Devices” Workspace

The [Network/Devices] button accesses two workspaces: the Network view and the Device Detail view. Network view is displayed first by default. If the Device Detail view is displayed, click **Network** in the Breadcrumb Navigation.

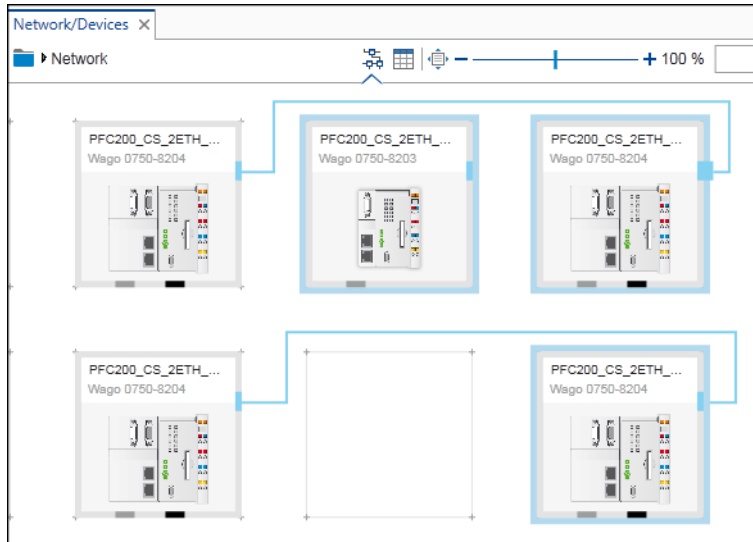


Figure 105: Opening Network View

The “NETWORK” tab provides functions for configuring the network.

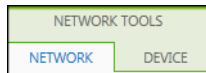


Figure 106: “NETWORK” Tab

2. Move to Device Detail view to display a single selected device. To do this, double-click the device in the Network view.

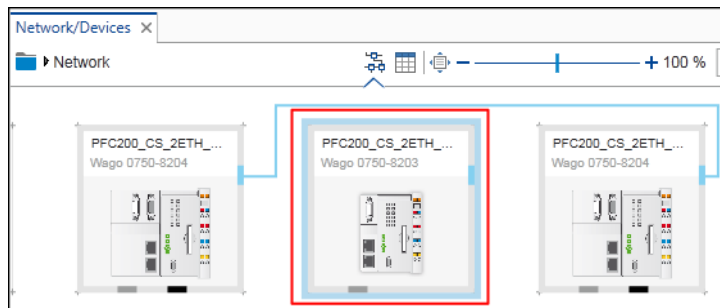


Figure 107: Opening Device Detail View via Network View

Alternatively, click [Device details] in the “DEVICE” tab.

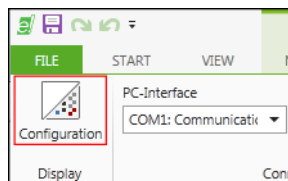


Figure 108: Opening Device Detail View via the “DEVICE” Tab

You can also open the Device Detail view by double-clicking a device in the Device Structure.

The particular device is displayed.

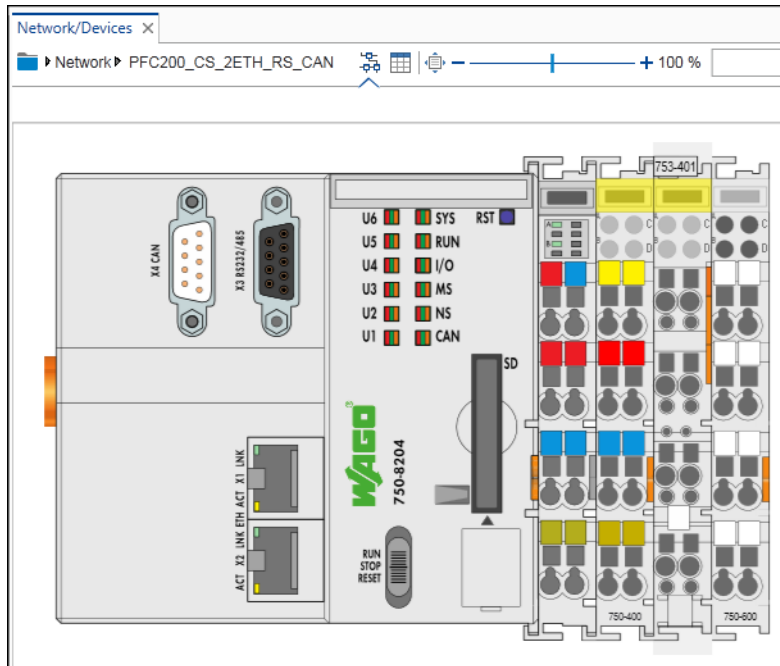


Figure 109: Opening Device Detail View

The “DEVICE” tab in the menu ribbon displays functions for configuring the device.

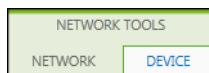


Figure 110: “DEVICE” Tab

3. Click “Network” in the Breadcrumb navigation to return to Network view.

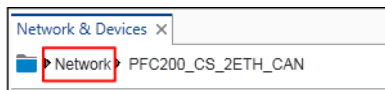


Figure 111: Use the Breadcrumb navigation to move to the Network view

Alternatively, click the [Network] button in the “NETWORK” tab.

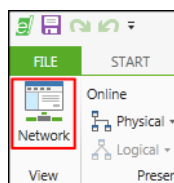


Figure 112: Use the “NETWORK” tab to move to the Network view

Moving between Network view and Device Detail view is possible by double-clicking the entries of the Device Structure:

4. To open Network view, providing an overview of all controllers, double-click “Controllers.”
5. Double-click the required device to open the Device Detail view.

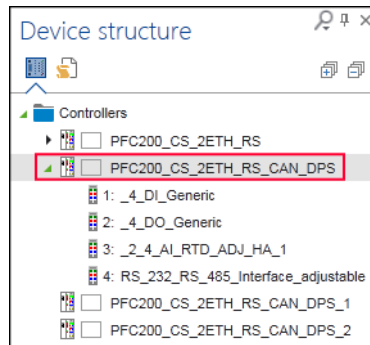


Figure 113: Moving between Network View and Device Detail View

You can choose between a graphical or tabular display both in Network view and in Device Detail view.

6. Click the appropriate button in order to move to the graphical or tabular view.

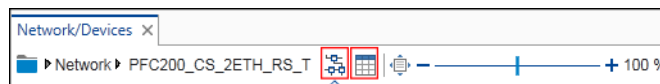


Figure 114: Toggling between Graphical and Tabular View

9.1.3.2 Opening the “Programming” Workspace

1. To open the “Programming” workspace for programming the controller, click the **[Programming]** button in the “START” or “VIEW” tab.

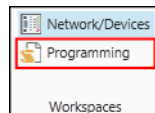


Figure 115: Opening the “Programming” Workspace

Alternatively, click the **[Programming]** button in the Device Structure.

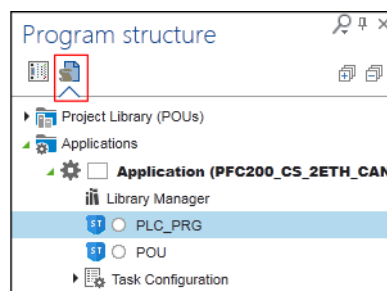


Figure 116: Opening the “Programming” Workspace

The “Programming” workspace is displayed.

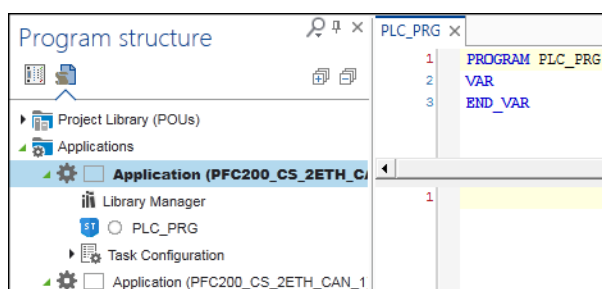


Figure 117: Opening the “Programming” Workspace

This opens the “PROGRAM” and “DEBUG” tabs in the ribbon, which contain additional functions for programming and debugging.

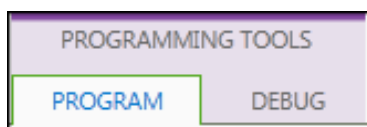


Figure 118: Opening the “PROGRAM” and “DEBUG” Tabs

9.1.3.3 Using User-Defined Workspaces

You can save and manage your own window layouts in the “Workspaces” group in the “START” tab. A saved workspace is stored as a user-defined button in the menu ribbon. You can restore modified default workspaces at any time. The information for user-defined workspaces is stored for each user in the local Windows properties.

9.1.3.3.1 Creating a Workspace

1. Open the “START” tab.
2. Click [**Manage**] in the “Workspaces” group.
3. Click [**Create new workspace**].
4. Assign a name for the workspace.
If required, select any pictogram of your choice (JPG, PNG or BMP) for the button via [**Change image**]. The image is automatically scaled to a size of 32 x 32 pixels.

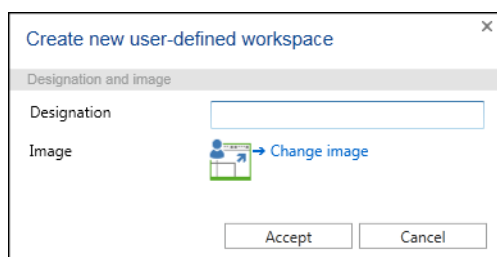


Figure 119: Creating a New Workspace

5. To save the workspace, confirm your setting with [**Accept**].
6. The new button is displayed in the menu ribbon.



Figure 120: New Button in the Menu Ribbon

The active workspace is displayed on a blue background.

9.1.3.3.2 Editing a Workspace

Changing Name and Image:

1. Click [**Manage**] in the “Workspaces” group of the “START” tab.
2. Select the workspace to be edited under “User-Defined Workspaces.”
3. Click [**Edit workspace**].
4. Change the name and/or image of the workspace.
5. Save the changes by clicking [**Accept**].

Changing the Window Layout:

1. To make future modifications to the window layout, create the layout first:
Rearrange the panels or open others.
2. Click the [**Save the current workspace**] button in the “START” tab.

The previously created workspace is overwritten with the new settings.

Please note that changes are only possible within one application area, i.e., the “Network/Devices” workspace cannot be changed to a “Programming” workspace. Changes refer only to panels and their visibility, position and size.

9.1.3.3.3 Deleting a Workspace

1. Click [**Manage**] in the “Workspaces” group of the “START” tab.
2. Select the workspace to be deleted under “User-Defined Workspaces.”
3. Click [**Delete workspace**].

9.1.3.3.4 Restoring a Workspace

The two predefined standard workspaces “Network/Devices,” and “Programming” can be changed with user-defined settings. To restore the default settings:

1. Move to Backstage view (“FILE” tab).
2. Select “Options.”
3. Click [**Standard workspaces**].
4. Click [**Reset standard workspaces**].

9.1.4 Using Key Combinations

Use key combinations to quickly access frequently required functions.

- To enter key combinations, press the keys stated in the following list. Keys to be pressed simultaneously are marked with a plus sign (+).

9.1.4.1 General

Table 54: General Key Combinations

Function	Key Combination
Save	[Ctrl] + [S]
Select All	[Ctrl] + [A]
Print	[Ctrl] + [P]
Remove Multiple Selections	[Esc]
Open Input Assistant	[F2]

9.1.4.2 “START” Tab

Table 55: Key Combinations, “START” Tab

Function	Key Combination
Copy	[Ctrl] + [C]
Cut	[Ctrl] + [X]
Paste	[Ctrl] + [V]
Edit	[Ctrl] + [E]
Delete	[Del]

9.1.4.3 “PROGRAM” Tab

Table 56: Key Combinations, “PROGRAM” Tab

Function	Key Combination
Search	[Ctrl] + [F]
Replace	[Ctrl] + [H]
Next	[F3]
Next (selection)	[Ctrl] + [F3]
Previous Search	[Shift] + [F3]
Previous Search (selection)	[Ctrl] + Shift + [F3]
Build	[F11]
Connect	[Alt] + [F8]
Disconnect	[Ctrl] + [F8]

9.1.4.4 “DEBUG” Tab

Table 57: Key Combinations, “PROGRAM” Tab

Function	Key Combination
Start	[F5]
Stop	[Shift] + [F8]
Single cycle	[Ctrl] + [F5]
Toggle	[F9]
Step Over	[F10]
Single Step	[F8]
Step Out	[Shift] + [F10]
Write Values	[Ctrl] + [F7]
Force Values	[F7]
Unforce Values	[Alt] + [F7]

9.2 Using Projects

A project contains configured networks, as well as their entered devices and programming.

9.2.1 Creating a New Project

1. Launch *e!COCKPIT*.
2. Select a template in the Start view, e.g., “Empty Project.”

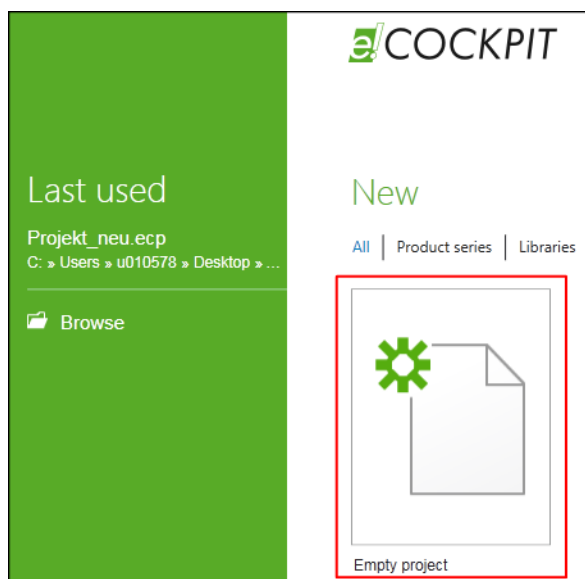


Figure 121: Selecting a Template

This creates a project. The main view is opened.

Information



Additional Information

The “Empty Project” template can be used, for example, for online configuration in mutli-controller projects.

The “750 Product Series” template can be used, for example, for offline configuration in single-controller projects.

9.2.2 Opening an Existing Project

1. Launch *e!COCKPIT*.

Recently edited projects are displayed in the Start view under “Last used.”

2. To open a project, click one of the displayed project entries.

Alternatively click **[Browse]** to open projects of other sources (**[Open]** button) via the Start view.

Information



Dock the position of the project file!

To keep frequently used projects in the “Last used” list and place them at the top of the list, click the gray pin symbol behind project name. The pin is shown in green.

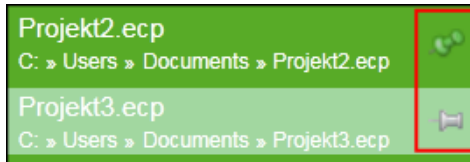


Figure 122: Docking the Position of the Project File

Click the pin again to float the project once more from the position. The pin is then shown in gray.

9.2.2.1 Opening the Project with an Obsolete Device Description

When opening a project containing devices for which new device descriptions are available, this is shown in a dialog window. New descriptions may require a firmware update for your devices.

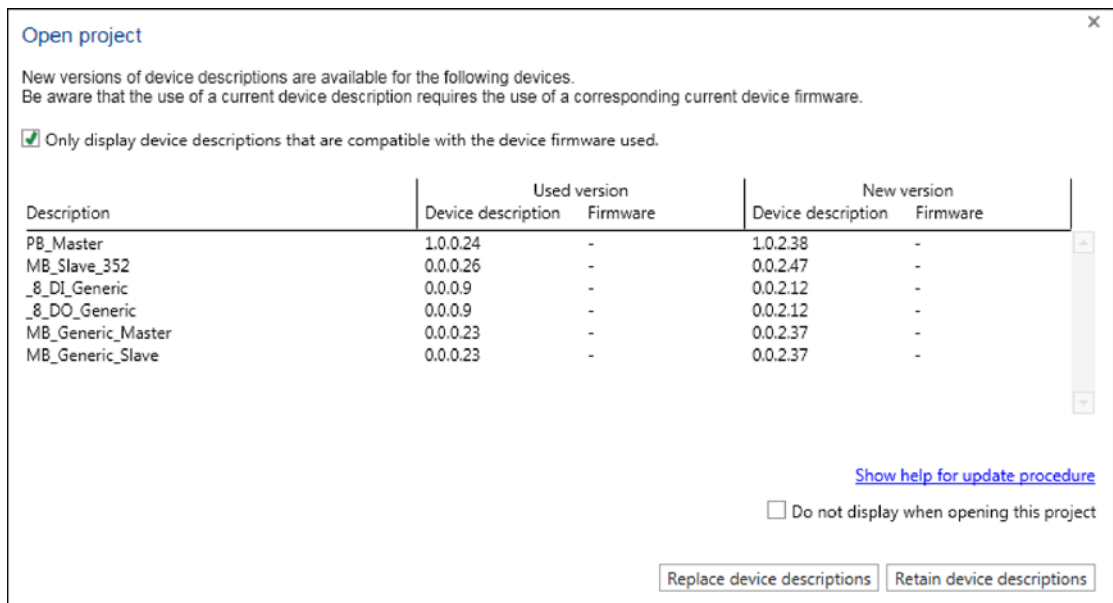


Figure 123: Replacing Device Descriptions

- To display devices descriptions compatible with the current firmware, place a tick next to the checkbox “Only display device descriptions field that are compatible with the device firmware used”.
Remove the tick to display all device descriptions. In this case the device description files that are not compatible with the current device firmware are also displayed.

I/O modules and devices that are not programmable are always displayed in the list and always updated irrespective of the set tick in this dialog.



Note

New device descriptions may require a firmware update!

New device descriptions may require an update of the firmware of your device. Update your firmware to continue using the device in your project.

- Save the setting whether you wish to display only suitable device descriptions for the firmware of your device or all available device descriptions via “Do not display when opening this project.”
If you remove the tick, this dialog is then no longer displayed if new device descriptions exist.
If the tick is set, this dialog window continues to be displayed if new versions of device descriptions are available.
- To update the displayed device descriptions click the [**Replace device descriptions**] button.
To close the dialog window without updating the device descriptions, click the [**Retain device descriptions**] button.
- If you do not wish to update one device description but the device descriptions of all devices used in the project, click the [**Replace all**] button.

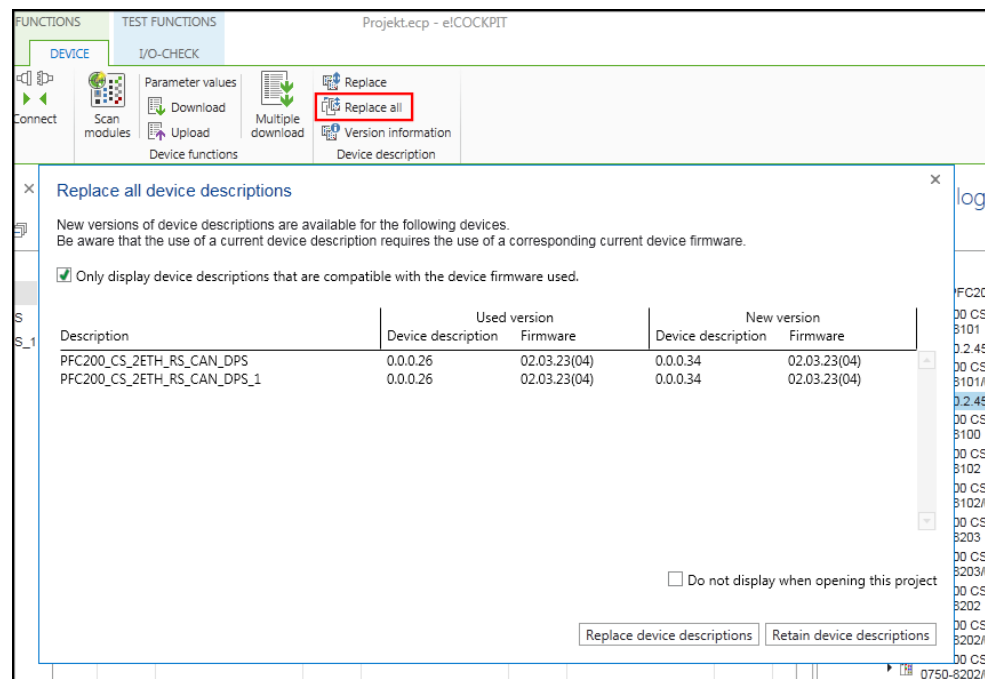


Figure 124: Replacing All Device Descriptions

- Replace the device descriptions as described in the Section “Opening project with obsolete device description”.

9.2.3 Saving a Project

1. Create your project.
2. Open the Backstage view by clicking the “FILE” tab.
3. To save the project, click [**Save**] or [**Save as**].
4. Select a location. Click [**Desktop**], [**Own Documents**] or choose another memory location via [**Browse**].
5. Enter the file name and click [**Save**].

9.2.3.1 Saving Projects Automatically

Projects are saved automatically every 10 minutes if changes are carried out. Proceed as follows to activate/deactivate this function or change the save interval:

1. Open Backstage view.
2. Click [**Options**] in the navigation bar.
3. Click [**Restore**].
4. To activate the function tick “Restore – Save information”
To deactivate the function, untick the option.
5. To change the save interval, enter the corresponding number of minutes in the input field.

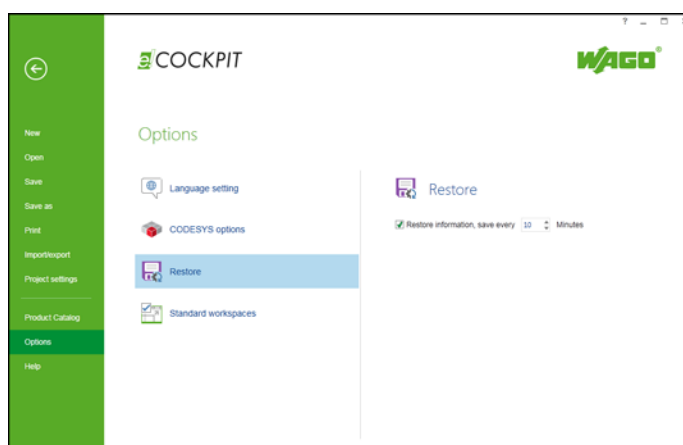


Figure 125: Saving a Project Automatically

9.2.3.2 Restoring Projects after a Software/Hardware Crash

A dialog opens when the software is restarted after a software/hardware crash. You are notified of any restored project files.

1. Click **[Yes]** to display the project files.

The “Open” page of the Backstage view is opened. This page lists all projects that can be restored (in descending order by modification date).

If several instances of *e!COCKPIT* were opened before the crash, the projects of all instances are offered for restoring.

