OPUS Projektor Manual

Welcome

1   Welcome

If you are new to this program, the best starting point is the [Quick Start Tutorial](#)[11].

New content for this version

[Heavily enhanced reporting functionality with issue view](#)[730]
[Image Pool project tree node](#)[54]
[A new event OnProjektInitFinished](#)[430]

Many useful changes like:
[setVariableValid JavaScript function](#)[559]
[String Fields](#)[504] now can have up to 65535 characters
[Containers](#)[466] can be pasted into containers (don't do it, though)
[Create JavaScript array from objects](#)[43]
new event [OnCancel](#)[435]
possibility to deactivate multi touch with [@MultiTouchEnabled](#)[174]

and much more...

Version check!

Starting with version 2020.4, a version check is implemented in the PClient. Only if the major and minor version of the used development version and PClient version are the same, the project will be loaded. Otherwise a red warning screen will be shown that informs of the version discrepancy.

Important disclaimer!

Please note that for i.MX6 devices, installing the OS 2.x will prohibit a downgrade back to OS 1.x!
Once an OS 2.x is installed, there is currently no way to downgrade back to OS 1.x!

Limitations in the demo version

The demo version is limited to the creation of up to 2 pages and up to 5 user-defined variables.
Additional pages or variables cannot be created.
If a project is opened that exceeds one of these limitations, the project can not be downloaded to the device.
Read [here](#)[63] how to get a license!

Known issues

- GIF files can only be used as Image to Display for Picture Graphic objects, but not as background, needle, border etc. images

- For devices with multi touch, buttons should not be put too close together, especially for contrary functionalities (up/down or plus/minus). With distances < 1.5 cm a second press can sometimes not be recognized as such but will instead be registered as a false coordinate for the first finger, resulting in an unwanted release for the first button.
- Properties of objects should not excessively be changed before the modifications were actually seen on the screen. This is especially relevant with i.MX 35 devices and string properties like Image to Display.

This manual has 141 chapters with ~162.000 words and 732 images.
Last change: 2022/10/07, revision 1268.
- new event OnCancel added
- Search Tips & Tricks added
- new and changed welcome project screens added
- possibility to bulk-delete CAN mappings added in user interface
- validation information for J1939 DM1/2 translation files added
- language project properties updated with new functionality
- adjusted max. object width / height to 2 * max. device resolution
- new chapter Image Pool in User Interface
- new event OnValidStateChange
- video mirror information corrected and example images added
- Several small improvements
- Rounding up of power management timeout variables added to documentation
- New events OnProjektInitFinished and OnPageInitFinished
- New reporting features added
- JS function setVariableValid added
- Heartbeat page added to Welcome Project (2.3.9) documentation
- FAQ about PDF reader added
- known problem added
- remark about camera frame size and position added
- new screenshots for Welcome Project 2.3.1
- audio options in events and actions corrected
- copy buttons added in key/ encoder color page in Welcome Project
- New pages low power times and alarm logging in Welcome Project
- Error -93 in Updater FAQ for Ethernet transfer
- variable list updated
- multi touch added to known issues
- FAQ about screenshots added
- constraints of using GIF files added, incl. FAQ and known issues
- Updater CLI documentation added
- tradeshow dashboard added
- New screenshot for new themes

- Introduced text variables for pre-defined variables chapter device descriptions
- Update of Ethernet FAQ
- Updated some screenshots for updater settings in welcome project
- New screenshot for DayNight
- OS/PClient update with partition in update manual
- New splash for pdf manual
- Icon reference in welcome project added
- New chapter for project search[108]
- New chapter for versioning system[123]
- New chapter for undo-redo[142]
- New page multi language in Welcome Project
- Added font FAQ
- Added documentation for rsc JS command
- Corrections in variable logging chapter
- Touch Input property for list objects documentation added
- list object documentation improved
- runSystemCommand function added
- documentation for openPDFReader JS function added
- EEPROM page added to welcome project documentation
- CANopen PDO page added to welcome project documentation
- info about rtsp availability
- updated tour link
- known issues added on welcome page

Imprint
Privacy policy

Quick Start Tutorial

2 Quick Start Tutorial

[PLAY BUTTON] Tour of the program
https://www.youtube.com/playlist?list=PLhLYlu1YL064ENX6BsFDBaUDtaBr6TD4L

This quick start tutorial demonstrates the basic steps to create a display project. Start with [creating a project]^{□11}, then continue with [designing the page content]^{□13}, then [simulate the project]^{□28}..