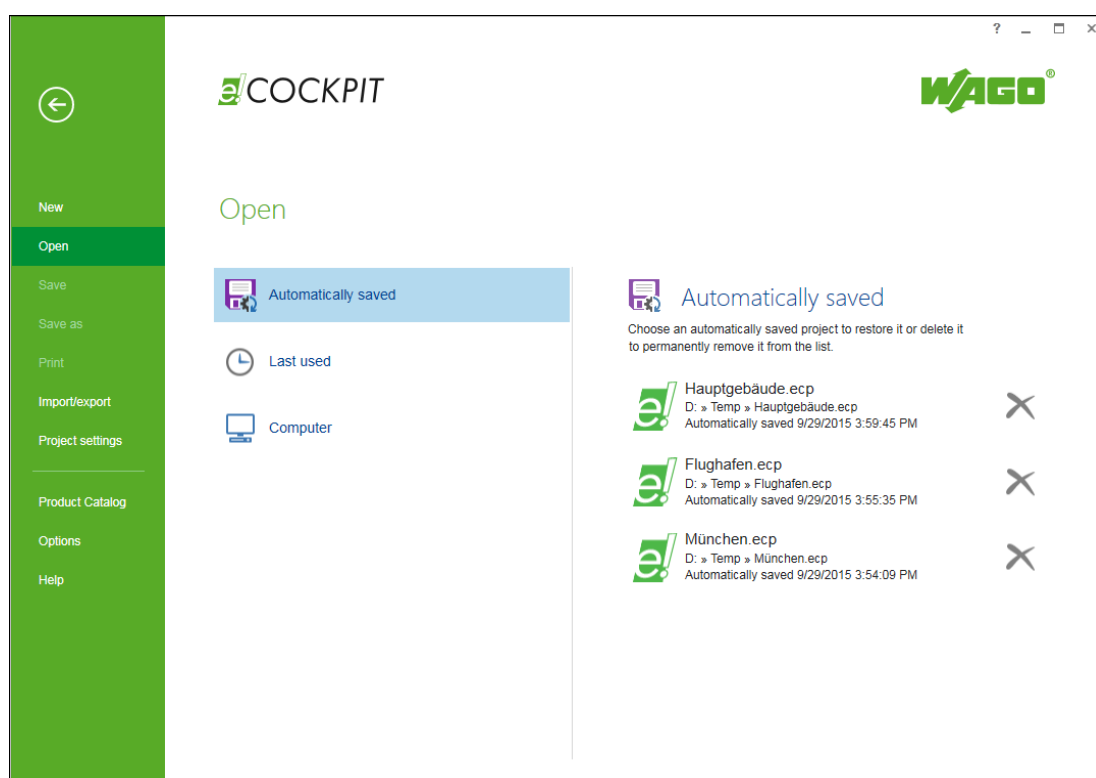


Figure 126: Automatically Saved Projects on the “Open” Page

2. Delete any restored projects that are not required by clicking the cross next to the corresponding file.
3. Double-click one of the displayed files to open it.

Another dialog will open. You are asked whether you wish to open the restored file or the original document. The restoration of the original file discards the restoration file.

4. Select accordingly **[Move to restoration view]** or **[Open original document]**.
5. To save the project click the “FILE” tab and then **[Save]**.

9.3 Configuration and Parameterization

9.3.1 Offline Configuration

In offline configuration mode, you configure the devices and network first without connecting to real devices. The configuration is saved and transferred at a later time.

9.3.1.1 Adding Devices in the Project

If there are imported devices in the Product Catalog (see Section “Setting and Managing” >...> “Adding Devices”), they can be used as follows:

1. To add devices to a project, drag the individual devices (in this example, a PFC200) from the Product Catalog and drop them onto a free tile in Network view.

To place devices on a free tile automatically, double-click the device name or click the green “+” symbol after the device name.

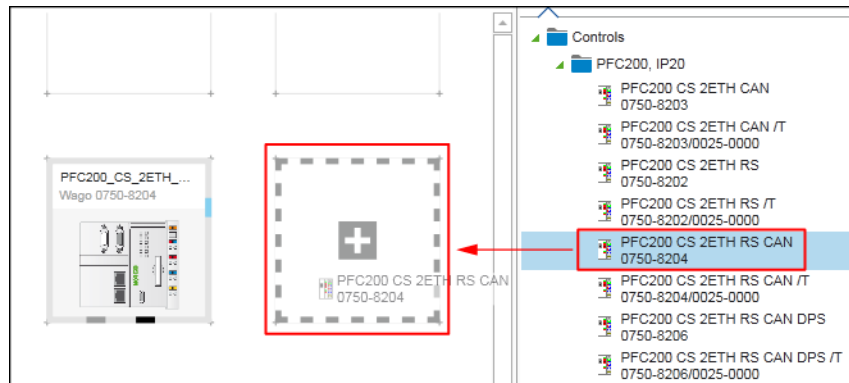


Figure 127: Dragging Devices to the Network View

2. Open the Device Detail view by double-clicking the device.
3. Drag the modules from the product catalog to any position after the head station.

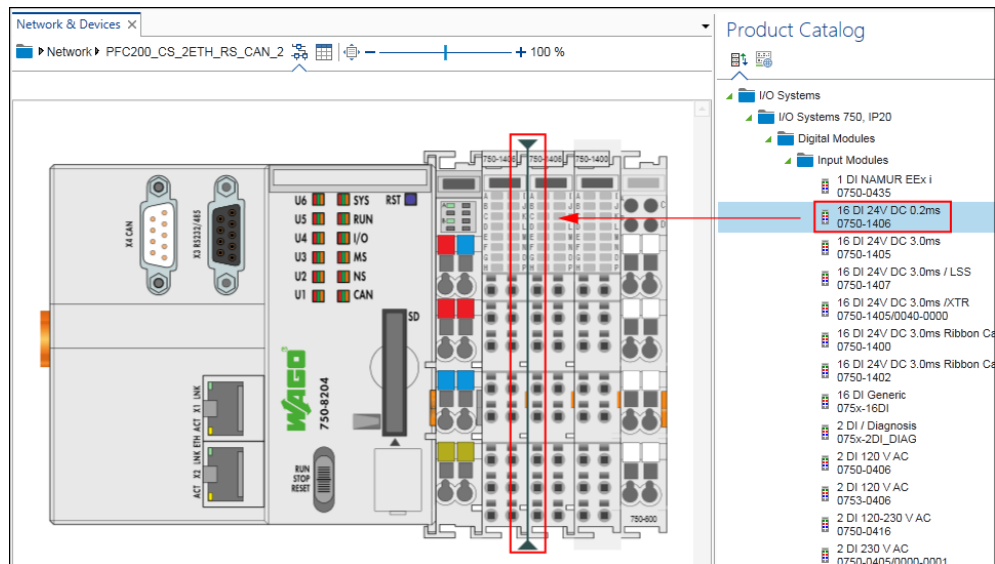


Figure 128: Adding Modules

This will display a [+] next to the mouse pointer. Positioning lines between existing modules indicate at what point the module will be placed when you release the mouse button.

Alternatively, the module can also be added by double-clicking it from the Product Catalog behind the currently selected module or at the last location.

The Device detail view only allows the configuration of one head station with connected modules.

9.3.2 Online Configuration

In online configuration mode, you are physically connected to your devices in the network via *e!COCKPIT*. The configuration can be transferred directly.

Click the [**Connect**] button to connect to available devices directly or perform a network scan first.

9.3.2.1 Scanning the Network and Devices

To display devices present in the network, scan the network first:

1. Click [**Network/Devices**] in the device/program structure.
2. Open the “NETWORK” tab.
3. If the Network view does not appear in the workspace, click the [**Network**] button in the menu ribbon.
4. First check the communication settings by selecting from the selection field in the “Scan Settings” group the medium in which the search will be performed, such as ETHERNET.

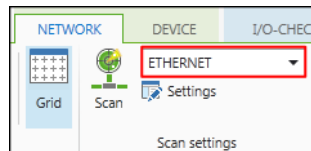


Figure 129: Selecting the Medium for Scanning

5. Click **[Settings]** to make other settings.
6. For instance, restrict the scan range for the ETHERNET settings.

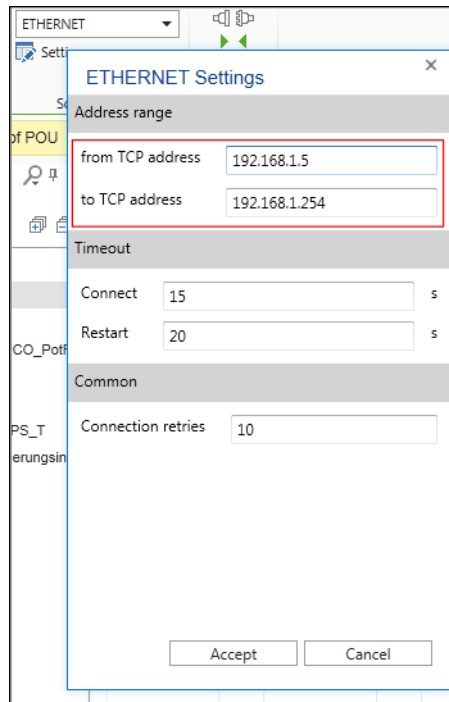


Figure 130: Restricting the Scan Range

7. Click **[Accept]**.
8. To search for devices within the specified range, click the **[Scan]** button in the menu ribbon.

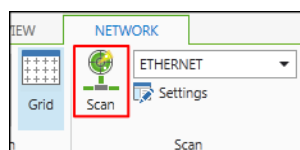


Figure 131: Starting the Scan Operation

After the scan has finished, the search results of the configuration are displayed:

- Green: Configured and scanned device match.
- Red: Configured and scanned device are different.
- Blue: New device detected (if the device cannot be selected, this is an unknown device).

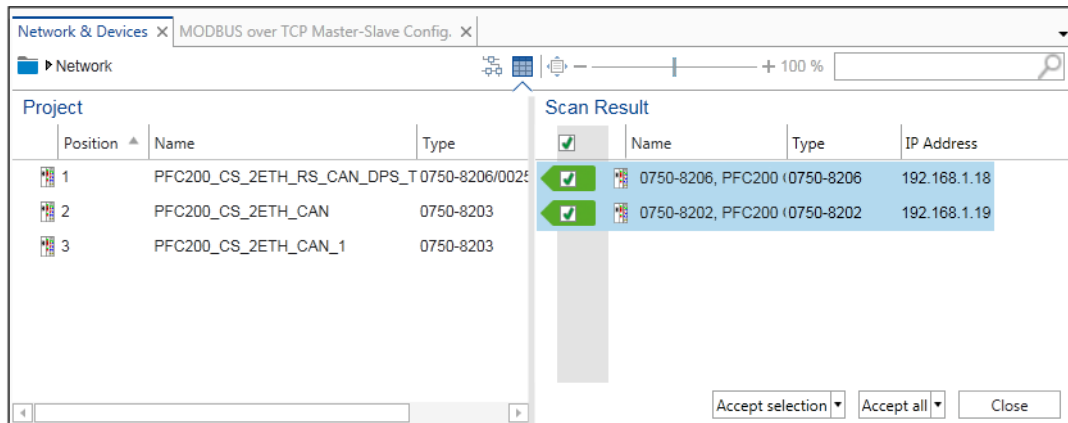


Figure 132: Displaying a Scan Result

A red exclamation mark on a device indicates that it is not in the Product Catalog. In this case import the corresponding device description file for this device by importing it (“FILE” tab > [Product Catalog] > [Add Devices]).

- Use the mouse to select the devices you wish to transfer to your project and click [Accept selection].
Click [Accept all] to transfer all devices.

Note



Scan head stations and modules separately or together!

In this example first select only the head stations to be accepted. Scan modules separately (point 13). Here you can scan the head stations including the connected modules by clicking [Accept selection (incl. Modules)].

It is recommended that you make any name changes in this overview or in the tabular network view as IP address, device type, etc. are displayed here clearly in list form.

- With the mouse, select a device in the “Configuration” list. Click the device again and change the device name.
- To switch from the scan results back to the graphical view (Network view), click the symbol for the graphical Network view.

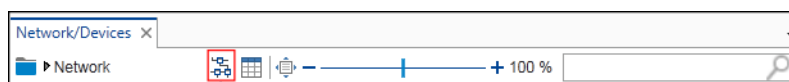


Figure 133: Opening Graphical Network View

- To identify the modules of the scanned head station, double-click the device to enter Device Detail view.
- Click [Scan modules] in the open “DEVICE” tab.

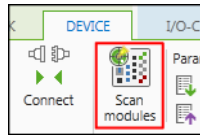


Figure 134: Scanning Modules

If the device is not connected, it is first automatically connected. The modules are then detected. The workspace shows the configuration and scan results. In the right-hand area, all modules which are connected with the device and which have their device description files loaded in *e!COCKPIT* are displayed.

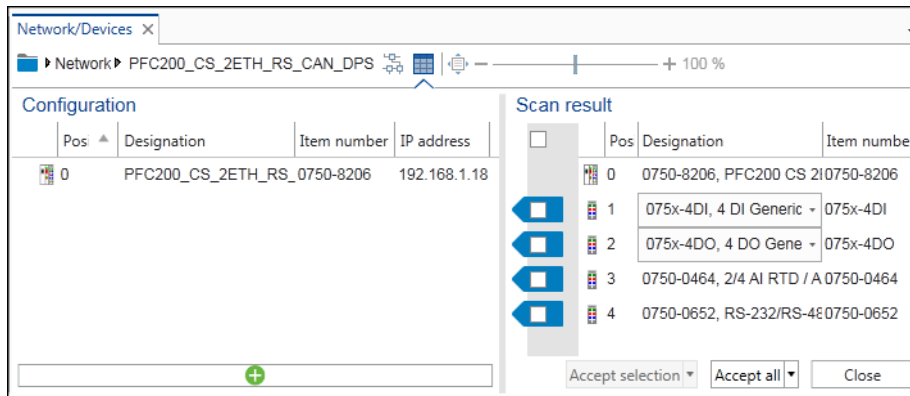


Figure 135: Displaying Scanned Modules

A device description file can represent several modules if these have the same process data structure. A detected module may also be represented by several device description files. For this reason, there can be selection boxes for certain modules.

14. Select the used module in the selection box.

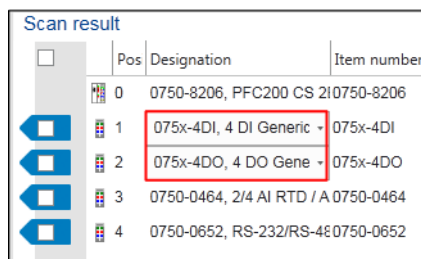


Figure 136: Selecting Modules

If there are several modules with the same scan result, the “Use for all suitable modules” checkbox is displayed in the selection field. Activate this checkbox to accept the selection for all suitable modules at the same time. The corresponding modules are marked for the transfer.

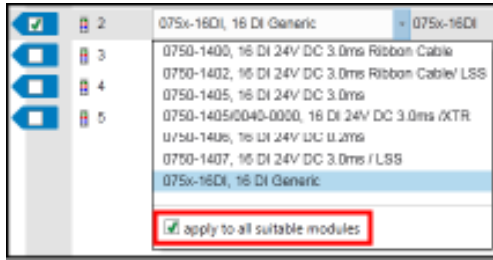


Figure 137: Accepting Modules with the Same Scan Result Together

15. Tick the checkboxes in front of the modules that you wish to add to your project and click **[Accept selection]**.
To transfer all modules, click **[Accept all]**.

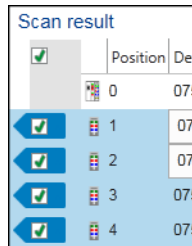


Figure 138: Applying Modules

The Device Detail view appears. You see the head station with the scanned modules.

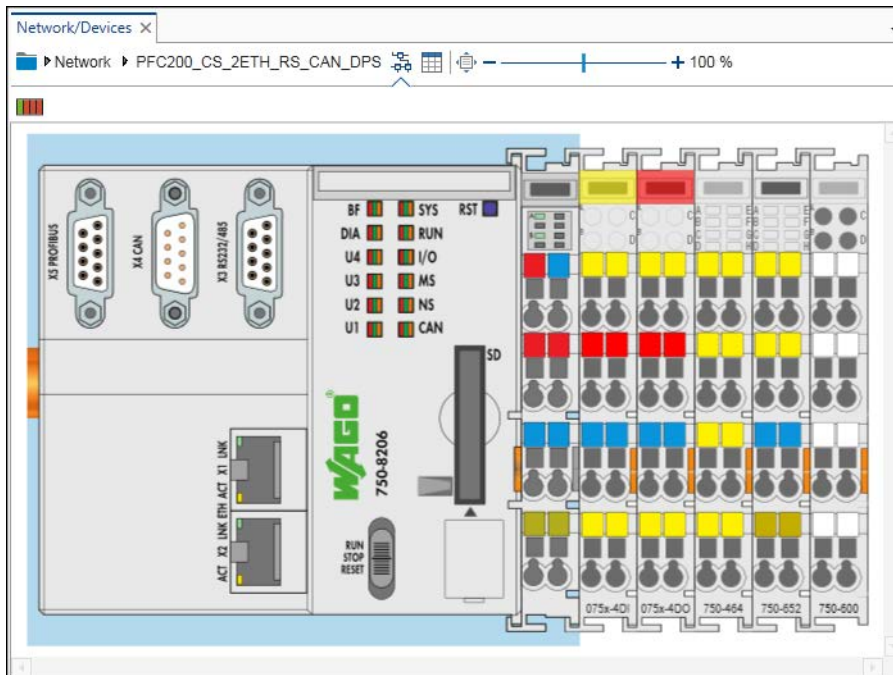


Figure 139: Displaying Scanned Modules

Information



Additional Information

The number of modules detected is displayed in the Network view at the bottom of the device image.

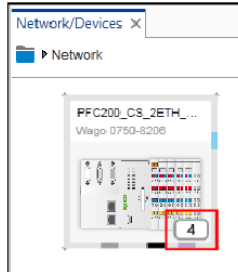


Figure 140: Display of Applied Modules

9.3.2.2 Scanning the Modules of Several Devices

If you have scanned several devices (head stations), you can scan arranged modules for these devices together.

1. Open the Network view.
2. Use the mouse to select several devices.
3. Right-click one of these devices and click **[Scan]** button in the context menu.

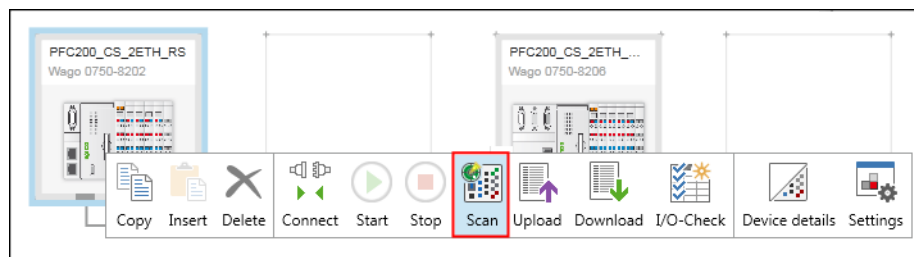


Figure 141: Scanning for Modules of Several Devices

e!COCKPIT scans the modules of the devices and shows these in the “Scan result” dialog.

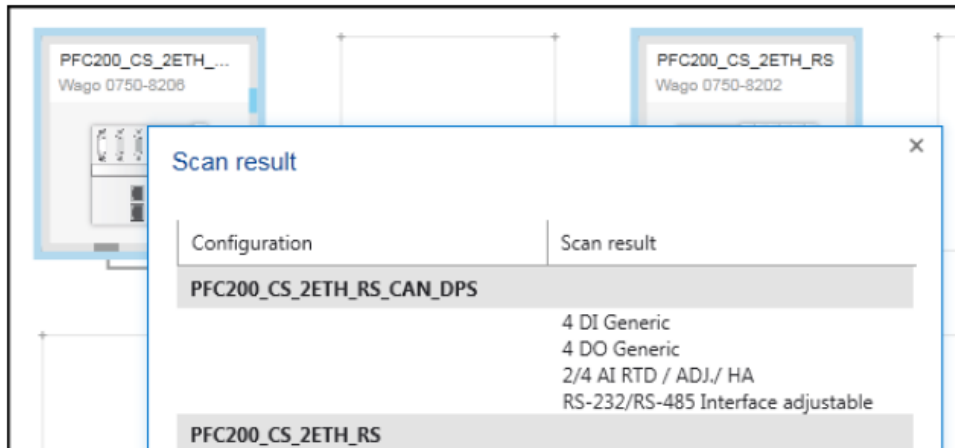


Figure 142: Displaying the Scan Report

4. Click [**Accept**] to add the displayed modules to the project.

9.3.2.3 Connecting and Disconnecting Devices

There are several ways to connect and disconnect devices. This function can also always be executed simultaneously for multiple devices (multiple selection).

- 1 Right-click the device to which you wish establish a connection and click [**Connect**] in the context menu.

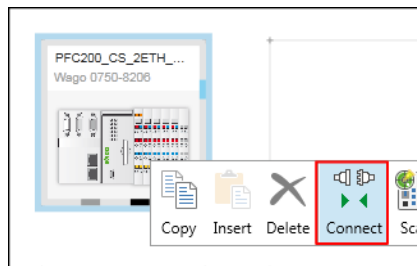


Figure 143: Connecting Devices (Network View)

Alternatively perform the connection via the context menu of the device in the Device Structure.

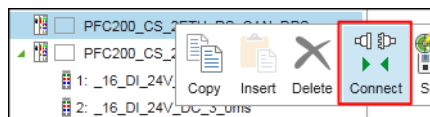


Figure 144: Connecting Devices (Device Structure)

The “Connect” function is also integrated in the menu ribbon. You can find the corresponding button in the “NETWORK”, “DEVICE”, “I/O CHECK”, “PROGRAM” and “DEBUG” tabs.

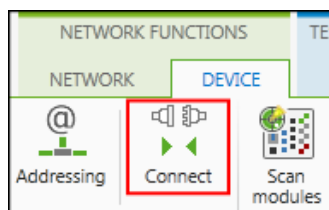


Figure 145: Connecting Devices (Menu Ribbon)



Information

Additional Information

The [**Connect**] button is active if at least one of the selected devices has all the necessary information for opening a connection (e.g., an interface via which the connection can be made).

5. Click the [**Connect**] button.

The connection to the device or the devices is established. An automatic login is carried out in *e!RUNTIME*. The [**Connect**] button changes to a [**Disconnect**] button.

If two or more devices have different states (connected/restricted connection/not connected), these devices are disconnected on the first click of the button and are all connected with the second click.

If parameters were changed, this is displayed in a dialog.

6. If the displayed parameters are to be downloaded to the device, confirm this with [**OK**].
7. If the connection of the selected devices is to be removed, click the [**Disconnect**] button.

The status of the button changes back to “Connect.”

9.3.3 Parameterizing Devices

1. In the workspace of the Network/Device Detail view or in the device structure right-click the device to be parameterized.
2. Click the [**Settings**] button in the context menu.

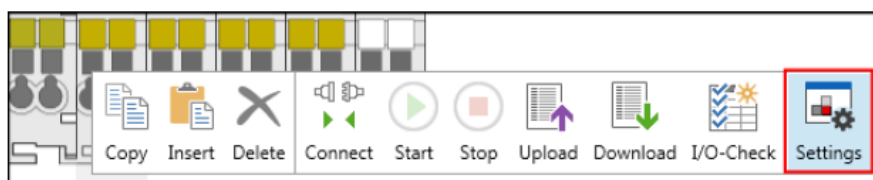


Figure 146: Opening the “Settings” Panel (here: Device Detail View)

The [**Settings**] button is also provided in the menu ribbon in the “VIEW” tab.

The “Settings” panel is displayed (see following Section).

9.3.3.1 Parameterizing Devices via the “Settings” Panel

Tabs, entry screens and selection fields in the panel depend on the particular device. Displayed setting options depend on the content of the particular device description file.