2.1 Creating a Project

To create a new project, click on the **New Project** button in the toolbar (1) or use the menu File -> New Project...
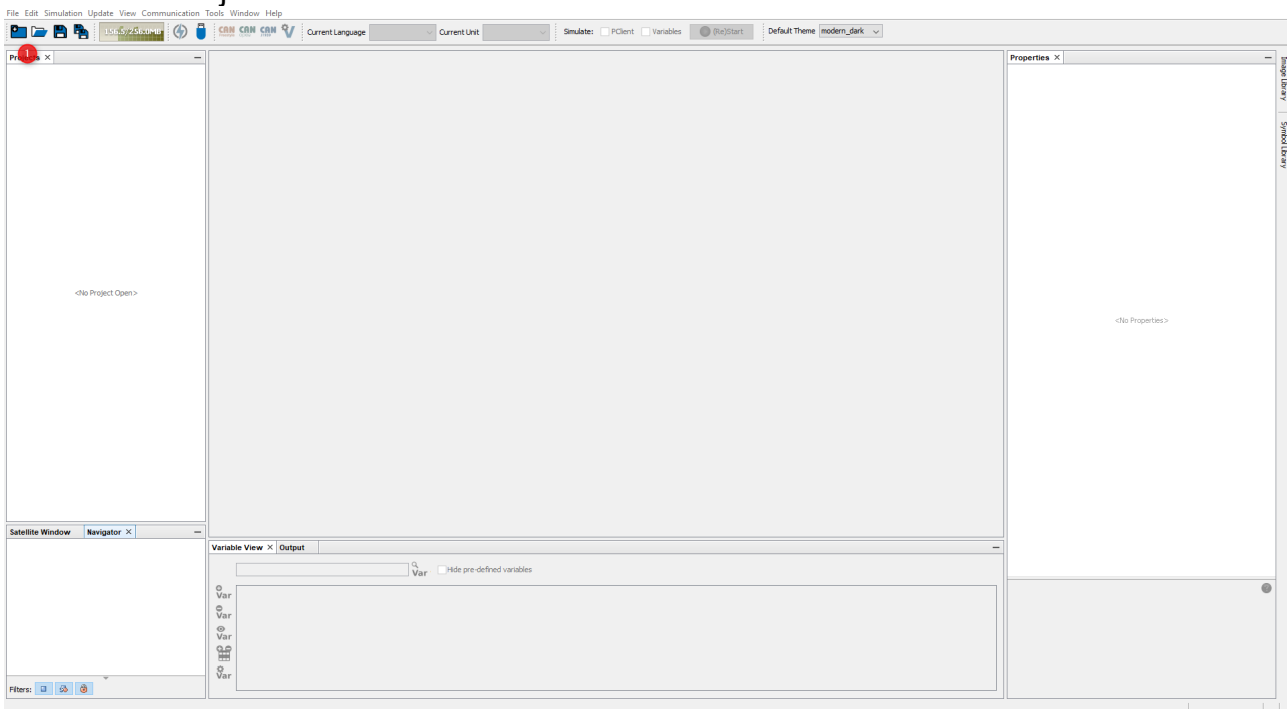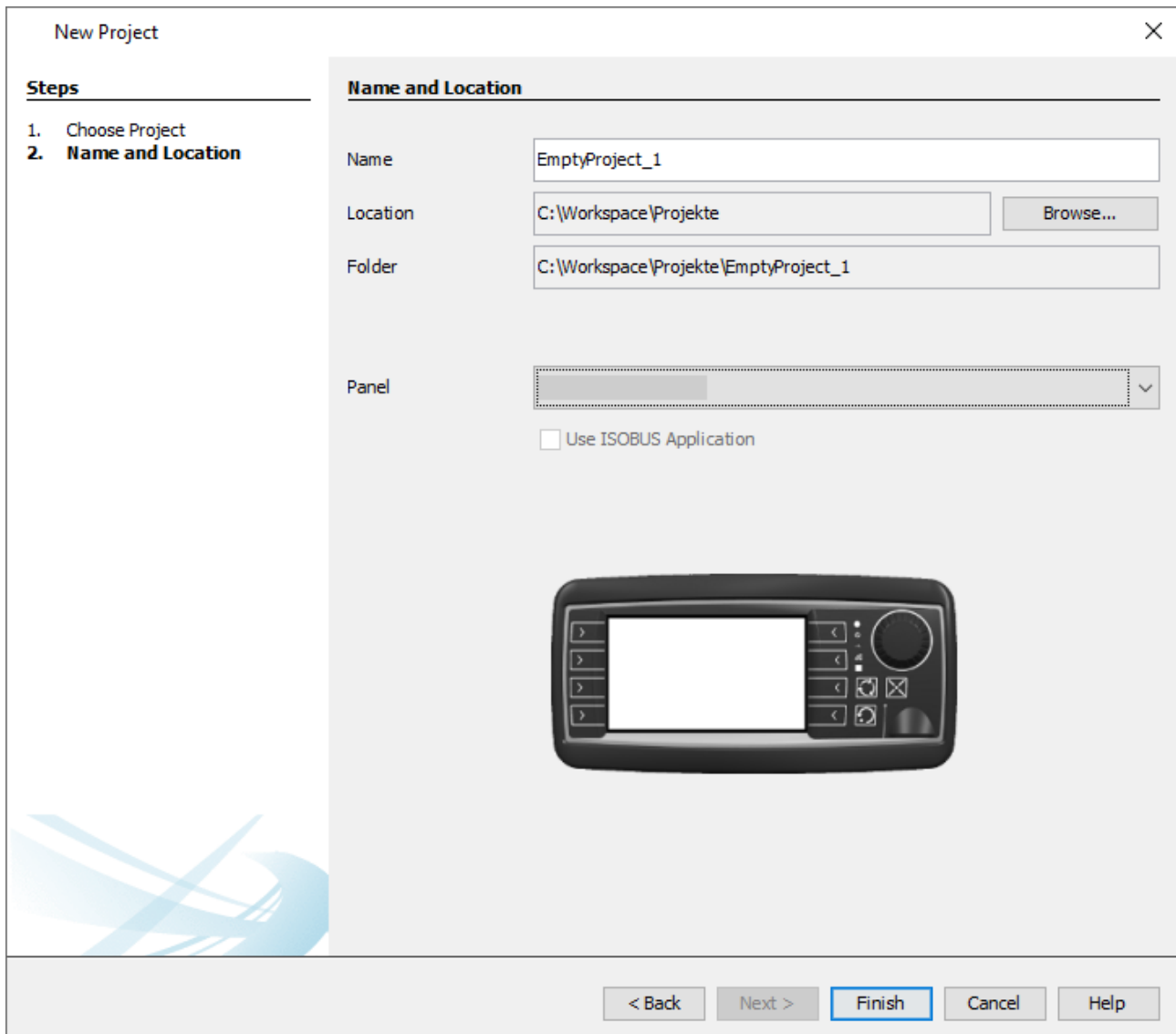
The **New Project** window opens. Stay on the General category and click the **Next** button (2) After you finished the tutorial, you are invited to check out some of the sample projects that you can access in this dialog.