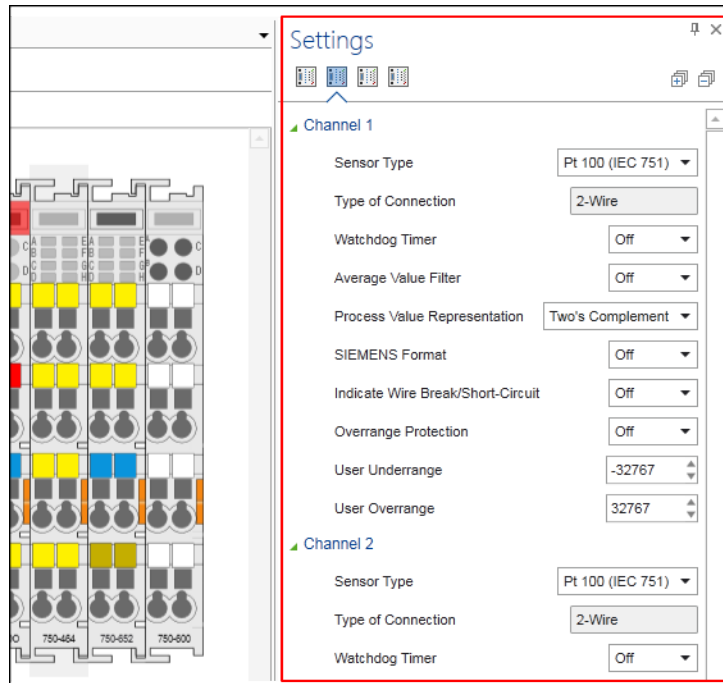


Figure 40: Opening the “Settings” Panel

1. Move between different tabs which contain settings groups by clicking the appropriate tab.

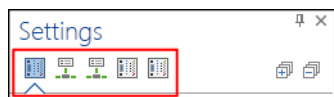


Figure 147: “Settings” Panel (Tab Navigation)

2. Tabs can contain subgroups. Open these by clicking the black arrow symbol.

The group is extended and marked with a green arrow symbol.

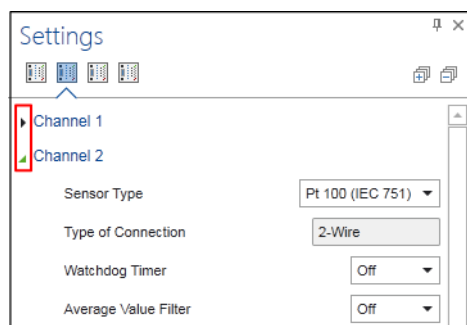


Figure 148: “Settings” Panel (Groups in Tabs)

3. Close the subgroups by clicking on the green arrow symbol.

4. Change the parameters and settings via the specific selection and entry options.

Information



Additional Information

The possible parameters, values and effects of parameter changes are described in the manual of the particular device.

9.3.3.2 Using Device-Specific Configurators

Simple modules can be parameterized via the “Settings” panel. Devices with more complex setting options, such as I/O modules that have mailbox functions, are assigned parameters in their own configurators. The configurators are opened from this panel via the relevant “Settings” buttons.

1. Click the device to be configured.
2. Click the appropriate button in the “Settings” panel to assign additional parameters.

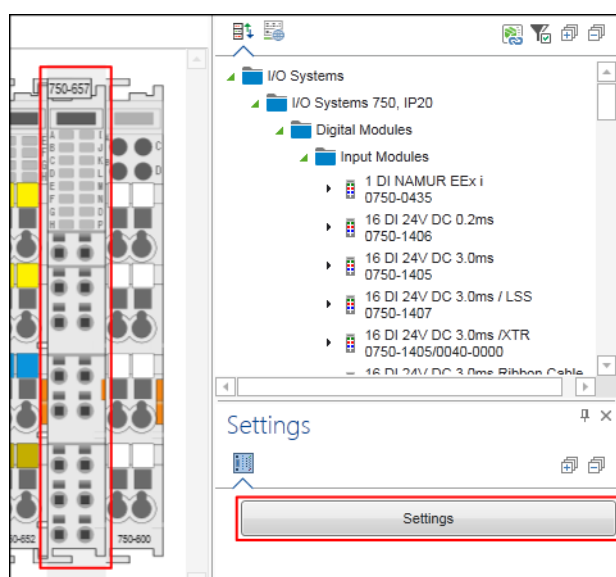


Figure 149: Opening Device-Specific Configurators

Information



Additional Information

The possible parameters, values and effects of parameter changes are described in the manual of the particular device.

9.3.3.3 Uploading and Downloading Parameters

1. Right-click the device with the set parameters and click the **[Download]** button in the context menu.



Figure 150: Writing Parameters to the Device

The parameters are transferred to the device.

2. To read parameters from the selected device with the set parameters, click the **[Upload]** button in the context menu of the device.

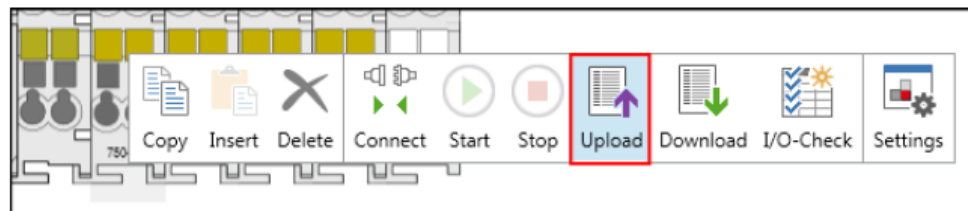


Figure 151: Reading Parameters from the Device

The parameters are loaded from the device to the software and displayed in the “Settings” panel.

The **[Upload]** and **[Download]** buttons are accessed via the context menu of a device in the Device Structure or via the corresponding buttons in the “DEVICE” tab of the menu ribbon.

9.3.4 Configuring Networks

Besides the parameterization of individual devices you can also design complete networks via *e!COCKPIT*. Choose devices from the Product Catalog, assign roles (master/slave), drag connections between the connectors of the devices with the mouse and select communication protocols and communication parameters.

9.3.4.1 Creating a Simple MODBUS Network

1. Select two devices from the Product Catalog and drag them with the mouse into the Network view.

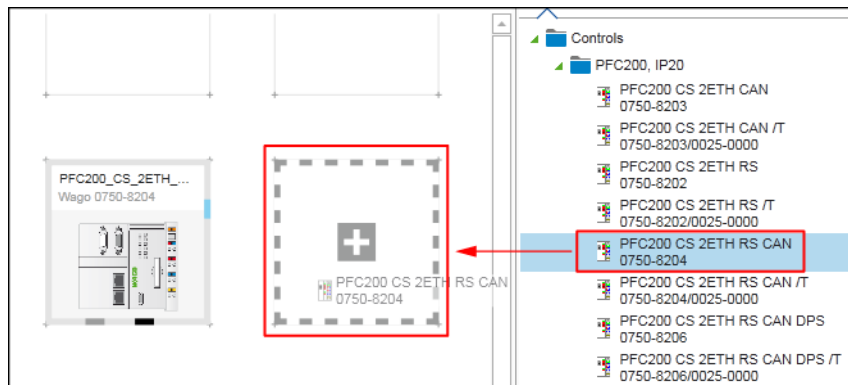


Figure 152: Dragging Devices to the Network View

2. To change the settings or parameters of the devices added, click the respective device, open the “Settings” panel in the context menu and make your changes there.

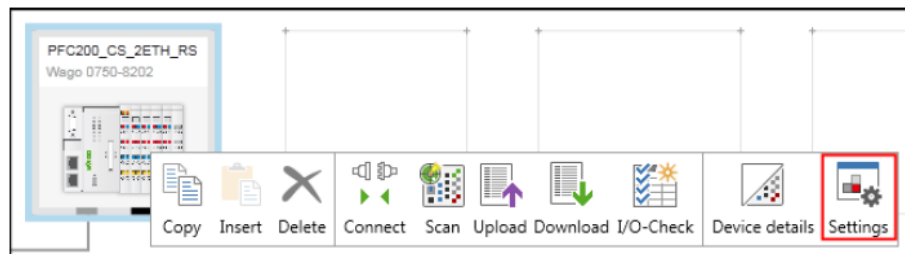


Figure 153: Opening the “Settings” Panel

3. Now connect the devices. Click the gray connector (ETHERNET/MODBUS) for one of the devices in the Network view and hold down the mouse button.
4. To connect the devices to each other, drag the connection line to the same type connector of the second device. Release the mouse button as soon as a green plus sign appears.

The line initially symbolizes the physical ETHERNET connection.



Figure 154: Establishing Network Connections

5. Click a device connector to open the interface.

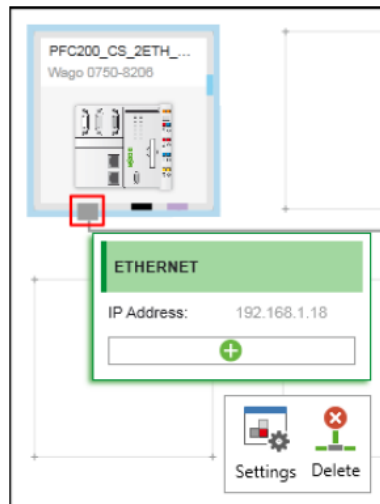


Figure 155: Opening the Interface

6. Click the green plus sign.

Possible connections are displayed for each device. The selection depends on the device used.

7. Select the required connection. In this example, select “MODBUS via (TCP) Master”. You have thus defined the role and protocol used for this connection of the first device (role: client/master, protocol: MODBUS TCP).

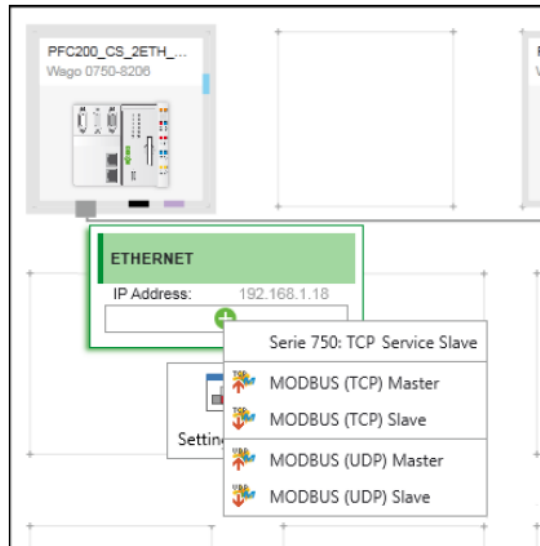


Figure 156: Selecting a Connection

8. A broken line is displayed. Similar to the first line, drag this to the connector of the partner device to establish a logical connection for the existing physical connection.

You have also defined the role and protocol of the second device for this logical connection (role: slave/server, protocol: MODBUS TCP).

The connection to the partner device(s) is displayed.

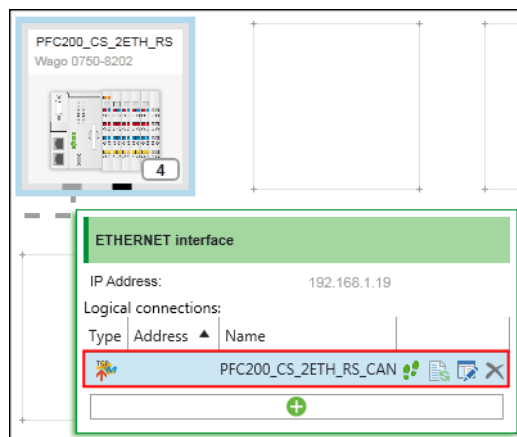


Figure 157: Displaying Devices That Are Logically Connected

9. Move the mouse over the logical connection in the list, so that other buttons are visible.
10. Click the button to open the connection settings.

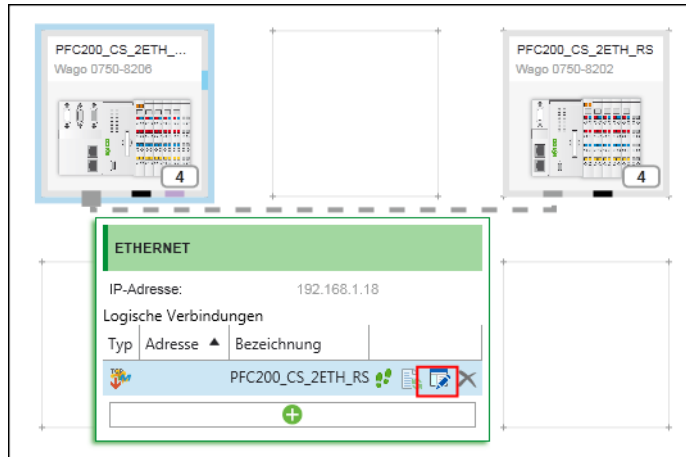


Figure 158: Opening Connection Settings or MODBUS Slave Configurator

The MODBUS Configurator opens matching the role of the device in this connection. In this example, the connection configurator for the MODBUS slave/server is opened (open the configurator for the MODBUS master/client in the same way as for the first configured device).

Use the MODBUS Configurator to define which variables of the slave (server) are available on the master (client).

The figure below shows the MODBUS Configurator for the slave. No variables for communication between master and slave have been defined yet.

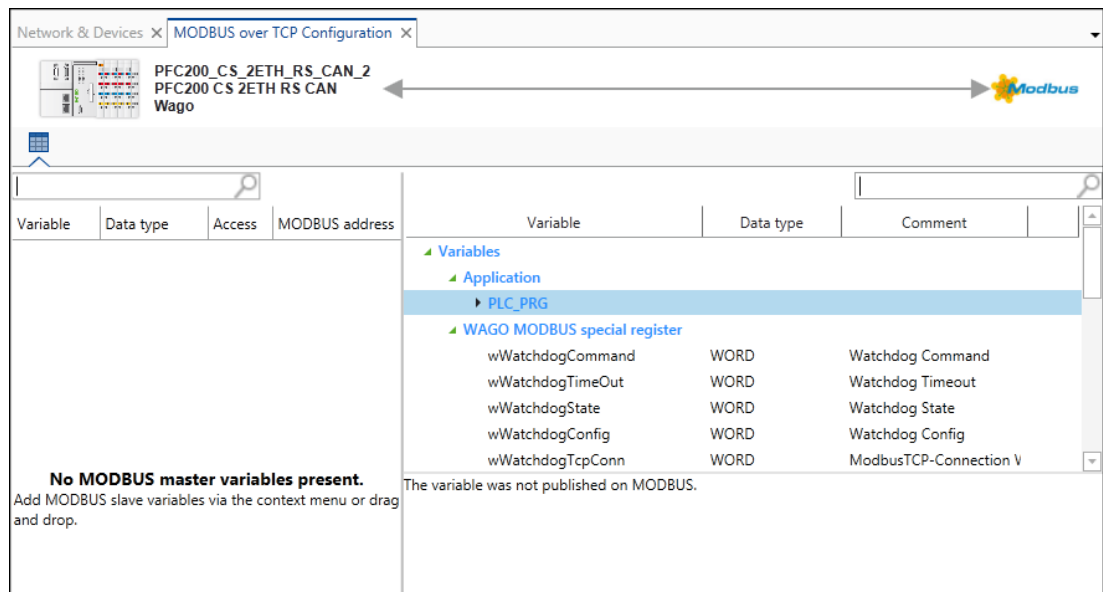


Figure 159: Variable Configuration MODBUS, View for the Master (Client)

The configurator is divided into two columns:

The variables on the master are displayed on the left. The variables available on the slave and used in the slave application are displayed on the right. The tree displays them in a hierarchical view according to the program structure.

Create variables:

1. Right-click “PLC_PRG.”

The context menu opens.

2. Click the button to add a data point to create a variable in your program “PLC_PRG.”

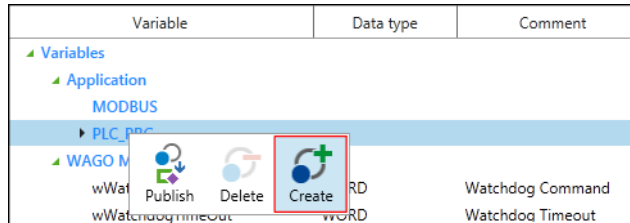


Figure 160: Creating New Variables on the Slave

Click the new variable to display the settings for MODBUS access to the new variable “newVAR” in the bottom half of the Slave view (“Variable Settings”).

3. Enter settings in this area if you want configure access to this variable via MODBUS (Description, Data Type, Array, Access, MODBUS Address).

The variable is created in the Slave application.

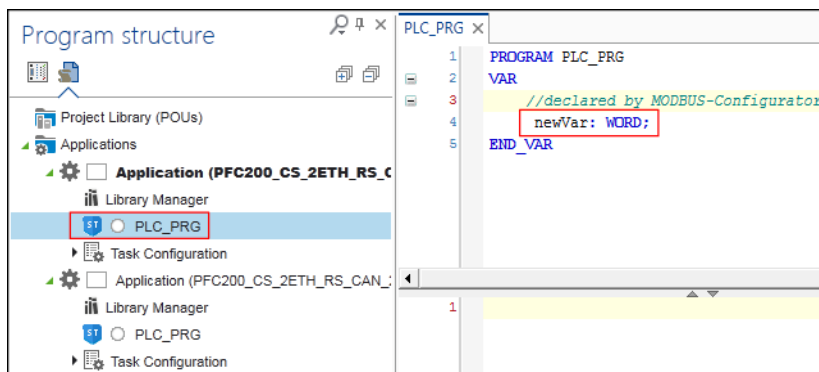


Figure 161: New Variable in the Slave Application

A variable is created in the master/client application by default (corresponding to the path (namespace) in the slave application). The variable is available on the master directly.

4. Click the master connector in the Network view.
5. Open the master configurator by clicking the button to open the connection settings.

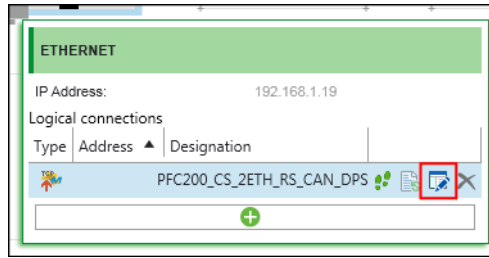


Figure 162: MODBUS Connection Settings

The MODBUS Configurator for the master opens. The variable created on the master (1), the variable on the slave (2), the cycle time for updating the variables via MODBUS and MODBUS access to the variables (4) are displayed.

ReadOnly (RO): Master reads

WriteOnly (WO): Master writes

ReadWrite (RW): Master reads and writes

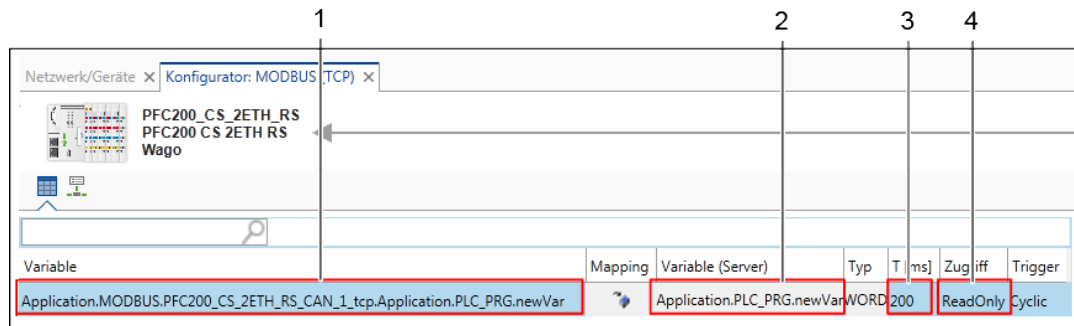


Figure 163: New Variable on the Master

Open the “Programming” workspace and open the main program of the master.

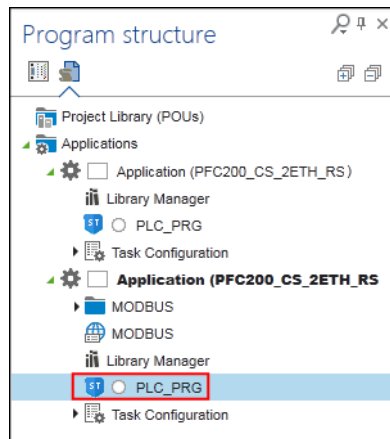


Figure 164: Opening the Main Program of the Master

In the master application, access to the “newVar” variable as shown in the following figure. Tip: Press the [F2] key to open the Input Assistant for easy variable selection.

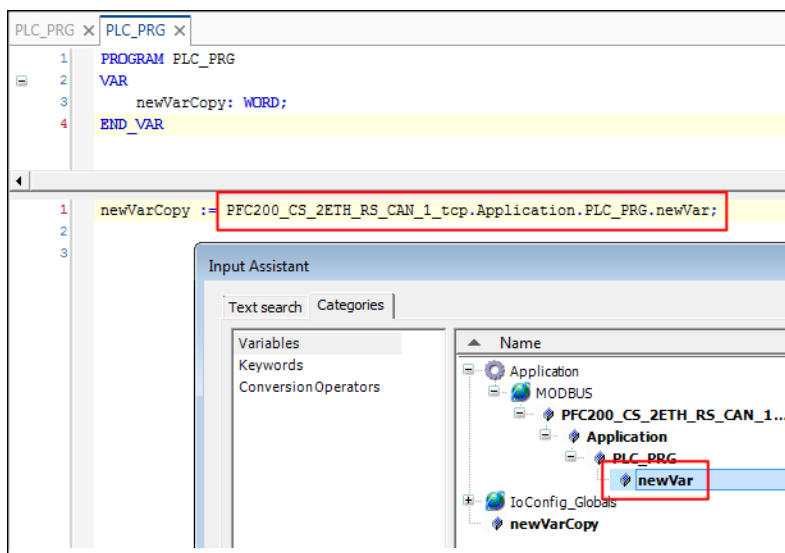


Figure 165: Accessing the “newVar” Variable in the Master Application

In this way, a variable (“newVar”) has been declared in the slave application, made accessible via MODBUS and read in the master application.

9.3.4.2 Creating a Simple CANopen Network

In the following description you connect two CANopen devices and create data points for joint communication.

1. Select two devices from the Product Catalog (e.g., 750-8204) and drag them with the mouse into the Network view.

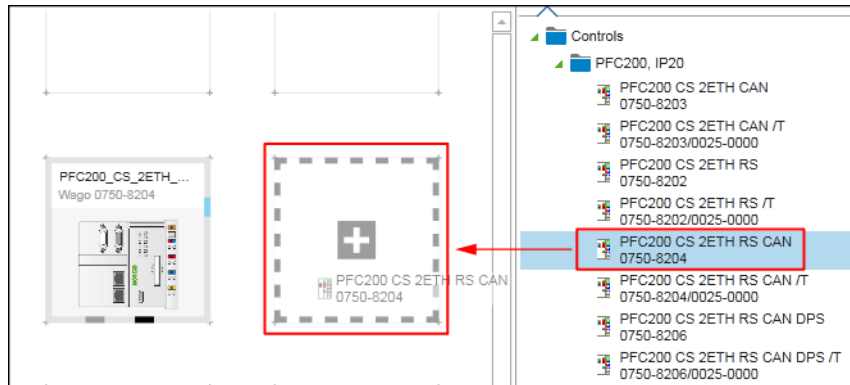


Figure 166: Dragging Devices to the Network View

2. Select the first device and click the “DEVICE” tab in the menu ribbon.
3. Click the [Address Management] button.
4. Enter the IP address of the device in the entry field and click [Ok].

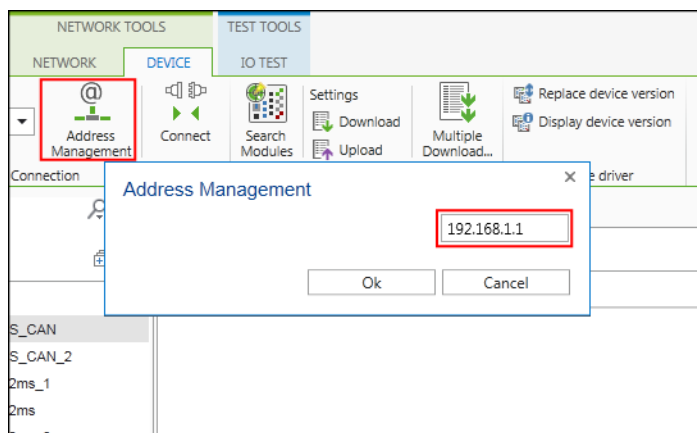


Figure 167: Assigning an IP Address

5. Proceed in the same way with the second device. The IP addresses of both devices must be different.
6. Open the “Settings” panel by right-clicking the first device and clicking the [Settings] button in the context menu.

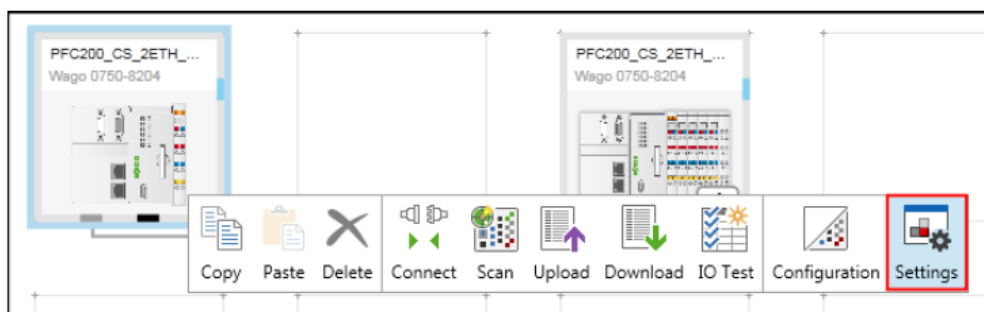


Figure 168: Opening the “Settings” Panel

7. Assign the master role to the first device by selecting “Master” in the drop-down menu.

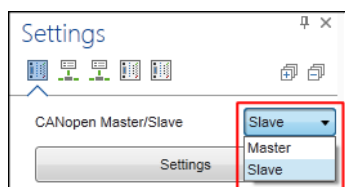


Figure 169: Assigning the Role of the Device

8. Configure the second device (if not preset) in the same way, however, as a slave.
9. Create a logical connection between the devices by dragging a connection from the CANopen connector of the master to the CANopen connector of the slave.

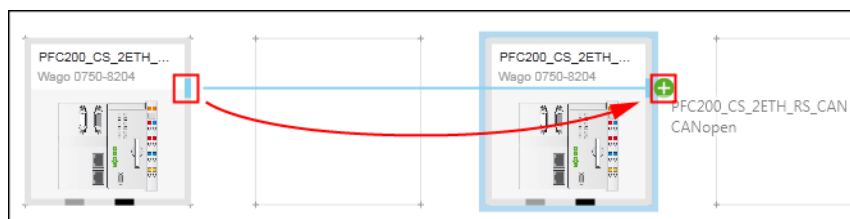


Figure 170: Creating a Connection between CANopen Devices

Preparing the Slave:

10. Select the CANopen slave with the mouse.
11. Change if required the “Node ID” (e.g., “1”) and the baud rate (e.g., 250000) in the “Settings” panel.
Note that “Node IDs” within a CANopen network must always be different.

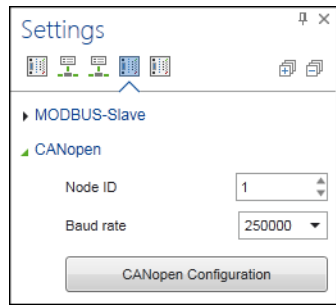


Figure 171: Adapting “Node ID” and Baud Rate

Create a data point for the slave (UINT):

12. To display the logical connection to the master click the CANopen connector of the slave and then the button for opening the connection settings.

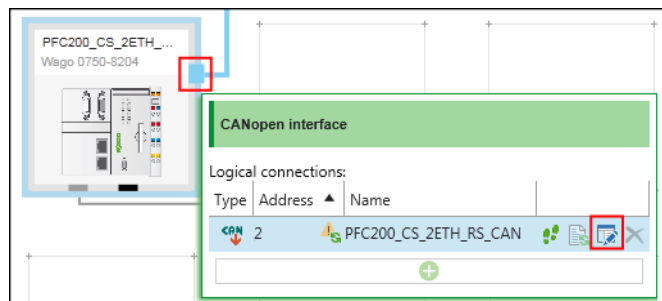


Figure 172: Opening the CANopen Connection Configurator

The CANopen connection configurator for the slave is opened.

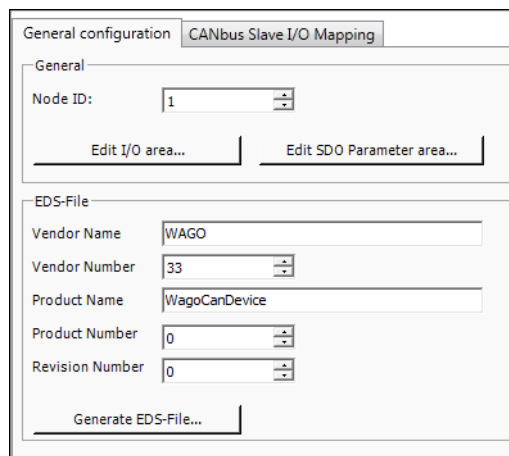


Figure 173: CANopen Connection Configurator (Slave)

13. To add a data point, click first the **[Edit I/O area...]** button in the “General Configuration” tab.
14. Click the **[Add area...]** button in the following dialog.
15. Enter the I/O type, name, number and data type of your data point and confirm via **[OK]**.
Create several data types if required.

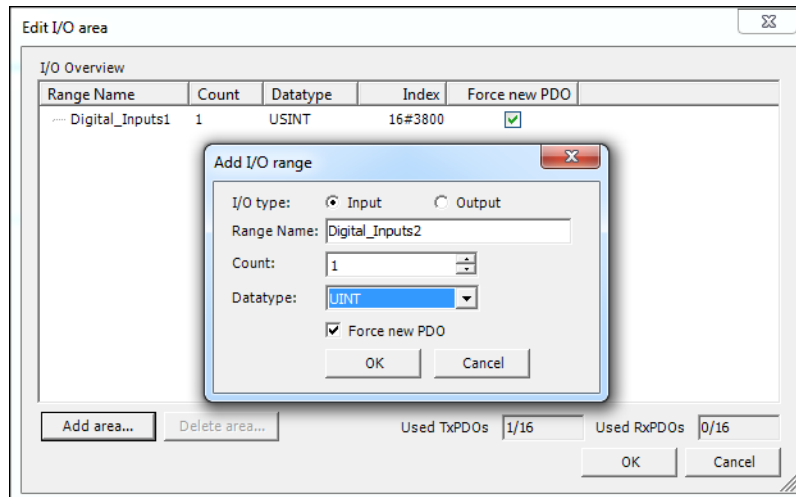


Figure 174: Adding a Data Point

16. Move to the “CANbus Slave I/O Mapping” tab.

The created data points and your channels within the I/O range are displayed here (if the display is incomplete, close the dialog and reopen it.).

Preparing the Master:

17. Select the CANopen master with the mouse.
18. Change the “Node ID” (e.g., “2”) and the baud rate (e.g., 250000) in the “Settings” panel.
Note that “Node IDs” within a CANopen network must always be different.

The configuration of the slave was changed. This is indicated as a warning symbol (yellow triangle with an exclamation mark). The configuration must be updated in order for all created slave data points to be available in the master.

19. To do this click the connector of the master and click the button for updating the configuration.

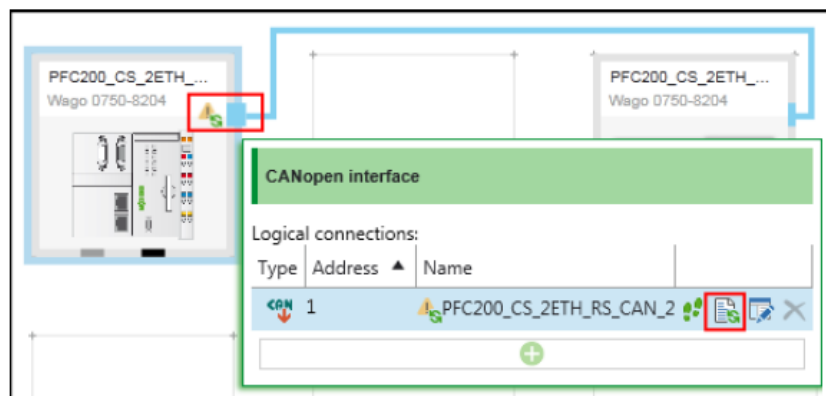


Figure 175: Updating the Configuration

20. In the “Choose Update Behavior” dialog activate the “Create New” checkbox.

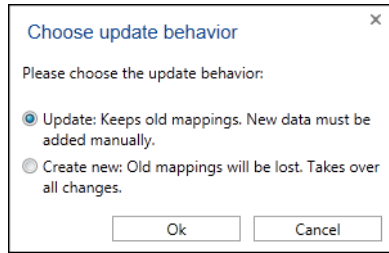


Figure 176: Choosing the Update Behavior

The configuration is updated. The warning symbol disappears.

21. Open the CANopen connection configurator for the master. To do this click the connector of the CANopen master and click the button for opening the connection settings.

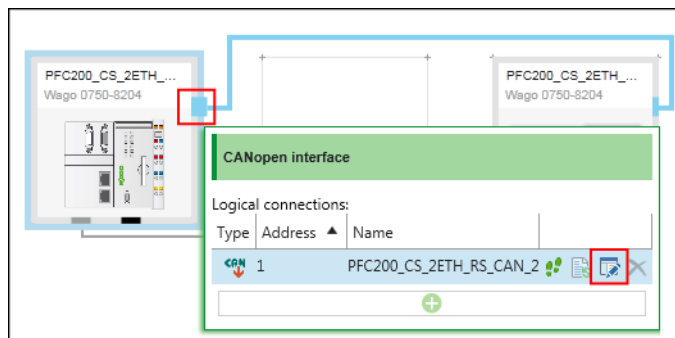


Figure 177: Opening the CANopen Connection Configurator

The CANopen connection configurator for the master is opened.

22. Open the “CANopen I/O Mapping” tab.
23. Assign variables to the channels created in the slave by clicking on the appropriate variable field and entering a variable.

Alternatively, use the Input Assistant by clicking the [...] button.

CANopen Remote Device		PDO Mapping	Service Data Object	CANopen I/O Mapping			
Channels							
Variable	Mapping	Channel	Address	Type	Unit	Description	
	...	Digital_Inputs1_1	%IB1	USINT			
		Digital_Inputs2_1	%IB2	USINT			

Figure 178: Entering Variables

The configuration is then complete. The variables can be used in *e!RUNTIME* programs of both devices.

9.3.4.3 Creating a Simple PROFIBUS Network

To create a PROFIBUS network, you need at least one PROFIBUS master and one slave. The PROFIBUS slave function is provided with a PFC200 (order number 750-8206). The configuration of the PROFIBUS slave requires a connection to a generic PROFIBUS master.

1. Place the generic PROFIBUS master and the PROFIBUS slave in the Network view.

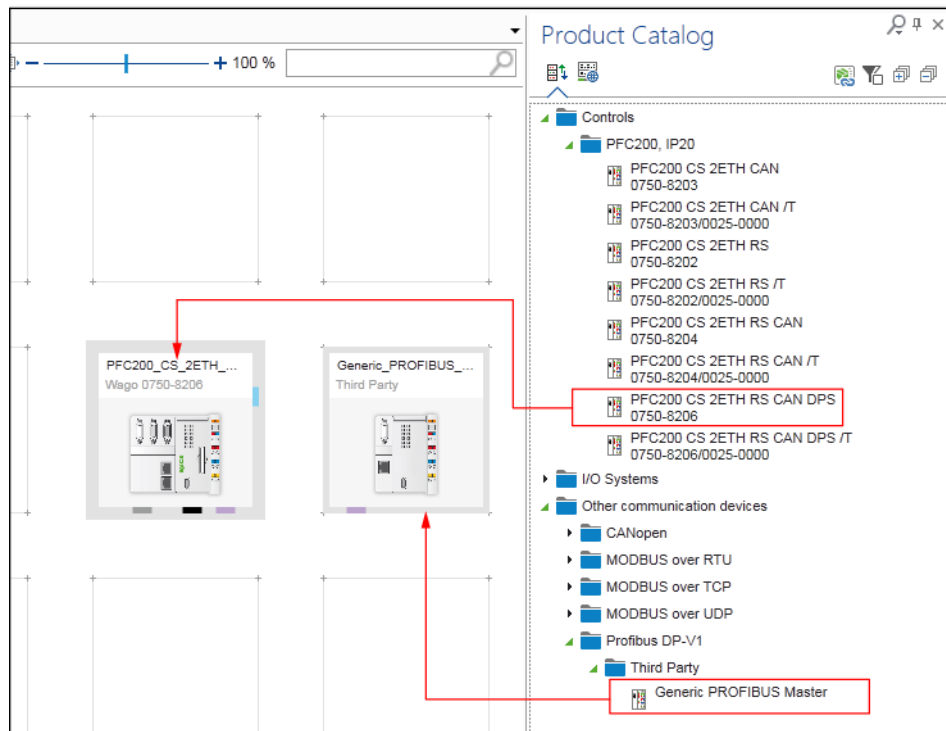


Figure 179: Placing the PROFIBUS Master and Slave in the Network View

2. Create a connection between both devices by connecting up the PROFIBUS connectors (purple) with each other.

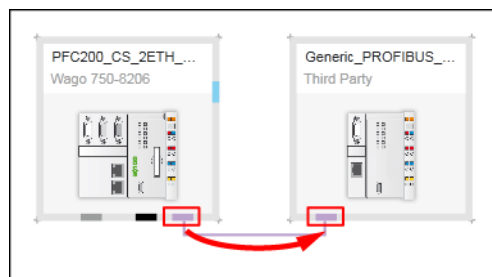


Figure 180: Connecting PROFIBUS Devices

In a PROFIBUS network, each device is assigned a unique address. The PROFIBUS address is set in the “Settings” panel.

3. If the “Settings” panel is not yet opened, open it by right-clicking the slave device and clicking the [Settings] button.

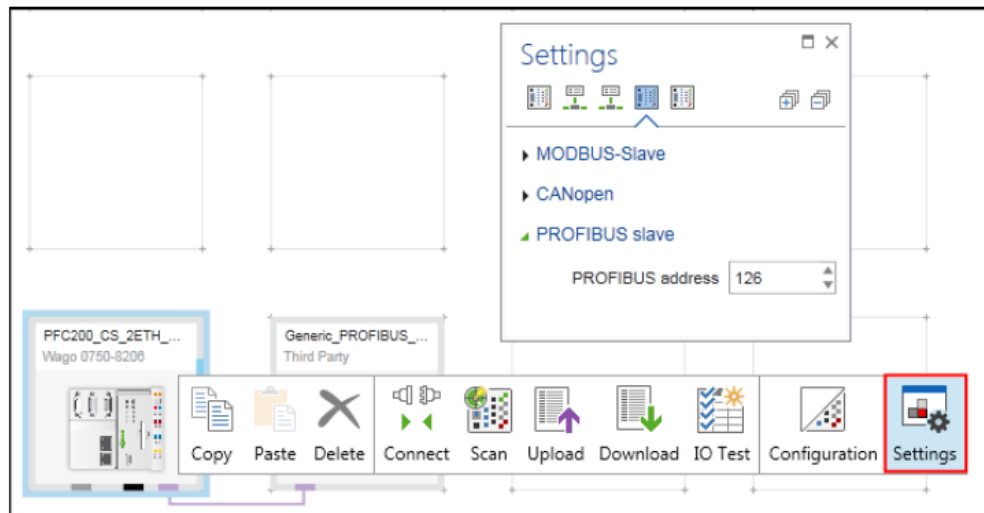


Figure 181: Opening the “Settings” Panel

4. Click the tab in the “Settings” panel in which the PROFIBUS address is set.
5. Enter a unique address in the “PROFIBUS Address” entry field.

The communication in the PROFIBUS network is performed using data points. These are set in the configurator of the PROFIBUS connection for the PROFIBUS slave.

6. Open the configuration interface (PROFIBUS interface) via the context menu of the PROFIBUS connector.

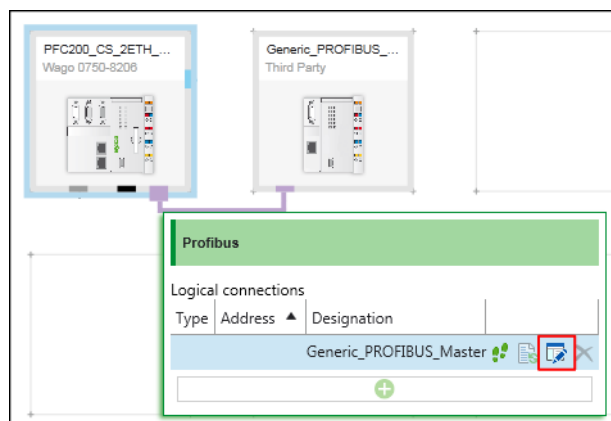


Figure 182: Opening the Configurator of the PROFIBUS Connection

The user interface of the configurator is opened.

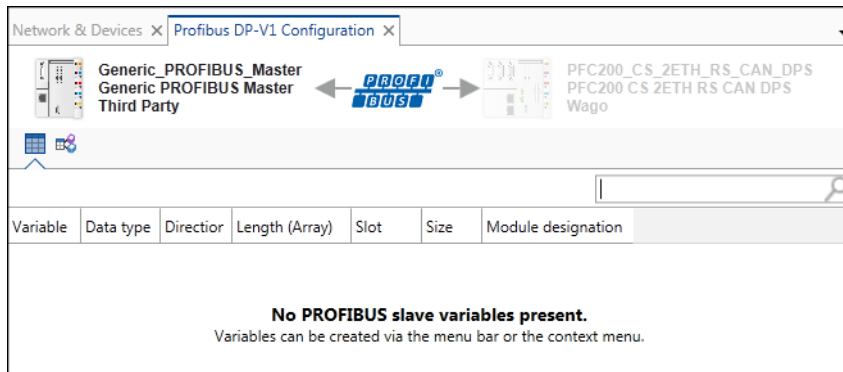


Figure 183: PROFIBUS Connection Configurator

The configuration user interface consists of two main areas, the “Variables” tab and “I/O Mapping” tab, which are arranged beneath the device images.

In the “Variables” tab, defined variables are listed and new ones are created. The “I/O Mapping” tab is used to perform (if required) the mapping to the variables.

7. Create a data point via the context menu in the “Variables” tab or by clicking the appropriate [New] button in the menu ribbon “Configurator: PROFIBUS-DP-V1.”

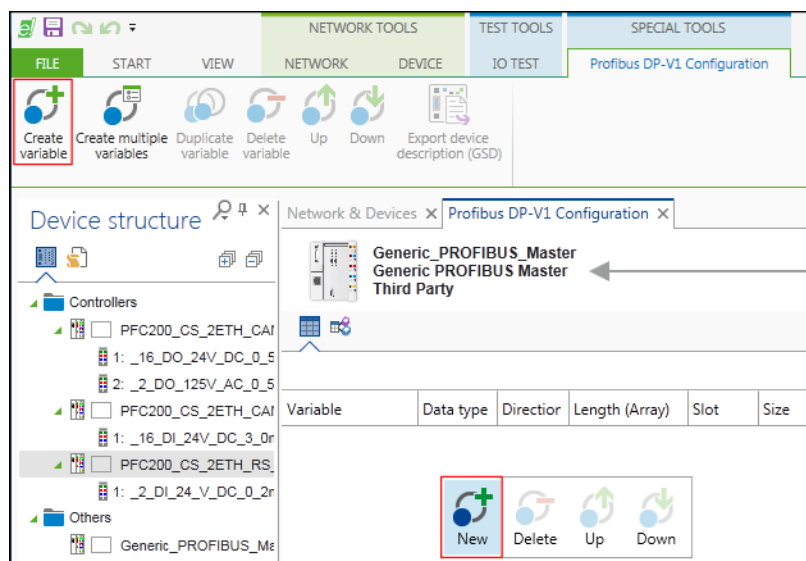
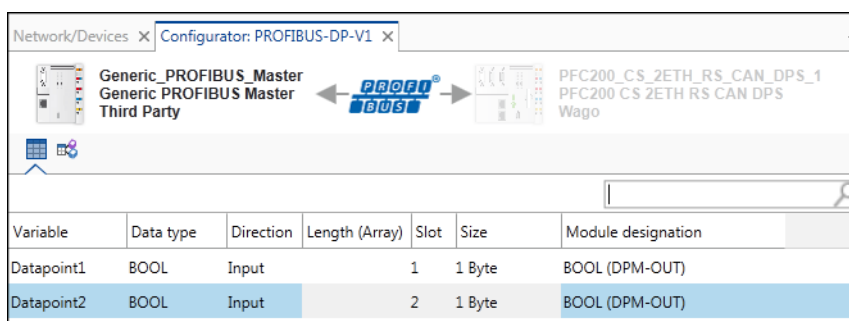


Figure 184: Creating a New Variable

The variable is created. The “Name,” “Data type,” “Direction,” Slot,” “Size” and “Module Designation” are provided with default values. These values are assigned for each new variable created.

Create additional variables and adjust the values as required.

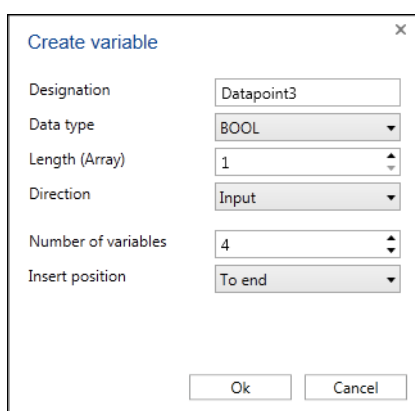


The screenshot shows the 'Configurator: PROFIBUS-DP-V1' window. At the top, there is a network diagram with a 'Generic PROFIBUS Master' (Third Party) connected to a 'PFC200_CS_2ETH_RS_CAN_DPS_1' module (Wago). Below the diagram is a table of configured variables:

Variable	Data type	Direction	Length (Array)	Slot	Size	Module designation
Datapoint1	BOOL	Input	1	1	1 Byte	BOOL (DPM-OUT)
Datapoint2	BOOL	Input	2	1	1 Byte	BOOL (DPM-OUT)

Figure 185: Configured Variables

To create multiple variables at once or to duplicate existing data points, use the corresponding [**Create Multiple Variables**] or [**Duplicate Variable**] buttons in the menu ribbon. In both cases these open a dialog in which multiple values can be set for the newly created variables.



The 'Create variable' dialog box contains the following fields:

- Designation: Datapoint3
- Data type: BOOL
- Length (Array): 1
- Direction: Input
- Number of variables: 4
- Insert position: To end

Buttons: Ok, Cancel

Figure 186: Creating Multiple Variables

Information



Place variables at any location!

If you have not selected a variable from the list, the new variable is placed at the last position. If you have selected one or multiple variables, the new variable is placed behind the last selected variable. This enables you to insert the new variable at any position.

You can delete the variable if required in the same way (context menu of the corresponding variable or [**Delete Variable**] button in the menu ribbon).

After a variable name is assigned, an IEC variable with the same name (global variable list) is created automatically in the background. You can use this variable in your IEC program.