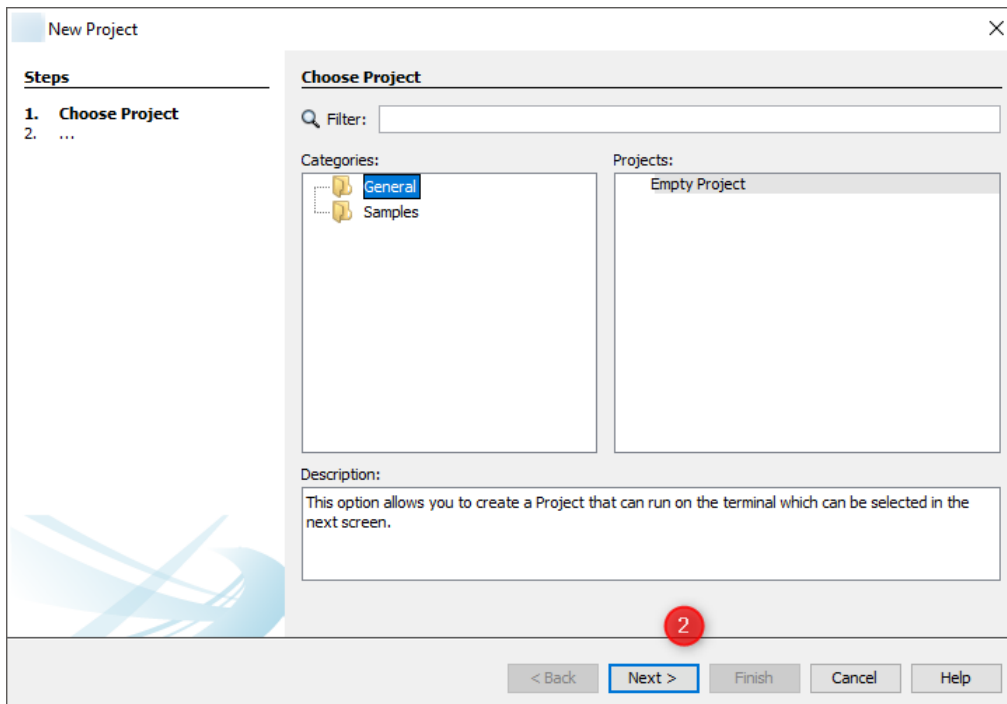
In the next step you have to provide the following information:
- The project name
- The directory where the project will be stored
- The display device and orientation the project shall run on (for the tutorial, leave the display that is selected by default)
- Click **Finish** button to complete the project creation.