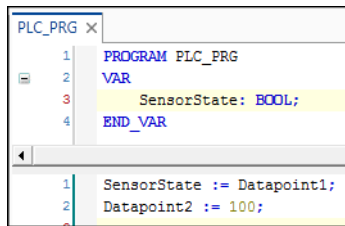


Figure 187: Using Created Variables

8. If you have already created IEC variables and wish to map them to data points, click the “I/O Mapping” tab to open the corresponding configuration user interface.

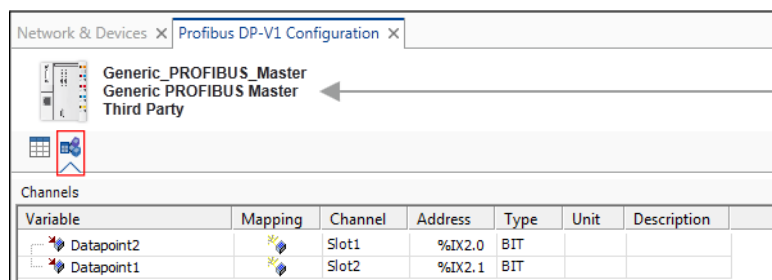


Figure 188: Opening the “I/O Mapping” Tab

There are two ways of mapping a data point to an existing IEC variable:

9. Double-click the corresponding variable name and enter the name of the existing IEC variable manually.

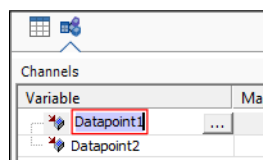


Figure 189: Manual Entry of the IEC Variable

10. Or click the [...] button to open the Input Assistant.

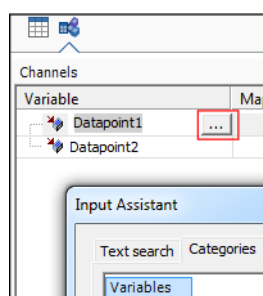


Figure 190: Entering IEC Variables via the Input Assistant

11. In the Input Assistant select the IEC variable and confirm with [OK].

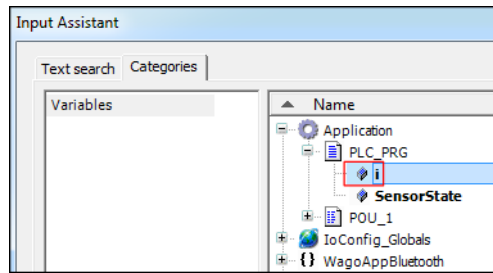


Figure 191: Selecting a Variable

The mapping is performed.

Variable	Mapping	Channel	Address	Type	Unit
Application.PLC_PRG.P...		Slot1	%IX1.0	BIT	
Datapoint1		Slot2	%IX1.1	BIT	

Figure 192: Mapping Completed

After the variables are created and optional mapping, you can save the configuration in the form of a device description file.

- To do this, click the **[Export device description (GSD)]** button.

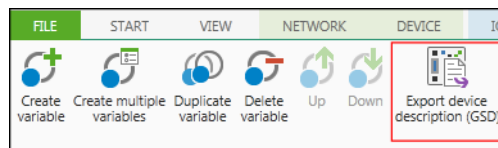


Figure 193: Saving the Device Description File

The “Save As” dialog for entering a file name is opened.

- Enter a name and confirm with the **[OK]** button.

Two files are saved. The two files contain identical configurations. Only the description language of the parameters is different. The file with the suffix “*.gsg” contains German text. The file with the suffix “*.gsd” contains English text.

The exported device description can be imported by other engineering tools.

9.4 Testing

9.4.1 Performing an Input/Output Test for Digital Input/Output Modules

To verify the wiring of digital input/output modules, switch an LED on or off with *e!COCKPIT*.

The input/output test requires the device to be connected online and Control mode to be active (“I/O-CHECK” tab).

1. Open the Device Detail view.
2. Click the LED of the relevant module.

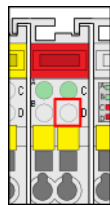


Figure 194: Switching an LED On/Off with the Mouse

3. Verify if the device’s LED mirrors its on-screen behavior in *e!COCKPIT*.

If this is the case, the module is connected correctly.

The display is reset as soon as the Device Detail view is closed.

9.4.2 Performing a Systematic Input/Output Test (I/O CHECK)

Check for correct wiring of your node using the systematic input/output test. You can test both individual modules and entire nodes.

Testing Individual Modules:

1. Open the Device Detail view.
2. Click a module.
3. Click the **[Build]** button in the “IO-CHECK” tab of the menu ribbon.

A tab is opened in the workspace, in which the selected module is displayed.

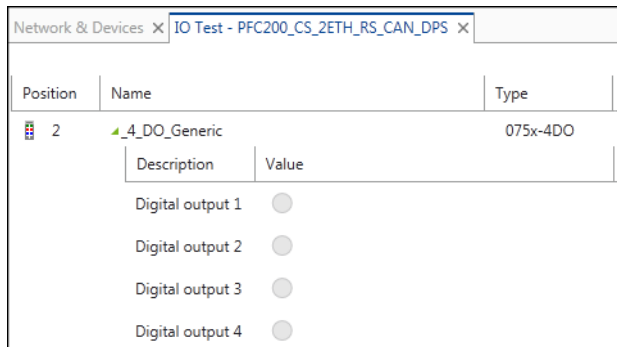


Figure 195: Display of an I/O Module for the Input/Output Test

- To add additional modules to this test, repeat steps 1–3.

Testing an Entire Node:

- Open the Network view.
- Click a device.
- Click the **[Build]** button in the “IO-CHECK” tab of the menu ribbon.

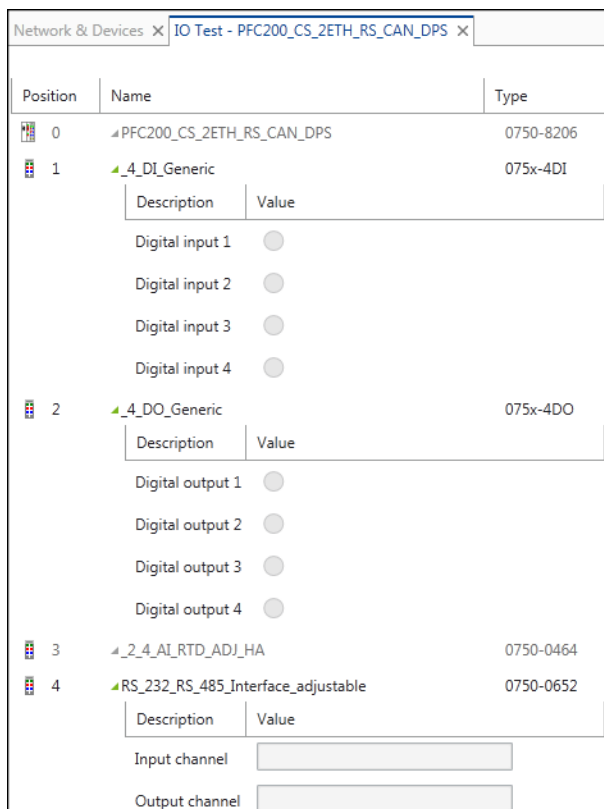


Figure 196: Display of a Node for the Input/Output Test

An overview of input/output channels of the corresponding modules is displayed in both cases.

- Click the **[Connect]** button in the menu ribbon.

The monitoring function is active first. The current values are displayed. If required, the “Result” column can be switched on/off via the button of the same name in the menu ribbon.

5. Compare the displayed values with your hardware, e.g., whether the green LED is actually lit and for each entry click either **[Ok]** or **[Error]**.

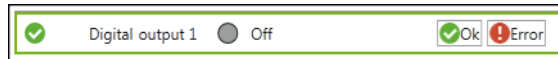


Figure 197: Performing a Test

Depending on your entry, the status is shown in the “Result” column.

Result	Position	Designation	Item number
	0	↵PFC200_CS_2ETH_RS_CAN_DPS	0750-8206
! ✓	1	↵_4_DI_Generic	075x-4DI
		Result Description Engineering value Result	
		✓ Digital input 1 <input type="radio"/> Off	✓Ok !Error
		! Digital input 2 <input type="radio"/> Off	✓Ok !Error
		! Digital input 3 <input type="radio"/> Off	✓Ok !Error
		✓ Digital input 4 <input type="radio"/> Off	✓Ok !Error
! ✓	2	↵_4_DO_Generic	075x-4DO
		Result Description Engineering value Result	
		✓ Digital output 1 <input type="radio"/> Off	✓Ok !Error
		! Digital output 2 <input checked="" type="radio"/> On*	✓Ok !Error
		✓ Digital output 3 <input checked="" type="radio"/> On*	✓Ok !Error
		✓ Digital output 4 <input type="radio"/> Off*	✓Ok !Error
	3	↵_2_4_AI_RTD_ADJ_HA	0750-0464
✓	4	↵RS_232_RS_485_Interface_adjustable	0750-0652
		Result Description Engineering value Result	
		✓ Input channel 0	✓Ok !Error
		✓ Output channel 0	✓Ok !Error

Figure 198: Status Display for Inputs and Outputs

Besides value monitoring, you can also write or read them.

6. Switch to control mode by clicking **[Control mode]** in the menu ribbon.

Change the individual values. For this use, the slide adjuster and entry fields for analog output terminals; for digital output terminals click, the LED symbol.

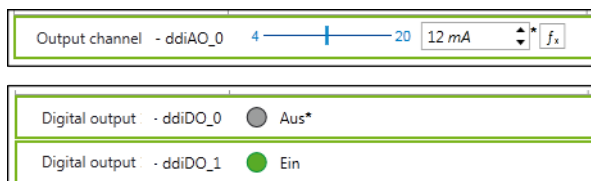


Figure 199: Changing Analog and Digital Values

Changed values are indicated with an asterisk (*) and the number of prepared changes shown as a number in the menu ribbon (here: 3) if “Collect” is active (blue background).

7. To load all changed values to the I/O module click the **[Write now]** button in the menu ribbon. The write operation is performed once.

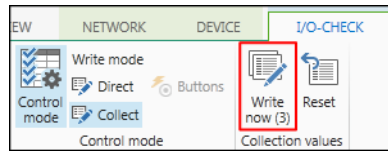


Figure 200: Write Collection Values

If the **[Direct]** button is clicked, the values are continuously written as they are entered.

9.5 Programming

e!RUNTIME is integrated in *e!COCKPIT* user interface. This is the 3S hardware-independent CODESYS IEC 61131-3 programming system (version 3.5).

e!COCKPIT can assign elements for program development to (fieldbus controllers), thus enabling for example the implementation of fieldbus and I/O configurations, visualizations, motion control and other tasks.

1. Open the “Programming” workspace.

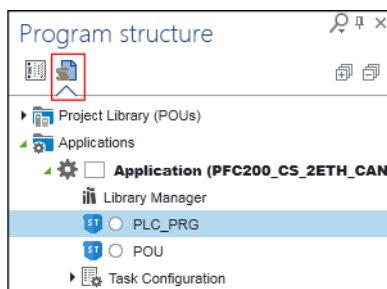


Figure 201: Opening the “Programming” Workspace

A CODESYS application with a library manager, “PLC_PRG” program and a task configuration is created for each head station incorporated. Other applications can be created on the same level or below.

A “project library” is created for POU, which applies to all applications.

You can incorporate additional CODESYS elements via the context menu in the Program Structure, such as additional POU, files, variable lists or visualizations (see CODESYS online help, “Adding an Object”).

Right-click the project library and choose [**C**opy] to copy, for example the entire content of this directory, to a directory of an appropriate project application. Copying and inserting is also possible between different instances of *e!COCKPIT*.

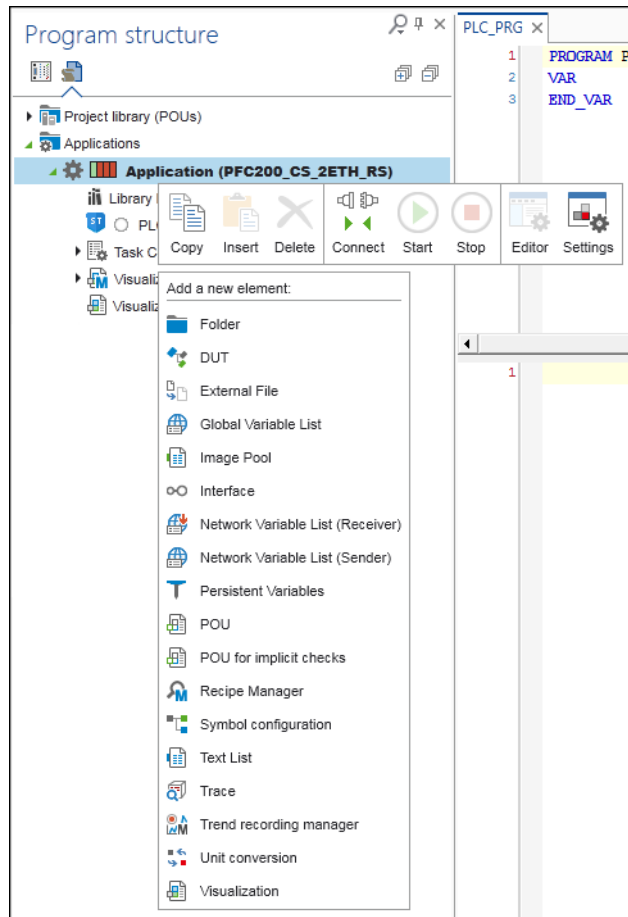


Figure 202: Context Menu in the Program Structure

“POUs” are program organization units/objects that form a controller program. When you create a POU, the workspace is adapted accordingly. Depending on the selected programming language, other tabs are added to the menu ribbon and new panels are shown (e.g., “Tools”).

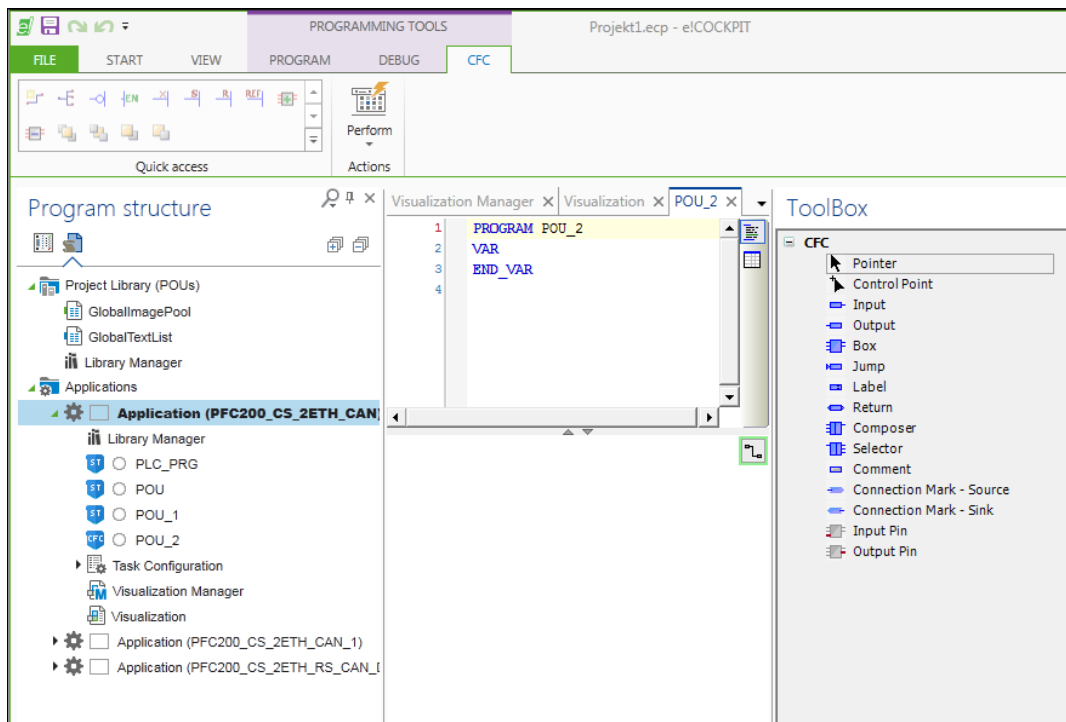


Figure 203: Adding a POU

The use of individual panels is described in the CODESYS online help.

The context menu of an application also enables you to copy, insert or delete the appropriate application, connect the device, start and stop the application, open the programming editor or view/change the settings of the application.

9.5.1 Creating Libraries

1. Select a library template in the Start or Backstage view.

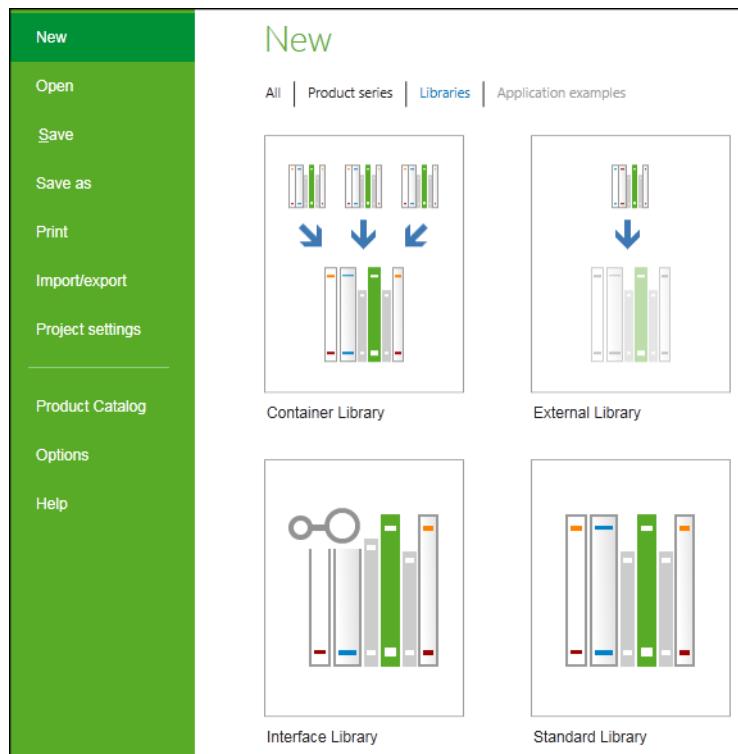


Figure 204: Selecting a Template for Libraries

The main view is opened.

2. Open the “Programming” workspace.
3. Create your library.

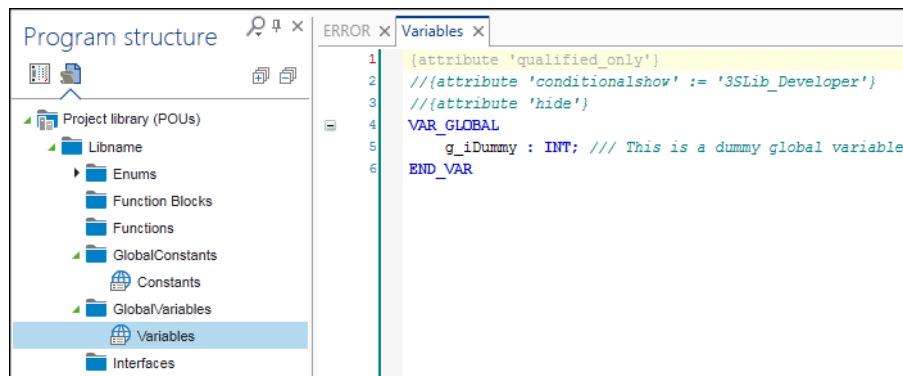


Figure 205: Creating a Library

4. To check the created library, click the **[Syntax check project library]** button in the “PROGRAM” tab of the menu ribbon.

This checks the written library according to IEC 61131-3.

5. After a successful check move to the Backstage view (“FILE” tab).

6. Click [**Import/Export**].
7. Click the [**IEC library**] button.
8. Saving and installing a library:
 - To save the library as a compiled library with the “compiled-library” suffix, click [**Compile and save**].
 - To save the library in the system repository, click [**Save in library repository**].

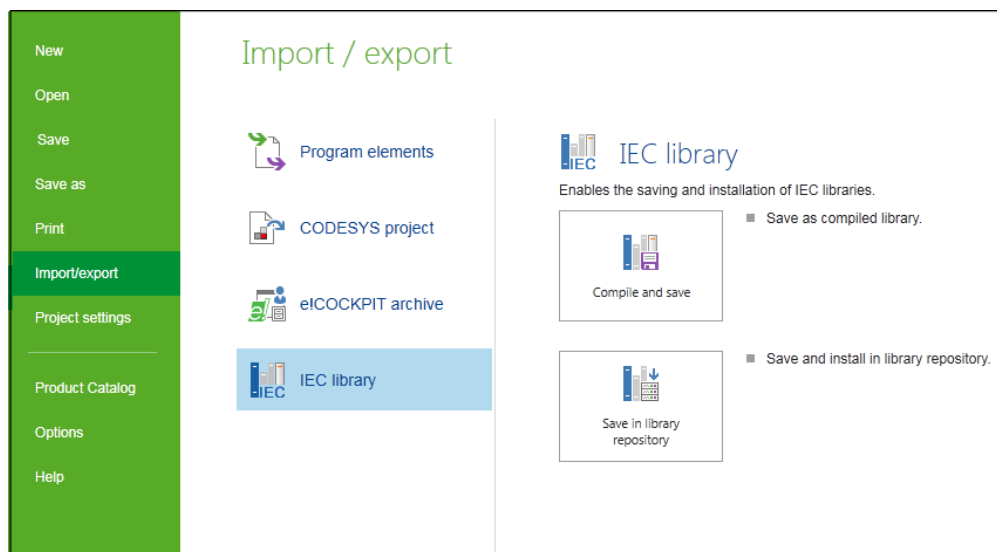


Figure 206: Saving and Installing a Library

9.5.2 Adding Libraries

Libraries contain collections of reusable objects such as functions, function blocks or variables that are incorporated into projects. The contents of these libraries can be used here without any new declaration. The library manager is used to manage the libraries in a project. This displays both application-related and function-related libraries as well as CODESYS and WAGO system libraries.

WAGO system libraries are already supplied. Add libraries that are not present as follows.

1. Open the “Programming” workspace.
2. Right-click the “Project Library (POUs)” entry and click **Library Manager** in the context menu.

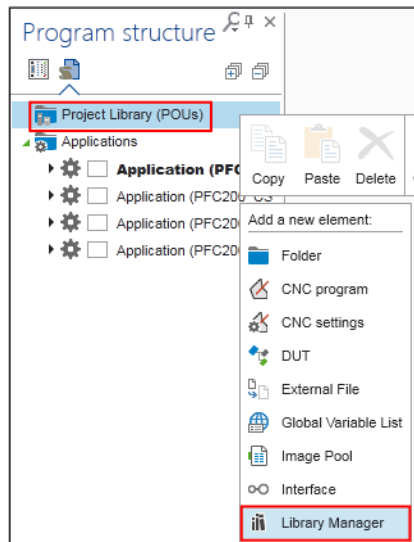


Figure 207: Adding the Library Manager

3. Click [**Add**] in the dialog.

The library manager is opened in the workspace.

4. To add a new library that has not been installed on the system, click [**Add library**].
5. Select the relevant library in the dialog.
6. Confirm your selection with [**OK**].

The library manager incorporates a reference to a pre-installed library into the project.

To add a library that is not yet installed on the local system, click the [**Library repository**] button to install it.

To add target-system neutral libraries, use the [**Placeholders**] button instead. Depending on the device used and its device description, the appropriate library is used. Alternatively, the standard library is used if no device is present, so that the program can be compiled error-free.

Note



Install libraries in e!COCKPIT!

Note that all libraries must be incorporated permanently. If you simply replace the library files in your local library directory, the libraries are not available in e!COCKPIT. They must be installed from the repository.

9.5.3 Creating Programs

The “PLC_PRG” main program is created automatically. You can create additional programs.

1. Right-click the corresponding application and select **[POU]** in the context menu.
2. In the dialog assign a name and select the type and the implementation language per IEC 61131-3.

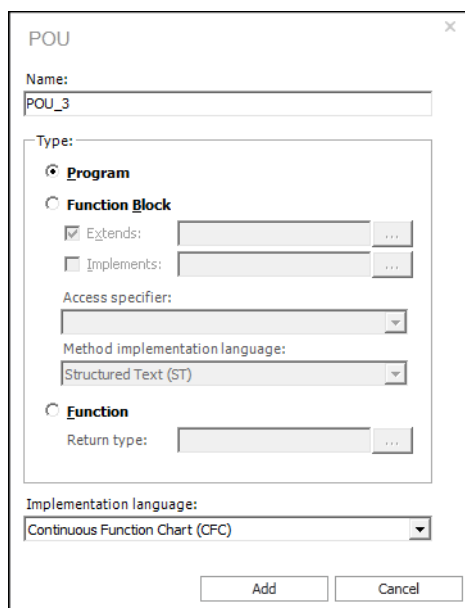


Figure 208: Creating a Program

The programs are automatically assigned to a task. If the task was deleted, you can create this again by right-clicking the application and selecting **[Task Configuration]** in the context menu.

3. To assign a program to the task, click the created task and add a **Program call**.

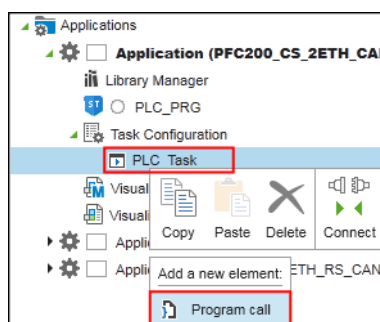


Figure 209: Adding a Program Call

A “Program call” dialog is opened.

4. In the “POU to Call” field, enter the program name.

Alternatively, use the Input Assistant by clicking the [...] button, select the program in the new window and confirm with **[OK]**.

Name	Type	Address
Application	Application	
PLC_PRG	PROGRAM	

Figure 210: Entering the Program Name via the Input Assistant

The program name is entered in the “Program call” dialog.

- To add the program call, click the **[Add]** button.
- To open and edit the added program double-click the program in the Project Structure.

Depending on the implementation language you have selected under point 2, the corresponding programming editor is displayed.

```

PLC_PRG x
1  PROGRAM PLC_PRG
2  VAR
3  i: INT;
4  END_VAR

1  i:=i+1;
2

```

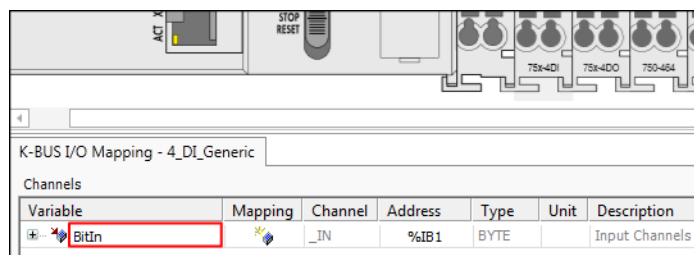
Figure 211: Programming Editor (in this example: Structured Text (ST))

- Write your program according to IEC 61131-3.

9.5.4 Performing I/O Mapping

By I/O mapping is meant the mapping and assignment of input, output, and memory addresses of the controller to project variables that are used by the application. The I/O mapping is displayed in the Device Detail view below the device image.

- Open the Device Detail view.
- Select the I/O module with the mouse.
- Click the “Variables” column in the corresponding fields and enter the variables/data points.



Variable	Mapping	Channel	Address	Type	Unit	Description
BitIn		_IN	%IB1	BYTE		Input Channels

Figure 212: Entering Variables/Data Points

The variable or data point is created and is available in the entire project.

You can use the Input Assistant to assign variables that are already known in the project.

4. To do this, double-click a field.
5. To open the Input Assistant, click the [...] button.

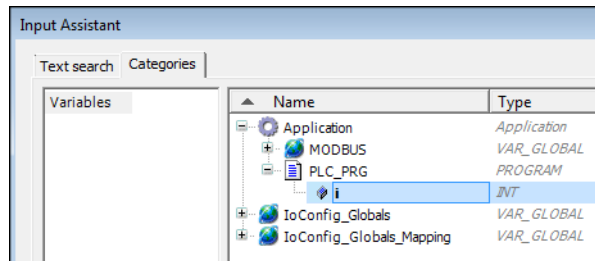


Figure 213: Entering Variables/Data Points via the Input Assistant

6. Select the corresponding variable or the data point and confirm with [OK].

The mapping is accepted for the project.

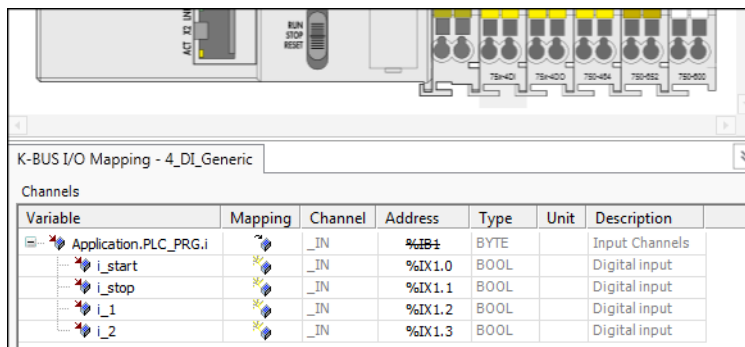


Figure 214: I/O Mapping

9.5.5 Compiling an Application

1. To compile the program click the **[Build]** button in the “PROGRAM” tab of the menu ribbon.

The program is compiled and a message bar shows the errors.

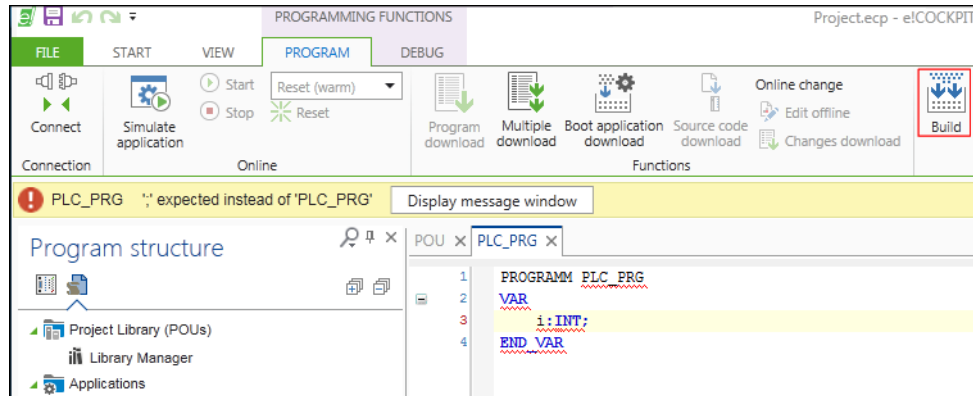


Figure 215: Compiling an Application

2. To display details of the validation, click **[Display message window]**. Alternatively, open the message window via the **[Messages]** button in the lower left area of the workspace.
3. All errors, warnings and messages are displayed first. To filter the view, click the appropriate button in the message window. Active displays are highlighted in blue.

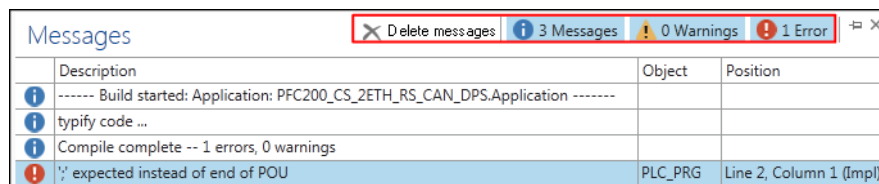


Figure 216: Message Window

4. Click an entry in the list to jump directly to the error location.
5. To clean the list, click the **[Delete messages]** button.

9.5.6 Excluding Programs for Compiling

If you wish to connect and compile an application, you can exclude individual programs that must not be compiled.

1. Right-click in the Program Structure the program that must not be compiled, e.g., “PLC_PRG.”
2. Click **Settings** in the context menu.
3. In the dialog click the “Build” tab and activate the “Exclude from compiling” check box.

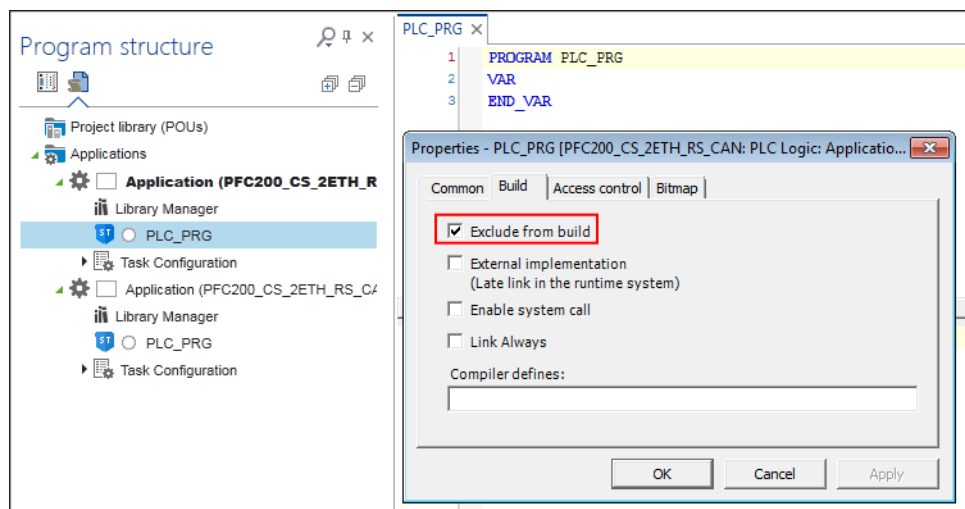


Figure 217: Excluding the Program from Compiling

4. Click **[OK]** to confirm.

The program is displayed in gray and in italics.

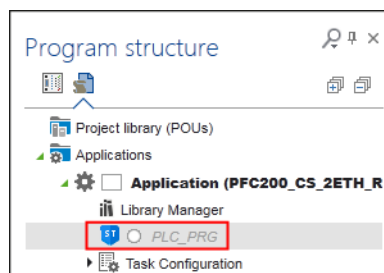


Figure 218: Program Excluded from Compiling

9.5.7 Debugging an Application

The debug function examines programming errors. This requires a connection to the device.

1. Move to the “DEBUG” tab in the menu ribbon.
2. To start the debug process, click **[Start]**.
3. To create a watch window, click the **[Monitoring]** button in the menu ribbon and select one of the five entries.

This will open the “Watch” panel.

4. Double-click the “Expression” field and then the [...] button displayed there for the Input Assistant.
5. Select the value or values to be monitored.

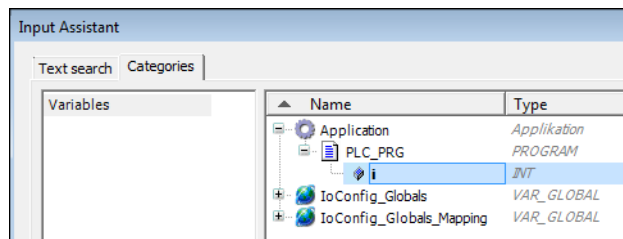


Figure 219: Selecting Values for Monitoring

6. Transfer the value or values by clicking **[OK]**.

Watch 1						
Expression	Execution point	Type	Value	Prepared value	Address	Comment
PF200_CS_2ETH_RS_CAN_DPS.Application.PLC_PRG.I	Zyklisches Monitoring	INT	11560	<Applicationnotf...		

Figure 220: Watch List

7. Click the **[New]** button in the “Breakpoints” group to add breakpoints.

Breakpoints can be set at specific positions in the program to stop execution. Specific conditions can be defined for the stop (see CODESYS online help, “Breakpoints”).

8. Then click **[Toggle]** to activate the breakpoint.

Step-by-step processing via **Single Step** (menu ribbon, “DEBUG” tab) enables the program to be run in controlled steps. The particular variable values can be examined at each stop. A **Call stack** (menu ribbon, “VIEW” tab) can be displayed to determine the current position.

The flow control can be activated to track previous sections of the application program. Unlike standard monitoring, which only displays the value of a variable between two cycles, the flow control supplies the value for each processing step precisely at the time of processing.

9.5.8 Simulating an Application

Simulation mode enables you to debug and test the behavior of an application without connecting to a real target system. No communication settings are required for the simulation.

1. Open the “PROGRAM” tab in the menu ribbon.
2. Click [**Simulate application**].

The button in the menu ribbon changes to [**Close simulation**]. In the status bar, the status display in front of the corresponding device and in the tooltip of the device (“Program structure” panel) the connection is shown in blue. Also in the Network view the tile of the corresponding device is outlined in blue.

3. Use the commands of the “DEBUG” tab to test the application.

9.5.9 Creating Visualizations

e!COCKPIT integrates the visualization functions of CODESYS. These enable the use of predefined visualization elements and commands for simulating, controlling and monitoring machine and plant systems.

A web server connection (CODESYS-Web-Visu) also enables a visualization to be operated with a web browser or displayed on an external device/display (CODESYS-HMI).

Note



Ensure compatibility with web browsers!

The view of the web visualization has been tested and optimized for the following web browsers:

- Firefox 30.0
- Chrome 34.0, 35.0
- Internet Explorer 11
- Opera 12.17
- Android 4.0
- Safari iPad2

Note



If the connection establishment fails, reduce the number of visualization tabs!

If many tabs containing the visualizations are opened in the programming workspace, it may not be possible to make a connection to a device. Use the “Close all windows apart from this one” function in the context menu of the tab to retain only the current tab and close all the others. Try again to connect the application.

1. First open the “Programming” workspace.
2. Add a visualization by right-clicking an application and selecting **[Visualization]** as a new element.

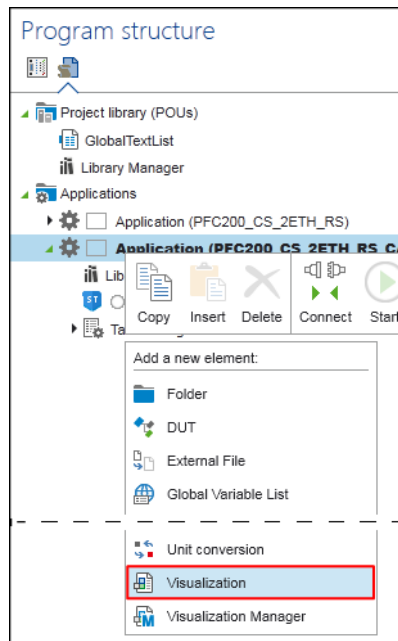


Figure 221: Adding a Visualization

The visualization editor is opened in the workspace.
This will display the “ToolBox” and “Properties” panels.

- Use the elements from the “ToolBox” panel, such as the bar display, by placing them in the visualization editor via drag and drop.

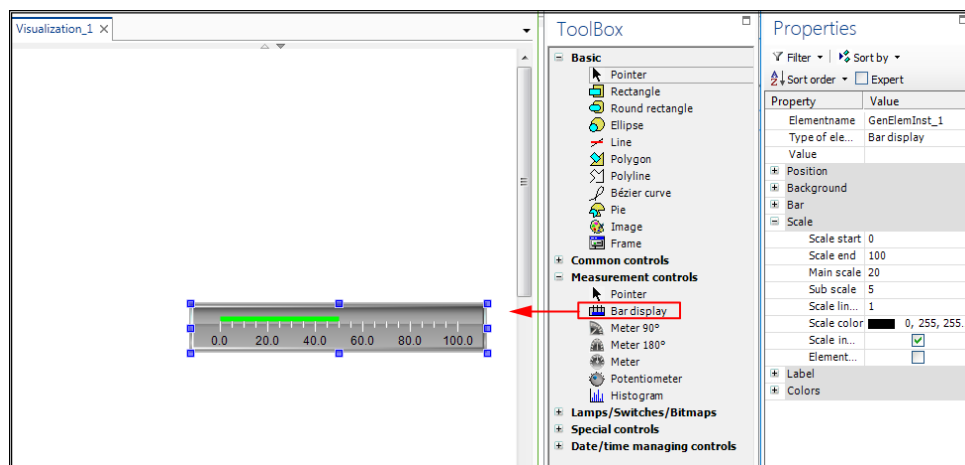


Figure 222: Creating a Visualization

Assign a value to the display:

- Click the “Value” field and press **[F2]** to open the Input Assistant.
- Via the Input Assistant select a value which will be used for the visualization. This example uses the value “i” from the “PLC_PRG.”

The value is accepted:

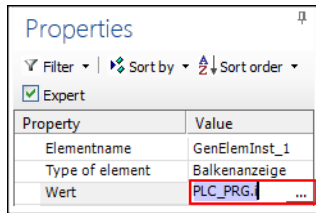


Figure 223: Assigning a Value

6. In the “DEBUG” tab of the menu ribbon, click [**Simulate application**].
7. To start the visualization click [**Start**].

This will start the visualization.

9.5.10 Connecting

To perform online actions such as a program download, program execution etc. for this application, connect the application with the target system.

1. **“Network/Devices” Workspace:** In the context menu of a device or via the menu ribbon (e.g., “DEVICE”, “NETWORK” tabs) click [**Connect**]. The device is switched to online mode. All applications of the device are active/connected online.

“Programming” workspace: In the context menu of one or several selected applications or in the menu ribbon (“PROGRAM”, “DEBUG” tabs) click [**Connect**]. The selected applications of one or several devices are active/connected online.
2. If an application was not previously stored on the controller, a prompt will ask whether an application should be created or downloaded. Confirm this with [**Yes**].

9.5.11 Downloading Applications to the Device

1. To download the selected application to the controller, click [**Program Download**] in the “PROGRAM” menu ribbon.

To load multiple applications/parameters on the device and set other options (e.g., Online Change), click [**Multiple download**] in the “PROGRAM” menu ribbon.

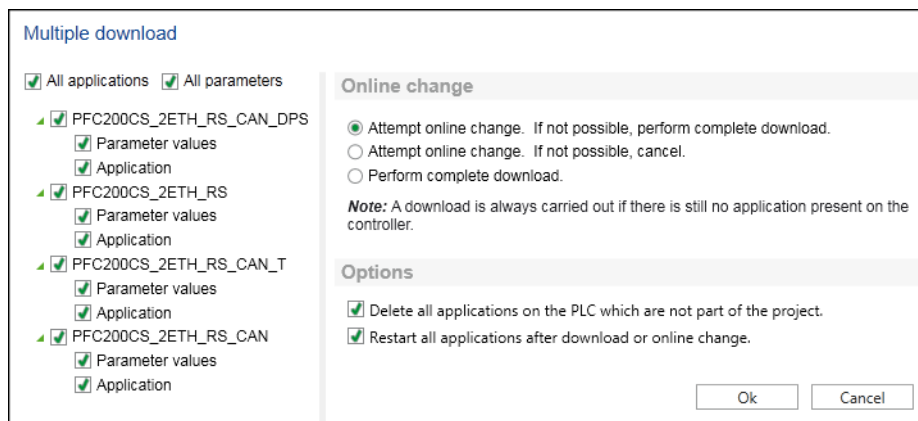


Figure 224: Multiple Download

2. All devices with associated applications and parameters are preselected by default. Use the mouse to untick components that are not to be loaded as required.

9.5.12 Importing CODESYS 2 and CODESYS 3 Projects

To load projects that you have created in CODESYS 2 (file suffix “.pro”) or CODESYS 3 (file suffix “.project”):

1. Open Backstage view (“FILE” tab).
2. Click the **[Import/Export]** buttons and then **[CODESYS project]**.

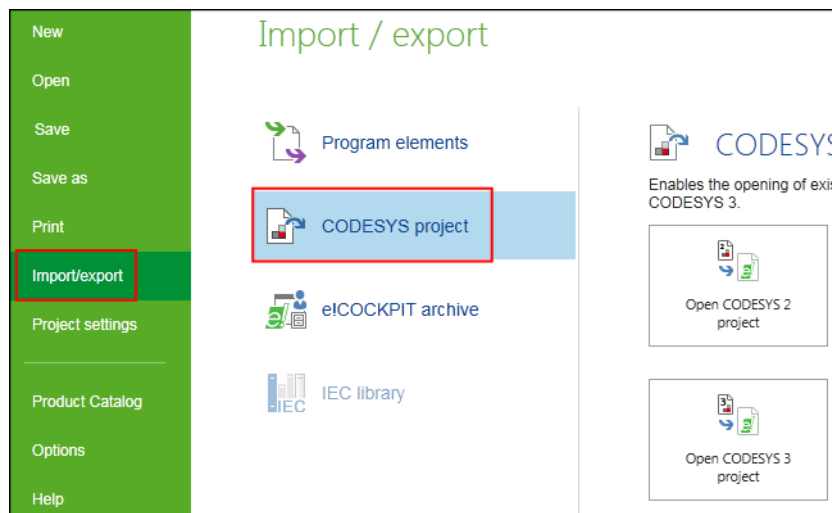


Figure 225: Loading a CODESYS Project

3. Click **[Open CODESYS 2 project]** or **[Open CODESYS 3 project]**.
4. Select the project file and click **[Open]**.

This starts the project import.

e!RUNTIME uses new libraries which have different names and content to the libraries of previous CODESYS projects.

5. If the following window is displayed for converting a library reference, first select the following setting: “Ignore the device. All application specific objects will not be available in the new project.”

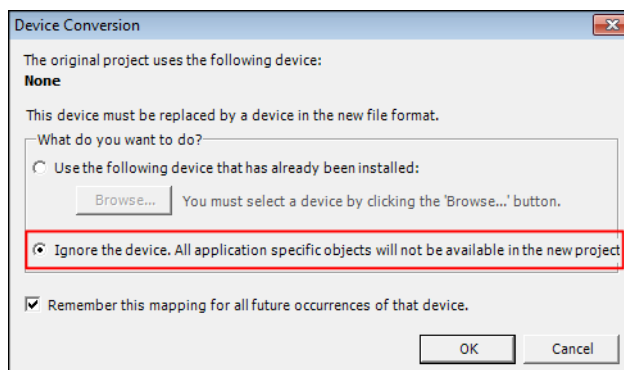


Figure 226: Converting a Library Reference

The window is displayed for each new library not found. You require a new version for these libraries.

6. Click **[OK]**.

This allows you to only load programs and devices first of all. You can update new libraries later.

Information



Additional Information

Further information on the libraries are provided in the manual “Libraries for *e!COCKPIT* (Overview and Migration Instructions)”.

The manual provides an overview of libraries which can be used in *e!COCKPIT*. Instructions are provided for migrating software projects which were created in the *WAGO-I/O-PRO*, *CODESYS 2* or *CODESYS 3* programming environments. The appendix of this manual provides an overview of corresponding functions in libraries for *WAGO-I/O-PRO* and *e!COCKPIT*.

Further instructions on converting between *CODESYS 2* and *CODESYS 3* projects are provided in the *CODESYS* online help (“Converting”).

If a device of the imported project is unknown, a suggestion list of available and compatible devices is displayed.

Note



Applications may be discarded when using compatible devices!

If the compatible device does not support all existing applications, a warning message will be shown before you replace the device.

Unsupported applications are discarded.

Select a corresponding device.

7. To accept the device click **[Replace device]**.

The main view is opened after the *CODESYS* project is loaded.