New Project ✕

Steps

Name and Location

1. Choose Project
2. **Name and Location**

Name: EmptyProject_1

Location: C:\Workspace\Projekte   Browse...

Folder: C:\Workspace\Projekte\EmptyProject_1

Panel: [▼]

☐ Use ISOBUS Application

< Back | Next > | Finish | Cancel | Help

Continue with designing page content [13].

For certain device types the option **Use ISOBUS Application** can be activated. Read the chapter ISOBUS Application [576] for more information.

2.2 Designing Page Content

A new project always comes with a page having a default page name. By following the next steps the page name can be changed.

- Double click on project name (7)
- Double click on "Pages" (7)
- Double click the **Home Page** in the project tree. (8)
- The **Home Page Editor** opens. (9)
- In the **Properties** window change the page name into **Gauges Page**. (10)
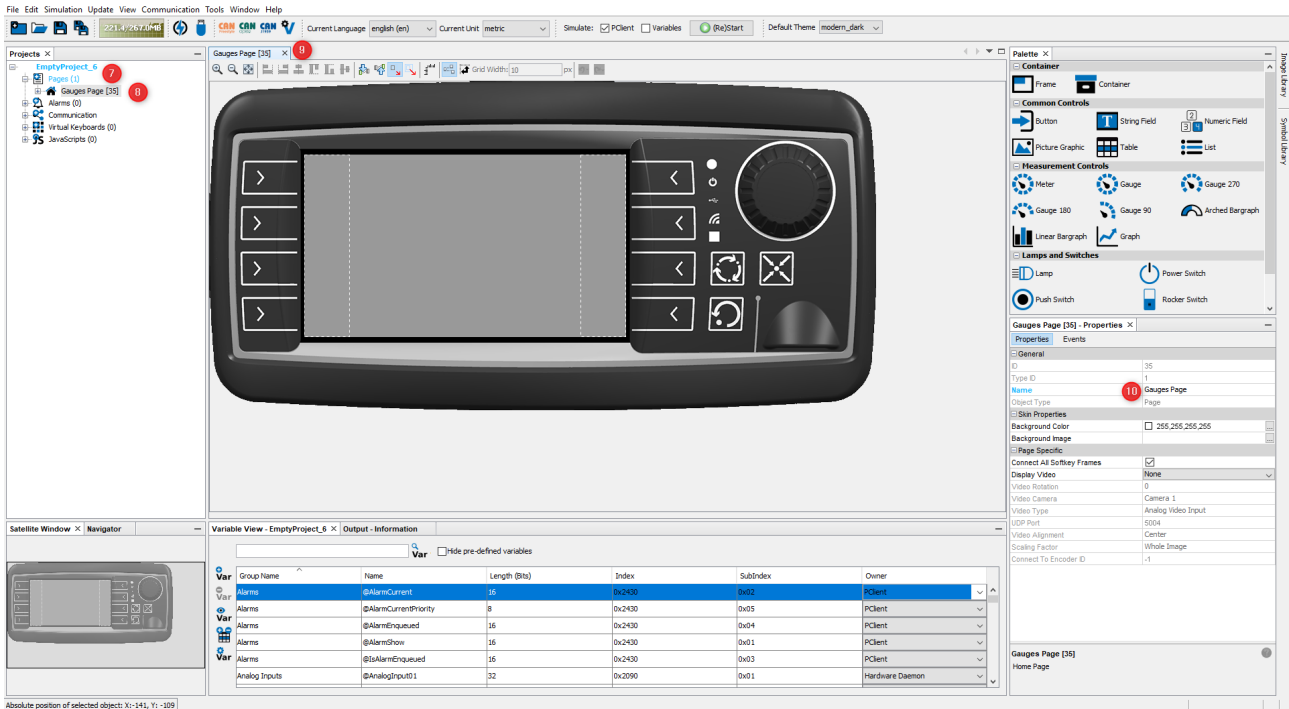
In this tutorial, the basic page layout is given by a background image. To add a background image, open the Image Library tab (11), and use the Browse button (12) to add the folder where the tutorial images were saved. Then select the image Training Background1.png (13) and drag & drop it onto the gray frame in the virtual device (14). Select the Frame 1 and then Background Image.
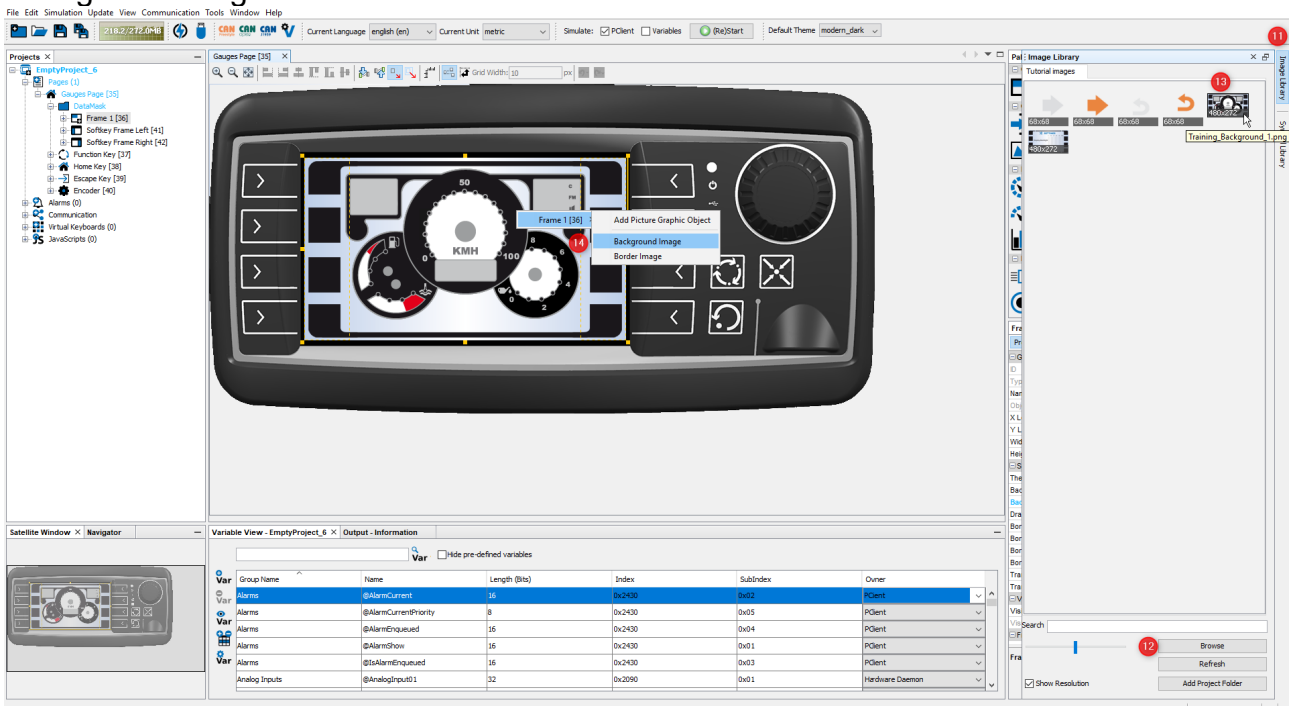
Note that the image now has a suffix _480_272 in the file name . A new instance of the image in this resolution is automatically created.

Visualization of Received Values

Adding a Gauges Needle

Now you're ready to add the **Meter** object.

- Drag the **Meter** object from the **Palette** (15) and drop it in the editor (16).

Now change the following properties of the **Meter** in the **Properties** window (for the min/max values use the exact order as listed here):

Make sure that the **Meter** object is selected in the scene!

- Select the **Theme Template** "modern_bright"
- **X Location**: 181
- **Y Location**: 51
- **Width**: 118
- **Height**: 118 (will be set automatically when you set the width)
- Check **Transparent**
- **Max Value**: 100
- **Absolute Max Value**: 100
- **Min Value**: 0
- **Absolute Min Value**: 0
- **Start Angle**: 325
- **End Angle**: 215
- Uncheck **Draw Ticks**

Here is what the result looks like:

Attaching a Variable to the Meter

To be able to change the value shown by the meter, you need to create a variable and attach it to the meter:

- Click the **Add New Variable** button (17) of the **Variable View**.
- A new variable with name **New Variable_1** is created.
- Change the Variable **Name** of **New Variable_1** into **kmh** (18).

To make it easier to work with variables you have created, there are two possibilities to reduce the amount of variables shown in the **Variable View**:

- One option is to check the **Hide pre-defined variables** check box (19).
- A second option is to enter a search sting (e.g. the variable name **kmh**) in the search area (20).
- In both cases, the amount of visible variables is reduced.

Now you need to attach the variable **kmh** to the meter:
- Click the **kmh** variable row to select it and drag & drop the variable over the meter. You can also drop the variable on the **Meter 1** in the project tree (22).
- A context menu appears. In the first level, select the Meter 1 object. In the second level, select the the **Variable Reference** option (21).
- You can now check in the **Properties** window, that the variable **kmh** is referenced by the meter object (23).
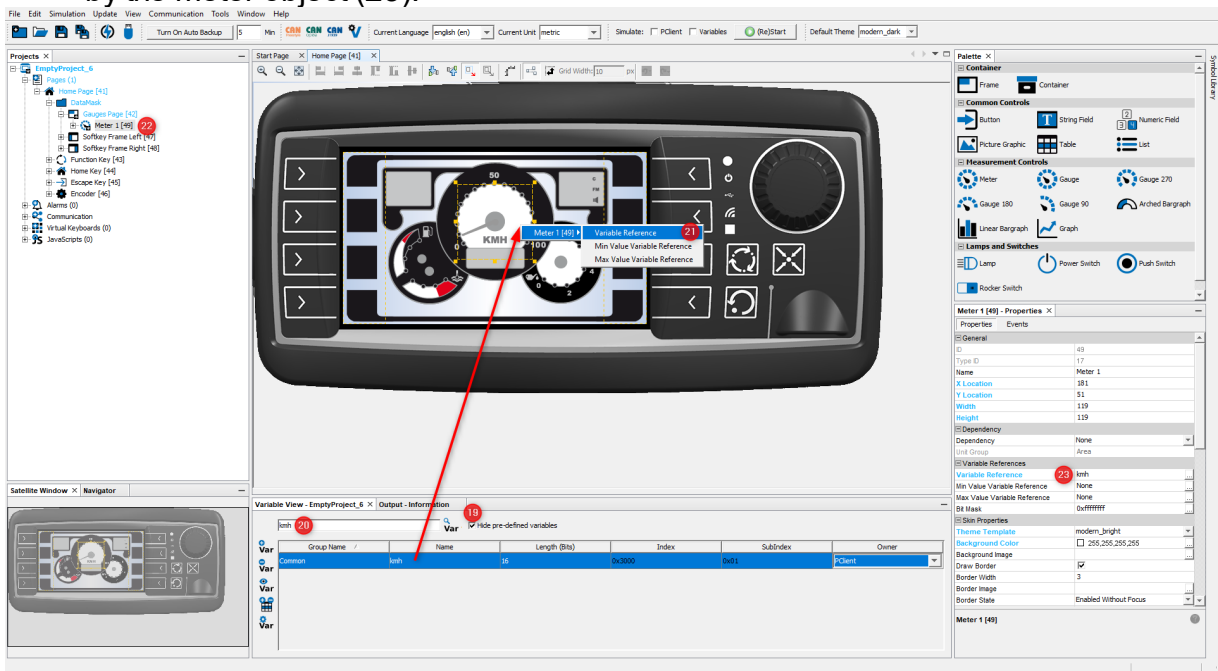
Later we will simulate the project. To be able to easily simulate the correct variable values, we have to set the minimum and maximum values of the **kmh** variable:
- Select the **kmh** variable in the **Variable View** (24).
- Open the right click context menu of the **kmh** variable and choose **Edit in Variable Manager** (25).
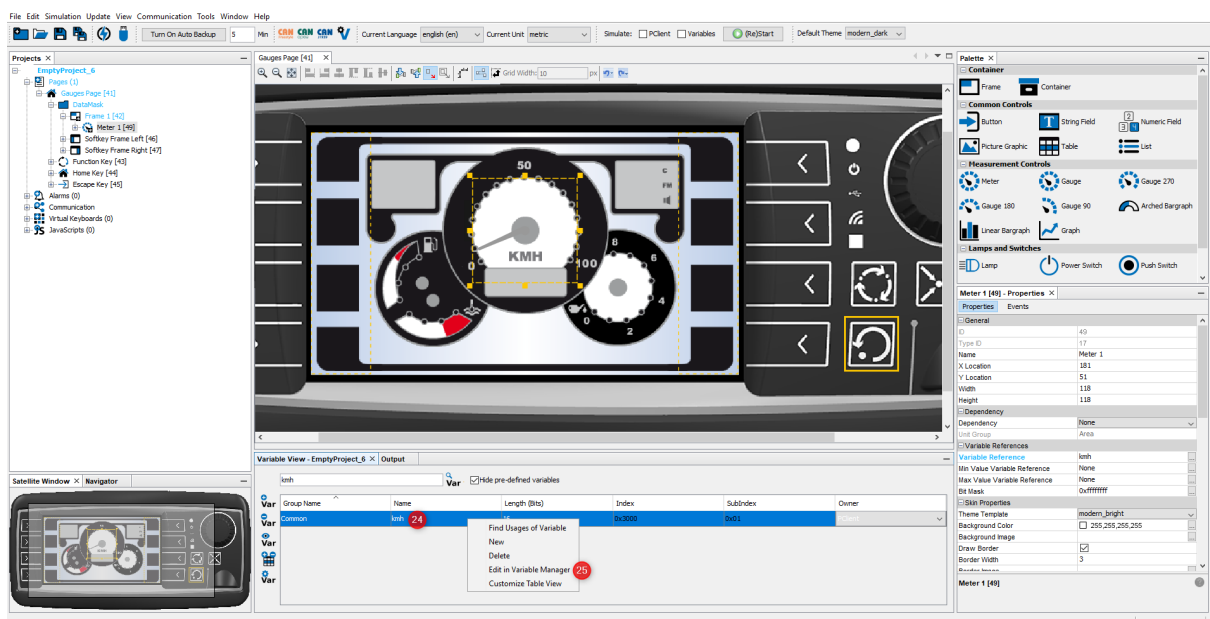
The **Variable Manager** window opens: Set the following properties:

- **Minimum Value**: 10
- **Maximum Value**: 170
- **Default Value**: 10

Then close the window by clicking the **OK** button (22).
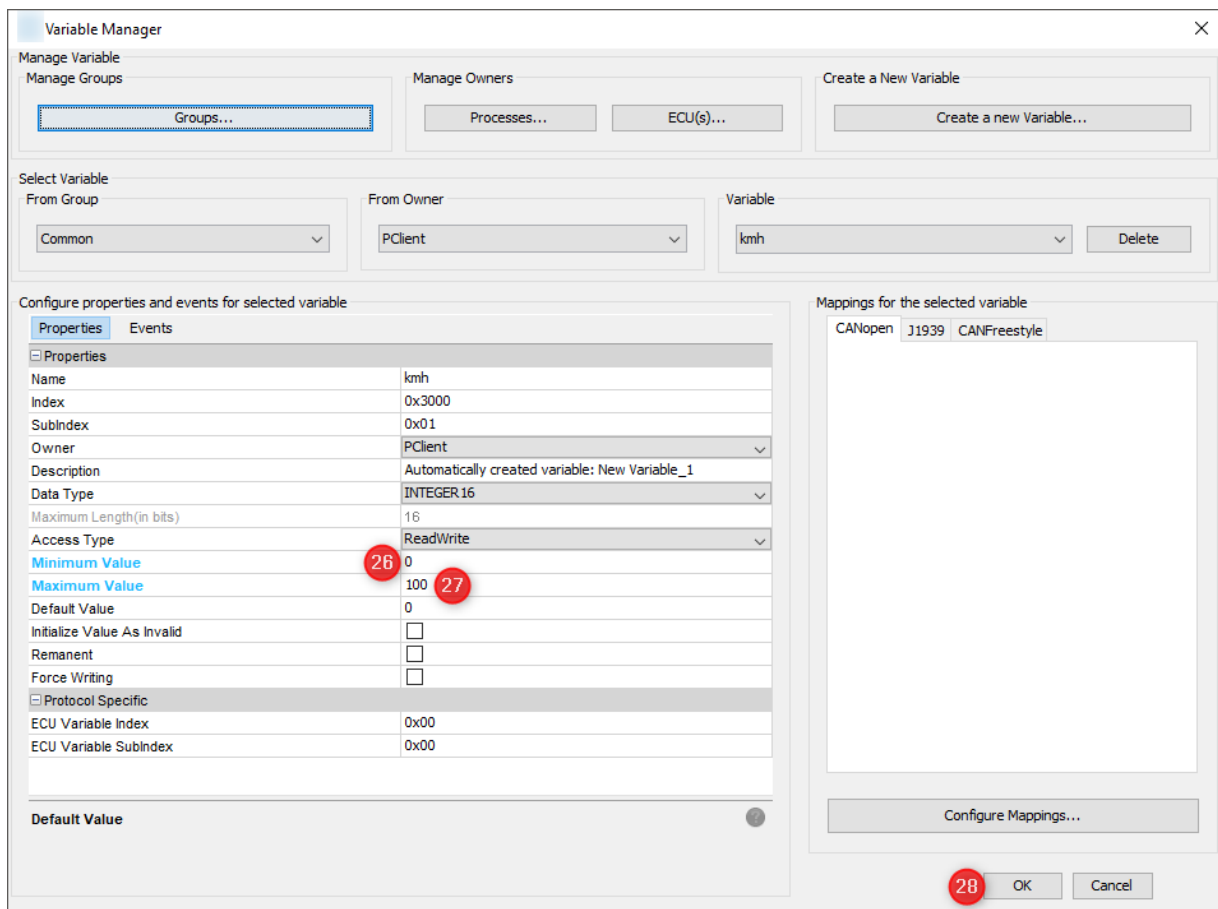
Configure How Variable KMH is Received via CAN

Setting up CAN communication start with the port configuration:
- Choose **Port Configuration...** from the menu **Communication** (29).