8. Open the “Programming” workspace.

Load the “Standard.lib” library first:

9. Double-click the library manager of the imported application.
10. Click **[Add library]** and select the “Standard” library.

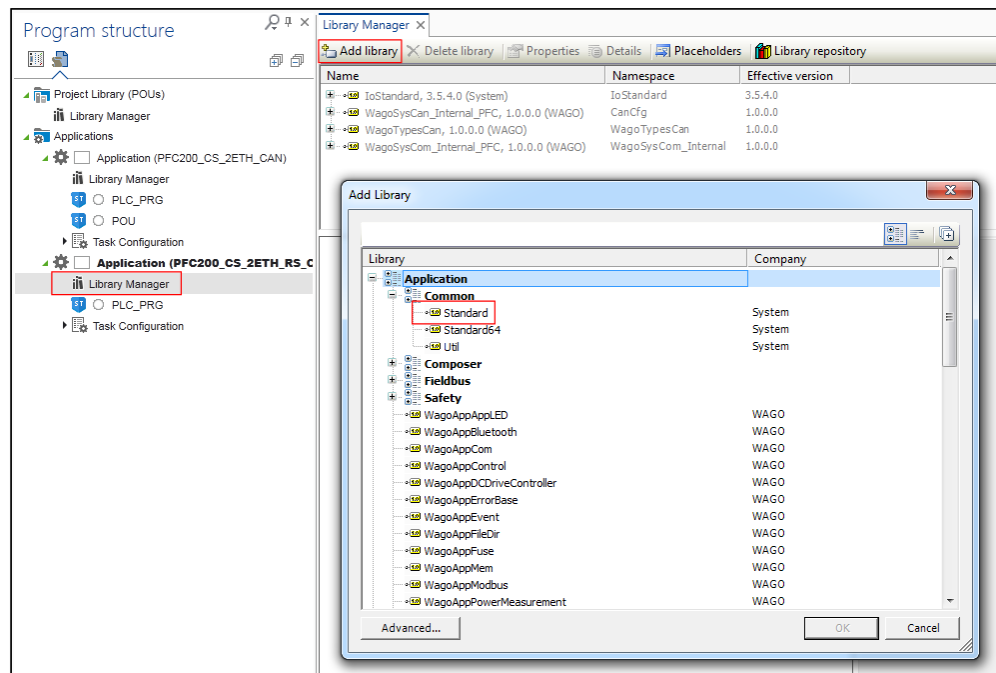


Figure 227: Adding a Library

11. Open your program/s and click **[Build]** in the “PROGRAM” tab.
12. If the message bar is opened, click **[Display message window]**.
13. Adjust your program according to the messages.
14. Load missing libraries (see point 5) via the library manager. Both libraries for your devices are available at <http://www.wago.com>.

9.5.13 Importing/Exporting Program Elements

In order to import or export program elements, use the appropriate button [**Import**] or [**Export**] in the menu ribbon (“PROGRAM” tab). Alternatively access these functions via the Backstage view, “Import/Export” page > [**Program Elements**].

Exporting

1. Click the [**Export**] button.

The “Export” dialog window is opened and the Program Structure is displayed.

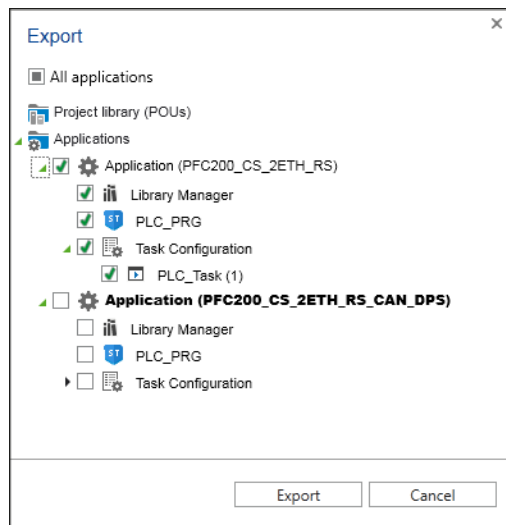


Figure 228: Exporting Program Elements

2. Tick the elements to be exported.
3. Click [**Export**].
4. Select the memory location in the file selection dialog and enter a file name.
5. Export the program elements by clicking [**OK**].

A file with the suffix “*.export” is saved.

Importing

1. Click the [**Import**] button.

This opens a file selection dialog box.

2. Select the “*.export” file to be imported and click [**Open**].

The “Import” dialog is displayed.

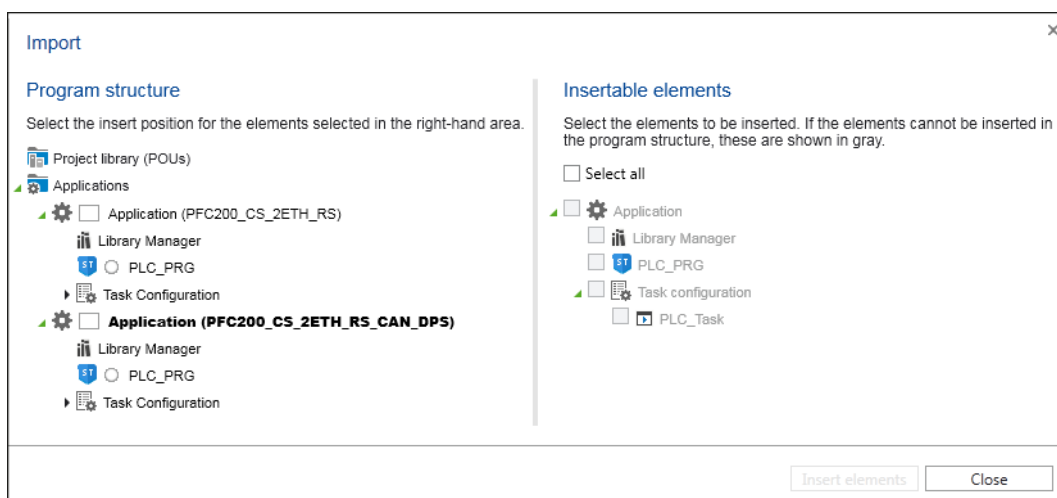


Figure 229: Importing Program Elements

The Program Structure is displayed in the left-hand area of the dialog. This is used to select the position where the imported elements are inserted.

The content of the import file is displayed in the right-hand area. The content is context-sensitive according to the selection in the Program Structure. If the content to be imported cannot be inserted at the current position in the Program Structure, the relevant objects are grayed out.

3. To insert the selected objects at the selected insert position click [**Insert Elements**].

The dialog does not close automatically. You can select a succession of elements and insert them at different positions. Multiple selection is also possible. This only allows the simultaneous selection of objects of the same class (e.g., programs). The selected objects are inserted at several positions at the same time.

4. Click [**Close**] to finish the import.

9.6 Setting and Managing

9.6.1 Documenting and Printing a Project

Overviews of all components contained in a project can be printed for project documentation.

1. Open Backstage view.
2. Click [**Print**].
3. Click [**Document project...**] and [**Document**].

A dialog will open, which displays the project components. Components which are printed are indicated with a green tick.

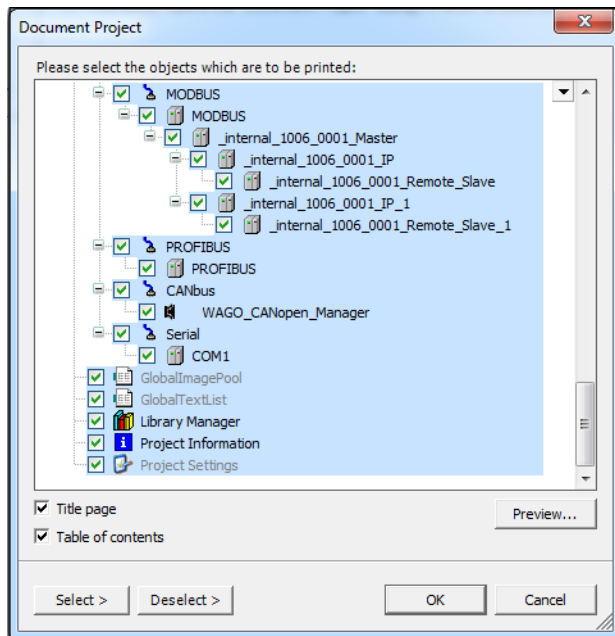


Figure 230: Documenting a Project

4. Untick components that are not to be printed as required.
5. Tick the appropriate checkboxes to create a **Title Page** or **Table of contents**.
6. Click [**OK**] to print the project.

An additional print function is provided for active editors in the “Programming” workspace.

1. Open the “Programming” workspace.
2. Open an element by clicking it in the Program Structure, such as a “PLC_PRG” program.

3. Move to the “START” tab.
4. Click [**Print**].

The content of the active editor is printed.

9.6.2 Managing Devices

To operate devices in *e!COCKPIT*, drivers containing device-specific description files are imported and interpreted by *e!COCKPIT*. These files contain information such as, device properties, interfaces and supported protocols.

The following device description files are supported:

- “Device Type Package” (DTP) for ETHERNET/MODBUS devices
- “Electronic Data Sheet” (EDS) for CANopen devices
- “Device master files” (GSD) for PROFIBUS devices
- “Device Configuration File” (DCF)

Device manufacturers provide these description files for their devices.

A device description file can apply to several similar devices. It is also possible to install several versions and use these in the same or different projects.

Devices and installed device description files are managed on the “Product Catalog” page of Backstage view.

1. To move to the Product Catalog management in the Backstage view, click the corresponding button the Product Catalog panel. Alternatively click the “FILE” tab and the [**Product Catalog**] button.

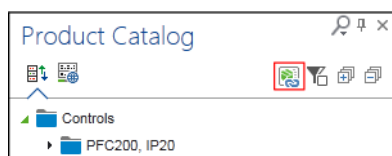


Figure 231: Moving to the Product Catalog Management

9.6.2.1 Adding Devices

When *e!COCKPIT* is installed, description files are also installed for standard WAGO devices so they can be displayed and used directly in the Product Catalog.

If you require WAGO or other devices, proceed as follows to import the description files for these devices on the “Product Catalog” page in the Backstage view:

1. Click the [**Add devices**] button.
2. In the “Open” dialog select the file type and one or several device description files.

3. Click **[Open]**.

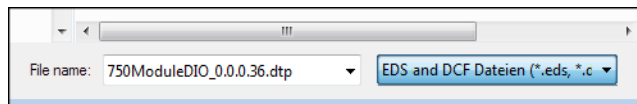


Figure 232: Selecting a Device Description

This will import the description files. The devices are available in the Product Catalog.

9.6.2.2 Deleting Devices

Please note that when deleting a device description file for a particular device, it may be required for several similar devices. You may therefore be removing several devices from your project.

1. In the “Product Catalog” page in Backstage view select the device that you wish to remove.

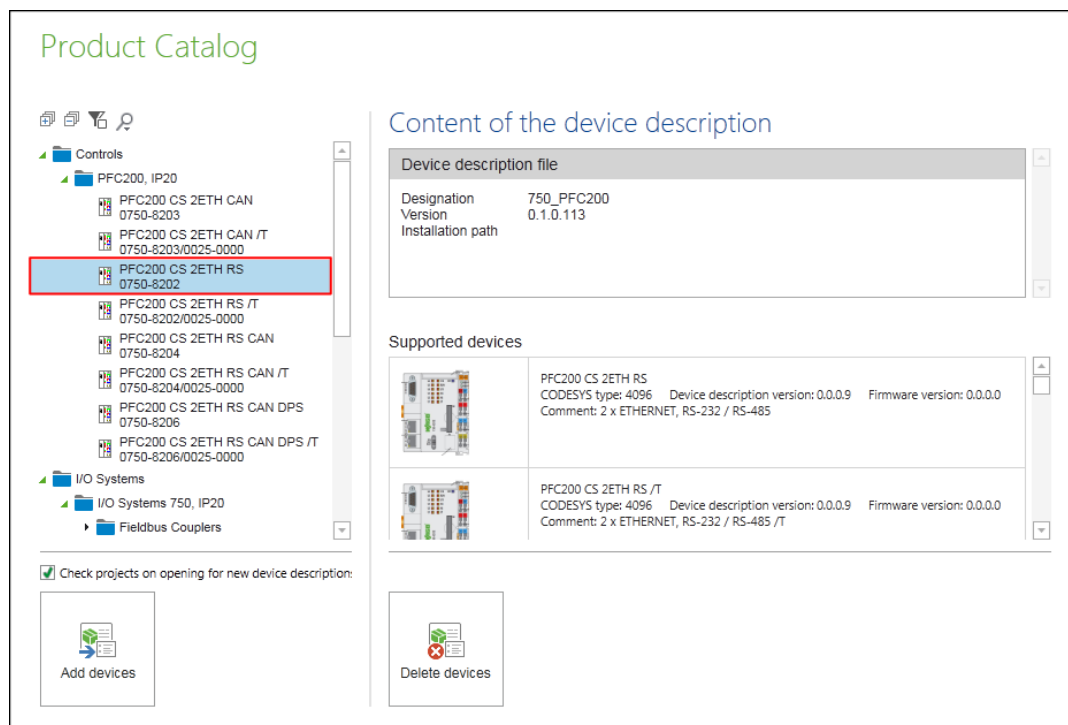


Figure 233: Selecting a Device

Information on the relevant device description file and a list of devices that belong to this description file and version are displayed on the right.

2. Click the **[Delete device...]** button.

The “Included Devices” area highlights in yellow the devices related to this device description file that will also be deleted.

3. Click **[Ok]** to actually delete the device types.

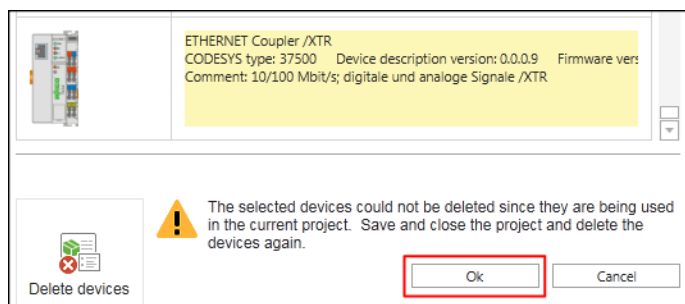


Figure 234: Deleting a Device Type

4. If the currently opened project contains devices that are affected by this deletion, save and close the project before deletion and repeat the step.

9.6.2.3 Displaying Device Type Versions/Device Descriptions of a Device

A device may be present in several device type versions, i.e., may be represented in different versions of device description files. Any of these versions can be used in a project. Versions of device descriptions are displayed on the “Product Catalog” page in the Backstage view and in the Product Catalog.

1. To display available versions of a device, first select a device or module in the Network or Device Detail view.
2. Click the [**Version information**] button in the “DEVICE” tab in the menu ribbon.

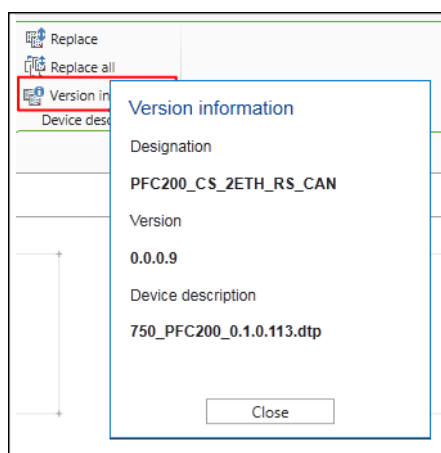


Figure 235: Displaying Device Type Versions/Device Descriptions

9.6.2.4 Showing Device Type Versions/Device Descriptions of Several Devices

1. To display the available versions of several or all devices, enable the **Display the Available Versions** in the Product Catalog so that a green tick is displayed.
This activation does not apply to all tabs of the Product Catalog, therefore switch the function on or off according to the tab.

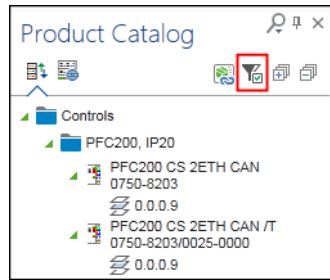


Figure 236: Displaying Available Versions

Other levels are displayed with the available device type versions in the tree structure of the Product Catalog.

When using the Search function, only the versions of devices are displayed that are present in the Search result.

9.6.2.5 Replacing Device Type Versions/Device Descriptions

Devices are integrated in *e!COCKPIT* via device descriptions. Every device description has a version. Different versions can be installed for a device, although versions can also be replaced by others. When returning to a previous version, a check for any functional restrictions is made via the DTP description. Incompatible versions are not displayed. Replacing is not possible if the device is connected online.

1. Double-click the required version of the device description to operate the used device in this version, for example, to maintain compatibility with other devices.

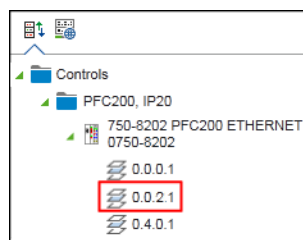


Figure 237: Replacing Device Type Versions/Device Descriptions

Alternatively, select a device in the Network or Device Detail view and then click the **[Replace]** button.

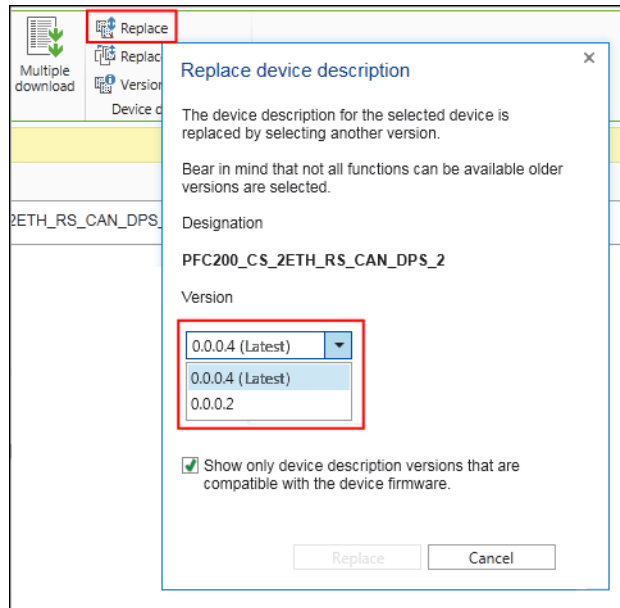


Figure 238: Replacing Device Type Versions/Device Descriptions

2. In the selection field, choose the required version and click [**Replace**].
3. To replace the device descriptions of all Devices used in the project click [**Replace all**].

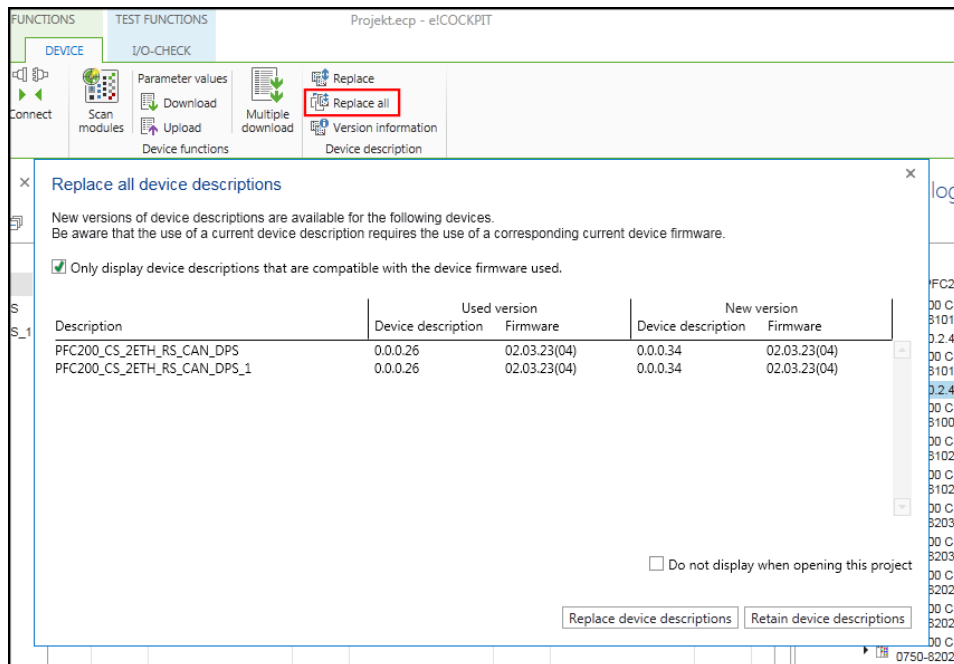


Abbildung 239: Replace All Device Descriptions

4. Replace the device description as described in section “Opening the Project with an Obsolete Device Description”.

9.6.2.6 Replacing Older Versions Automatically

1. Open the “Product Catalog” page in the Backstage view.
2. To search for new device descriptions when opening a project and to replace older versions automatically, tick the “Check projects on opening for new device descriptions” checkbox.

Older versions are only replaced if the new versions are compatible with the existing data and settings.

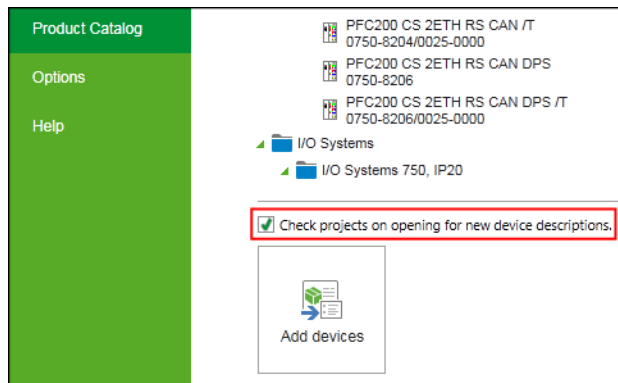


Figure 240: Replacing Older Versions Automatically

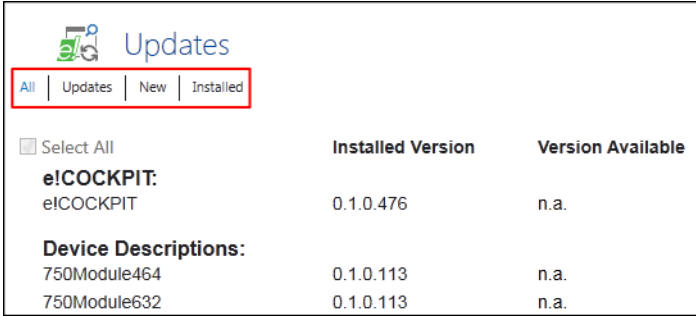
9.6.3 Performing Updates

Software updates can be started both automatically and manually on demand. These contain updates for *e!COCKPIT* itself, device description files for new WAGO devices and communication drivers.

1. Open Backstage view (“FILE” tab).
2. Click [**Help**] in the navigation bar.
3. Click [**Updates**].

The right-hand page shows information on installed components. When checking for updates, found updates and new components available are identified and also displayed in this area.

4. Use the “All,” “Updates,” “New” or “Installed” tabs to modify the display as required.



	Installed Version	Version Available
e!COCKPIT:		
e!COCKPIT	0.1.0.476	n.a.
Device Descriptions:		
750Module464	0.1.0.113	n.a.
750Module632	0.1.0.113	n.a.

Figure 241: Performing Updates

Searching for updates automatically:

5. Tick the “Check for updates and license information on starting *e!COCKPIT*” checkbox (Internet connection required during program start).



Note

Automatic restart after update!

Updates are activated after the next restart. If you are installing *e!COCKPIT* updates or new communication drivers, the software is restarted automatically. A restart is not required for importing new device descriptions.

Searching for updates manually:

6. To search for updates manually, click the **[Search for Updates]** button.

The updates found are then displayed in the categories *e!COCKPIT*, device descriptions and communication drivers, and highlighted in color. All updates are selected by default.

7. If required, exclude the individual updates by clicking on the relevant checkboxes and deactivating these updates.

The update installation process checks your license information with WAGO Kontakttechnik GmbH & Co. KG.

8. Agree to this check by activating the “I accept the WAGO SOFTWARE LICENSE AGREEMENT” checkbox.
9. To update all the selected components click **[Install updates]**.

9.6.4 Archiving

Saves the current project with all its associated data, such as devices used, device descriptions, libraries on the controller as an archive file. The archive combines project-related files so that they can be forwarded to third parties. An archive combines all the files contained and referenced in the currently opened project into a central “*.eca” archive file. The “Archive” function can be selected when a project is active.

9.6.4.1 Creating an Archive

1. Open Backstage view (“FILE” tab).
2. Click [**Import/Export**].
3. Click [**e!COCKPIT archive**].
4. Click the [**Save archive**] button.

A dialog will open, which displays the elements of the currently opened project. The archiving always includes all devices and applications. Libraries can be excluded from archiving by unticking the checkbox for the appropriate library.

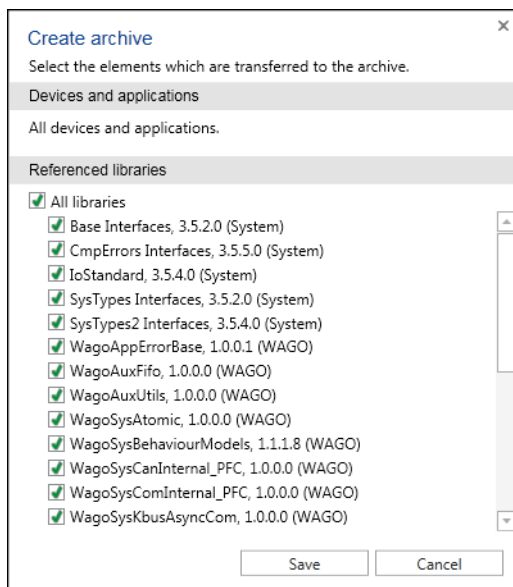


Figure 242: Saving Archives

5. Click [**Save**].
6. Select a location for the archive.
7. Click [**Save**] to finish the archiving.

9.6.4.2 Opening an Existing Archive

1. Open Backstage view (“FILE” tab).
2. Click [**Import/Export**].
3. Click [**e!COCKPIT archive**].
4. Click the [**Open archive**] button.
5. Select a saved “.eca” archive file.
6. Click [**Open**].

The archive file is extracted. Libraries are installed and loaded. The project is opened.

9.6.5 Deinstalling

Back up your project files before deinstalling in case you wish to reinstall *e!COCKPIT* in future. For this use the archive function (see Section “Operation” > “Setting and Managing” > “Archiving”).

1. Call up the Control Panel.
2. Click [**Uninstall Programs**] (such as with Windows 7).
3. Select “WAGO *e!COCKPIT*”.
4. Click [**Uninstall**].

The software with all integrated components is removed from your PC. Separately installed CODESYS versions are not affected by this.

9.7 Interfaces

9.7.1 *smart*DESIGNER

The *smart*DESIGNER from WAGO Kontakttechnik GmbH & Co. KG is an online configurator which enables terminal blocks, I/O modules, multi-terminal connectors, PCB terminal blocks and WINSTA[®] cables to be designed and projects to be validated. Labels can be created and printed for the components used. The graphical display of components on a DIN rail is the same as in *e!COCKPIT*. The *smart*DESIGNER can also be used to generate mounting plans, parts lists and 3D views. The tool features several CAE interfaces including EPLAN, a system for designing electrical installations. This interface supports export/import to/from *e!COCKPIT*, so that modifications made in one system can be transferred to the other.

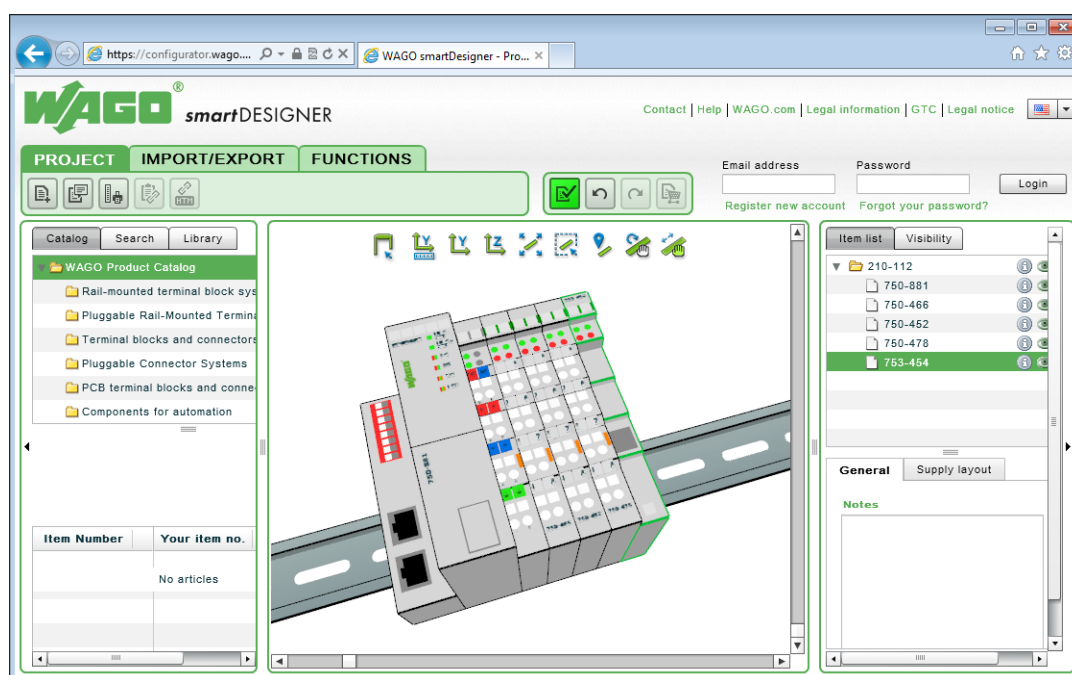


Figure 243: WAGO *smart*DESIGNER

9.7.1.1 Importing a Project from *smartDESIGNER*

1. To import projects created with *smartDESIGNER*, click [**Import/Export**] in the Backstage view.
2. Click [*smartDESIGNER*].
3. Click [**Import**].
4. Select a valid file (CAE file, *.xml) from the dialog that opens.

The content of the file is opened as a new project. Network view is opened automatically.

If a project was opened and modified beforehand, you will be asked to save it.

The “Import” dialog is opened. This contains all imported (green) and unimported (red) devices.

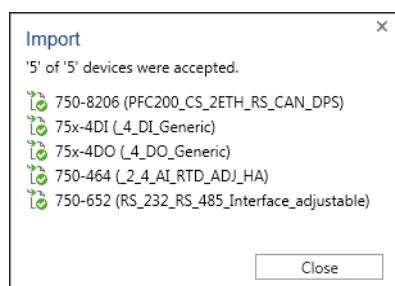


Figure 244: Overview of the Imported Devices

9.7.1.2 Exporting a Project for *smartDESIGNER*

1. To export projects you wish to open in *smartDESIGNER*, click [**Import/Export**] in Backstage view.
2. Click [*smartDESIGNER*].
3. Click [**Export**].

Select a valid file in the dialog.

All devices with modules are created in the export file.

The Backstage view stays open.

A warning message appears if the project contains a device for which a DTP is not installed. If this is a coupler, the modules are exported on a DIN rail without this coupler.

A warning message also appears if the project contains a device that does not have a valid order number, as this is possibly a generic item.

List of Figures

Figure 1: Display of the Start Screen during Loading.....	18
Figure 2: Display of the License Status.....	18
Figure 3: Entering the License Key.....	19
Figure 4: Entering Licenses.....	20
Figure 5: Activating Automatic Updates.....	21
Figure 6: Start View	22
Figure 7: Main View	24
Figure 8: Quick Access Toolbar (Standard View)	25
Figure 9: Customizing the Quick Access Toolbar	25
Figure 10: Menu Ribbon (Standard View).....	26
Figure 11: Managing workspaces.....	27
Figure 12: Creating a New Workspace	28
Figure 13: “VIEW” Tab	30
Figure 14: “NETWORK” Tab.....	32
Figure 15: “DEVICE” Tab	34
Figure 16: “Multiple download” Dialog	35
Figure 17: “PROGRAM” Tab.....	36
Figure 18: Multiple Download.....	38
Figure 19: “Search” Dialog	39
Figure 20: “DEBUG” Tab.....	42
Figure 21: “I/O CHECK” Tab.....	46
Figure 22: Views and Workspaces	49
Figure 23: General Status Display for Connections.....	50
Figure 24: Status Display for Devices (Online)	52
Figure 25: “Network/Devices” Workspace.....	53
Figure 26: Network View.....	54
Figure 27: Context Menu of a Device	55
Figure 28: Color Coding of Connectors	57
Figure 29: Context Menu of a Connector.....	57
Figure 30: Connection between Two Devices	59
Figure 31: Fieldbus-Specific Configurators	60
Figure 32: “MODBUS Master Variables” Tab	61
Figure 33: Slave Variables Context Menu	62
Figure 34: Master Variables Context Menu	63
Figure 35: “Connection Settings” Tab	63
Figure 36: MODBUS Slave Configurator.....	65
Figure 37: “CANopen Remote Device” Tab (view with “Expert Settings” checkbox enabled)	66
Figure 38: SD Channel Configuration.....	67
Figure 39: Heartbeat Consuming Properties	68
Figure 40: “PDO Mapping Tab”	69
Figure 41: “Service Data Object Tab”	70
Figure 42: Adding Service Data Objects.....	71
Figure 43: “CANopen I/O Mapping” Tab.....	72
Figure 44: CANopen Slave Configuration.....	73
Figure 45: Edit I/O Area.....	73
Figure 46: Add I/O Range	74
Figure 47: Edit Parameter Range	75

Figure 48: CANbus Slave I/O Mapping.....	76
Figure 49: CANopen Manager	77
Figure 50: Check and Fix Configuration.....	77
Figure 51: CANopen I/O Mapping	79
Figure 52: PROFIBUS Connection Configurator	80
Figure 53: Context Menu of a Data Point	82
Figure 54: Information in the Status Line of the PROFIBUS Configurator	82
Figure 55: PROFIBUS Connection Configurator	82
Figure 56: Device View (Example: Fieldbus Controller with Connected Modules)	84
Figure 57: I/O Module Context Menu.....	85
Figure 58: I/O Mapping.....	87
Figure 59: Elements of the Program Editor.....	90
Figure 60: Library Manager	92
Figure 61: Task Configuration	93
Figure 62: “PLC_Task” Main Task.....	94
Figure 63: Selecting a Program via the Input Assistant	96
Figure 64: Breakpoints	97
Figure 65: Call Stack	97
Figure 66: Cross Reference List.....	98
Figure 67: Creating and Managing User-Defined Workspaces	99
Figure 68: Panels	100
Figure 69: Displaying Panels	100
Figure 70: “Device Structure” Panel	101
Figure 71: “Program Structure” Panel.....	102
Figure 72: Context Menu of an Application	103
Figure 73: “Product Catalog” Panel	105
Figure 74: Displaying Available Versions	106
Figure 75: Dragging Devices to the Network View	106
Figure 76: Displaying Product Information	106
Figure 77: Adding a Device to a Project	107
Figure 78: “Settings” Panel with the Button to a Device Configurator (CANopen).....	108
Figure 79: Additional Plc Settings	109
Figure 80: Message Bar.....	110
Figure 81: Message Window.....	110
Figure 82: Status Bar	111
Figure 83: Backstage View	112
Figure 84: Backstage View, “New” Page	113
Figure 85: Backstage View, “Open” Page	114
Figure 86: Backstage View, “Save As” Page.....	115
Figure 87: Backstage View, “Print” Page	116
Figure 88: Backstage View, “Import/Export” Page	117
Figure 89: Backstage View, “Project Settings” Page.....	119
Figure 90: Backstage View, “Product Catalog” Page	120
Figure 91: Backstage View, “Options” Page	122
Figure 92: Backstage View, “Help” Page	124
Figure 93: Using Online Help	126
Figure 94: Displaying Panels	128
Figure 95: Closing a Panel	128

Figure 96: Docking Panels	129
Figure 97: Grouping Panels.....	129
Figure 98: Docking and Floating the Main View.....	130
Figure 99: Multiple Selection in Tree Structures or Network View	131
Figure 100: Multiple Selection with Frames	131
Figure 101: Moving Multiple Selection	132
Figure 102: Predefined Workspaces.....	133
Figure 103: Opening the “Network/Devices” Workspace	133
Figure 104: Opening the “Network/Devices” Workspace	133
Figure 105: Opening Network View	134
Figure 106: “NETWORK” Tab.....	134
Figure 107: Opening Device Detail View via Network View	134
Figure 108: Opening Device Detail View via the “DEVICE” Tab.....	134
Figure 109: Opening Device Detail View	135
Figure 110: “DEVICE” Tab	135
Figure 111: Use the Breadcrumb navigation to move to the Network view	135
Figure 112: Use the “NETWORK” tab to move to the Network view	135
Figure 113: Moving between Network View and Device Detail View	136
Figure 114: Toggling between Graphical and Tabular View.....	136
Figure 115: Opening the “Programming” Workspace	136
Figure 116: Opening the “Programming” Workspace	136
Figure 117: Opening the “Programming” Workspace	137
Figure 118: Opening the “PROGRAM” and “DEBUG” Tabs.....	137
Figure 119: Creating a New Workspace	137
Figure 120: New Button in the Menu Ribbon.....	138
Figure 121: Selecting a Template.....	142
Figure 122: Docking the Position of the Project File.....	143
Figure 3: Replacing Device Descriptions.....	143
Figure 10: Replacing All Device Descriptions.....	144
Figure 123: Saving a Project Automatically	145
Figure 124: Automatically Saved Projects on the “Open” Page.....	146
Figure 125: Dragging Devices to the Network View.....	147
Figure 126: Adding Modules	148
Figure 127: Selecting the Medium for Scanning.....	149
Figure 128: Restricting the Scan Range	149
Figure 129: Starting the Scan Operation	149
Figure 130: Displaying a Scan Result	150
Figure 131: Opening Graphical Network View	150
Figure 132: Scanning Modules.....	151
Figure 133: Displaying Scanned Modules	151
Figure 134: Selecting Modules.....	151
Figure 135: Accepting Modules with the Same Scan Result Together.....	152
Figure 136: Applying Modules	152
Figure 137: Displaying Scanned Modules	152
Figure 138: Display of Applied Modules.....	153
Figure 139: Scanning for Modules of Several Devices.....	153
Figure 140: Displaying the Scan Report	154
Figure 141: Connecting Devices (Network View).....	154
Figure 142: Connecting Devices (Device Structure)	154
Figure 143: Connecting Devices (Menu Ribbon)	155

Figure 144: Opening the “Settings” Panel (here: Device Detail View)	155
Figure 145: “Settings” Panel (Tab Navigation)	156
Figure 146: “Settings” Panel (Groups in Tabs).....	156
Figure 147: Opening Device-Specific Configurators.....	157
Figure 148: Writing Parameters to the Device	158
Figure 149: Reading Parameters from the Device	158
Figure 150: Dragging Devices to the Network View	159
Figure 151: Opening the “Settings” Panel	159
Figure 152: Establishing Network Connections.....	160
Figure 153: Opening the Interface.....	160
Figure 154: Selecting a Connection	161
Figure 155: Displaying Devices That Are Logically Connected	161
Figure 156: Opening Connection Settings or MODBUS Slave Configurator	162
Figure 157: Variable Configuration MODBUS, View for the Master (Client) ..	162
Figure 158: Creating New Variables on the Slave	163
Figure 159: New Variable in the Slave Application	163
Figure 160: MODBUS Connection Settings	164
Figure 161: New Variable on the Master	164
Figure 162: Opening the Main Program of the Master	164
Figure 163: Accessing the “newVar” Variable in the Master Application	165
Figure 164: Dragging Devices to the Network View	166
Figure 165: Assigning an IP Address.....	166
Figure 166: Opening the “Settings” Panel	167
Figure 167: Assigning the Role of the Device	167
Figure 168: Creating a Connection between CANopen Devices	167
Figure 169: Adapting “Node ID” and Baud Rate.....	168
Figure 170: Opening the CANopen Connection Configurator	168
Figure 171: CANopen Connection Configurator (Slave).....	168
Figure 172: Adding a Data Point.....	169
Figure 173: Updating the Configuration	169
Figure 174: Choosing the Update Behavior	170
Figure 175: Opening the CANopen Connection Configurator	170
Figure 176: Entering Variables	170
Figure 177: Placing the PROFIBUS Master and Slave in the Network View	171
Figure 178: Connecting PROFIBUS Devices	171
Figure 179: Opening the “Settings” Panel	172
Figure 180: Opening the Configurator of the PROFIBUS Connection	172
Figure 181: PROFIBUS Connection Configurator	173
Figure 182: Creating a New Variable.....	173
Figure 183: Configured Variables	174
Figure 184: Creating Multiple Variables.....	174
Figure 185: Using Created Variables	175
Figure 186: Opening the “I/O Mapping” Tab	175
Figure 187: Manual Entry of the IEC Variable.....	175
Figure 188: Entering IEC Variables via the Input Assistant.....	175
Figure 189: Selecting a Variable	176
Figure 190: Mapping Completed	176
Figure 191: Saving the Device Description File	176
Figure 192: Switching an LED On/Off with the Mouse	177
Figure 193: Display of an I/O Module for the Input/Output Test.....	178

Figure 194: Display of a Node for the Input/Output Test	178
Figure 195: Performing a Test.....	179
Figure 196: Status Display for Inputs and Outputs	179
Figure 197: Changing Analog and Digital Values	179
Figure 198: Write Collection Values.....	180
Figure 199: Opening the “Programming” Workspace	181
Figure 200: Context Menu in the Program Structure	182
Figure 201: Adding a POU.....	183
Figure 202: Selecting a Template for Libraries.....	184
Figure 203: Creating a Library	184
Figure 204: Saving and Installing a Library.....	185
Figure 205: Adding the Library Manager	186
Figure 206: Creating a Program	187
Figure 207: Adding a Program Call	187
Figure 208: Entering the Program Name via the Input Assistant.....	188
Figure 209: Programming Editor (in this example: Structured Text (ST)).....	188
Figure 210: Entering Variables/Data Points.....	188
Figure 211: Entering Variables/Data Points via the Input Assistant.....	189
Figure 212: I/O Mapping.....	189
Figure 213: Compiling an Application.....	190
Figure 214: Message Window.....	190
Figure 215: Excluding the Program from Compiling	191
Figure 216: Program Excluded from Compiling.....	191
Figure 217: Selecting Values for Monitoring.....	192
Figure 218: Watch List.....	192
Figure 219: Adding a Visualization	195
Figure 220: Creating a Visualization.....	195
Figure 221: Assigning a Value	196
Figure 222: Multiple Download	198
Figure 223: Loading a CODESYS Project.....	199
Figure 224: Converting a Library Reference	199
Figure 225: Adding a Library.....	201
Figure 226: Exporting Program Elements.....	202
Figure 227: Importing Program Elements.....	203
Figure 228: Documenting a Project.....	204
Figure 229: Moving to the Product Catalog Management	205
Figure 230: Selecting a Device Description.....	206
Figure 231: Selecting a Device	206
Figure 232: Deleting a Device Type	207
Figure 233: Displaying Device Type Versions/Device Descriptions	207
Figure 234: Displaying Available Versions	208
Figure 235: Replacing Device Type Versions/Device Descriptions.....	208
Figure 236: Replacing Device Type Versions/Device Descriptions.....	209
Abbildung 239: Replace All Device Descriptions	209
Figure 237: Replacing Older Versions Automatically	210
Figure 238: Performing Updates	211
Figure 239: Saving Archives	213
Figure 240: WAGO <i>smart</i> DESIGNER	216
Figure 241: Overview of the Imported Devices	217

List of Tables

Table 1: Number Notation.....	10
Table 2: Font Conventions	10
Table 3: System Requirements.....	15
Table 4: Key for Figure “Start View”	23
Table 5: Key for Figure “Main View”	24
Table 6: Buttons on the Quick Access Toolbar.....	25
Table 7: “START” Tab Operating Elements	26
Table 8: “VIEW” Tab Operating Elements.....	30
Table 9: “NETWORK” Tab Operating Elements	32
Table 10: “DEVICE” Tab Operating Elements	34
Table 11: “PROGRAM” Tab Operating Elements	36
Table 12: “DEBUG” Operating Elements.....	42
Table 13: “I/O CHECK” Tab Operating Elements	46
Table 14: General Operating Elements	50
Table 15: Key for Figure “General Status Display for Connections”	50
Table 16: General Status Display for Connections	51
Table 17: Elements of the Network View	53
Table 18: Key for Figure “Network View”	54
Table 19: Device Status.....	55
Table 20: Context Menu of a Device	56
Table 21: Color Coding of Connectors	57
Table 22: Operating Elements in the Context Menu of a Connector	58
Table 23: Key for Figure “MODBUS Master Variables” Tab.....	62
Table 24: Key to Figure “Slave Variables Context Menu”	63
Table 25: Key to Figure “Master Variables Context Menu”	63
Table 26: Key to “Connection Settings” Tab.....	64
Table 27: Key for Figure “CANopen Remote Device Tab”	67
Table 28: Key for Figure “PDO Mapping Tab”	70
Table 29: Key to Figure “Service Data Object Tab”	70
Table 30: Key to Figure “CANopen I/O Mapping” Tab.....	72
Table 31: General Configuration.....	73
Table 32: CANbus I/O Mapping	76
Table 33: CANopen Remote Device.....	77
Table 34: PROFIBUS Connection Configurator.....	81
Table 35: PROFIBUS Configurator	83
Table 36: Elements of the Device View.....	85
Table 37: Context Menu of a Device	86
Table 38: I/O Mapping	87
Table 39: Elements of the Program Editor	90
Table 40: Library Manager.....	92
Table 41: “PLC_Task” Main Task.....	94
Table 42: Context Menu of an Application.....	104
Table 43: Buttons of the “Product Catalog” Panel.....	105
Table 44: Message Bar Elements	110
Table 45: Elements of the Message Window	110
Table 46: Legend for Figure Backstage View.....	112
Table 47: Backstage View Operating Elements, “Open” Page.....	114
Table 48: Backstage View Operating Elements, “Print” Page.....	116

Table 49: Backstage View Operating Elements, “Import/Export” Page.....	118
Table 50: Backstage View Operating Elements, “Project Settings” Page	119
Table 51: Backstage View Operating Elements, “Product Catalog” Page.....	120
Table 52: Backstage View Operating Elements, “Options” Page.....	122
Table 53: Backstage View Operating Elements, “Help” Page.....	124
Table 54: General Key Combinations	140
Table 55: Key Combinations, “START” Tab	140
Table 56: Key Combinations, “PROGRAM” Tab	140
Table 57: Key Combinations, “PROGRAM” Tab	141

WE! INNOVATE!

WAGO Kontakttechnik GmbH & Co. KG
Postfach 2880 • D-32385 Minden
Hansastraße 27 • D-32423 Minden
Phone: 05 71/8 87 – 0
Fax: 05 71/8 87 – 1 69
E-Mail: info@wago.com
Internet: <http://www.wago.com>