In the **Port Configuration** window:
- Select **CAN:1** (30).
- Set the Bus Speed to **250 kbps** (31).
- Close the window by clicking **OK** button (32).
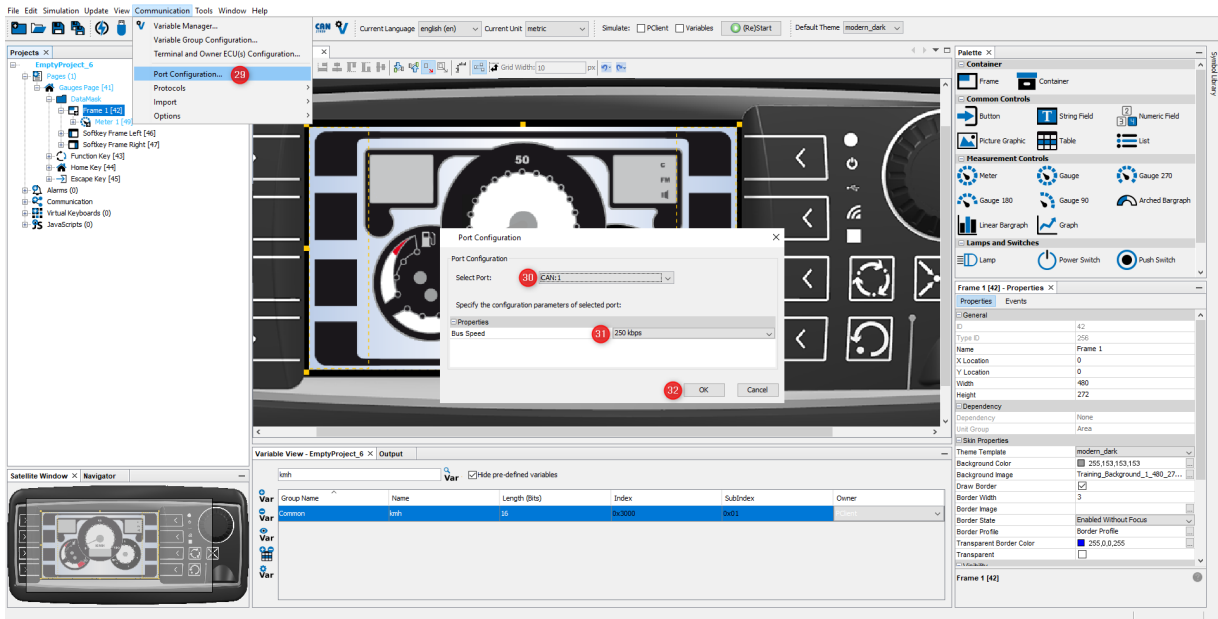
After configuring the CAN port, the CAN protocol and a communication partner, e.g. an ECU, must be defined:

- Click the **Terminal and Owner ECU(s) Configuration...** entry in the **Communication** menu (33).

The **Terminal and Owner ECU(s) Configuration...** dialog appears.

- Choose protocol **CANFreestyle running on CAN1** (34) and click the **Add** button (35).
- Choose **CANFreestyle running on CAN1** (36).
- Enter ECU Name **MyECU** (37) and click the **Add** button (38).
- Now the ECU is listed in the **Select ECU** dropdown menu (39).
- Close the window by clicking **OK** button (40).
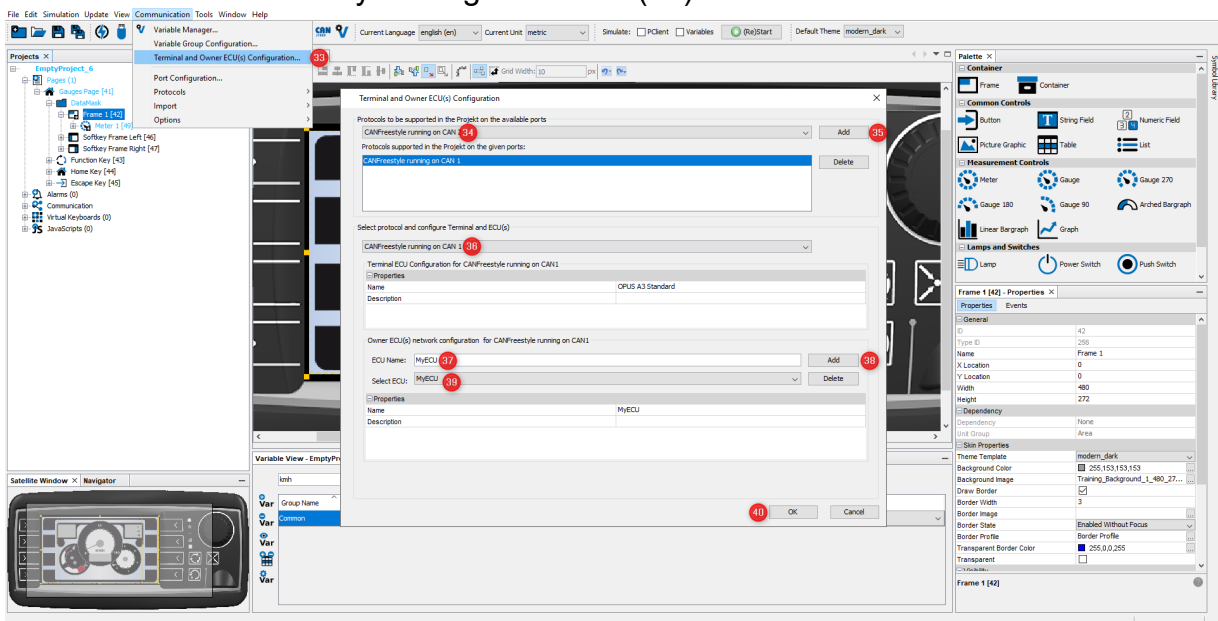
Now it's time to create a CAN mapping for the CAN message and add the **kmh** variable to this message:

- Click on the CANFreestyle shortcut button in the toolbar or, alternatively, go to the menu **Communication -> Protocols -> CANFreestyle -> Configure Mappings...** in the **Communication** menu.
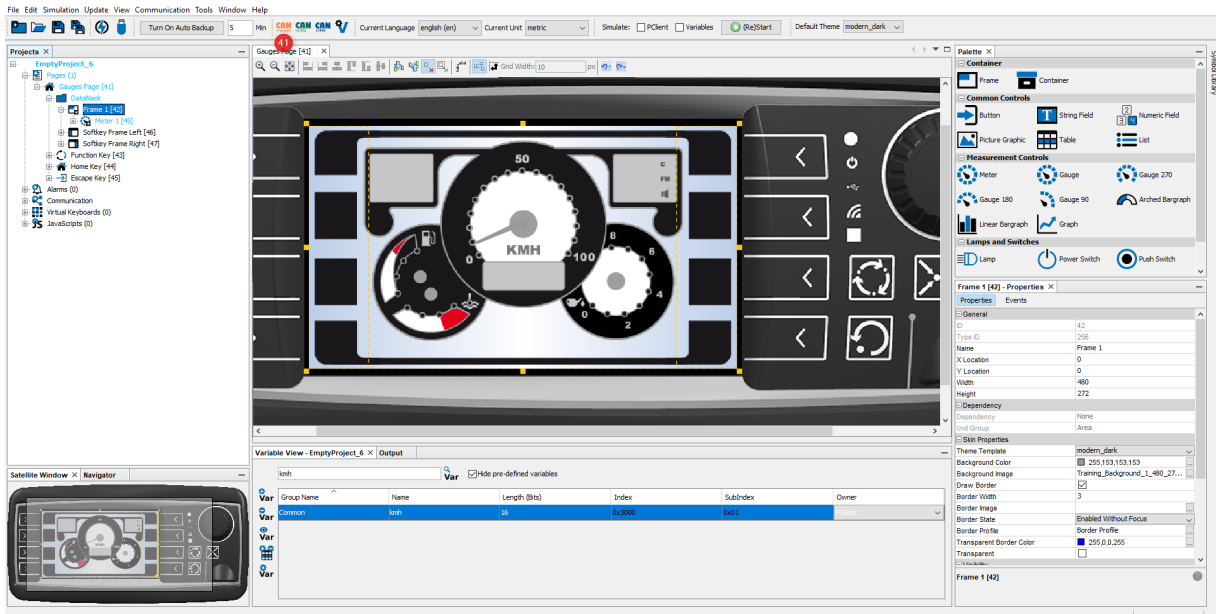
The **CANFreestyle Configure Mappings** dialog appears. Make the following settings:

- Enter **MessageFromMyECU** as the message name (41).
- Choose **Receive** to create a message that is received by the display (42).
- Click the **Add** button to create the receive message (43).
- Select **MessageFromMyECU** (44).
- Change the CAN ID Type to **29 Bit** (45).
- Enter the message CAN ID **0x4711** (46).
- Switch to **Visual CAN Mapping** (47).
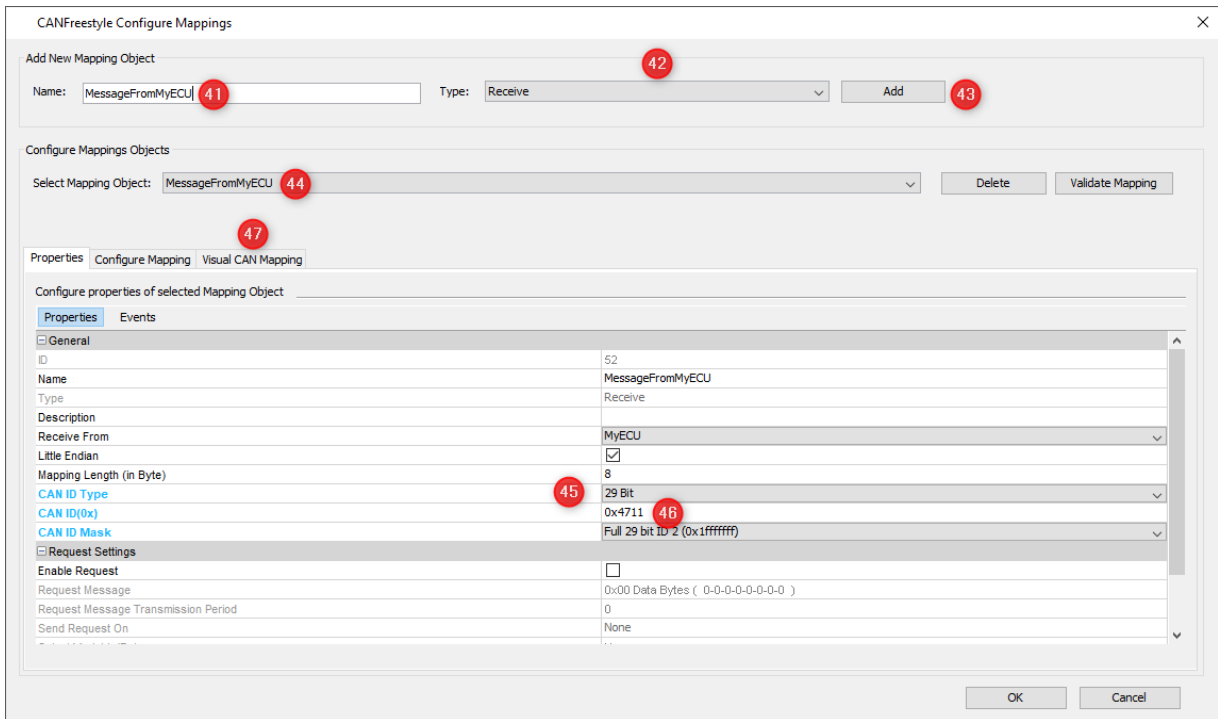
- In the search field on the right (47) type the name of the variable, **kmh**.
- Select the variable (23) and drag & drop it into Bit 1, Byte 1 of the CAN mapping (49).
- Close the window by clicking **OK** button.
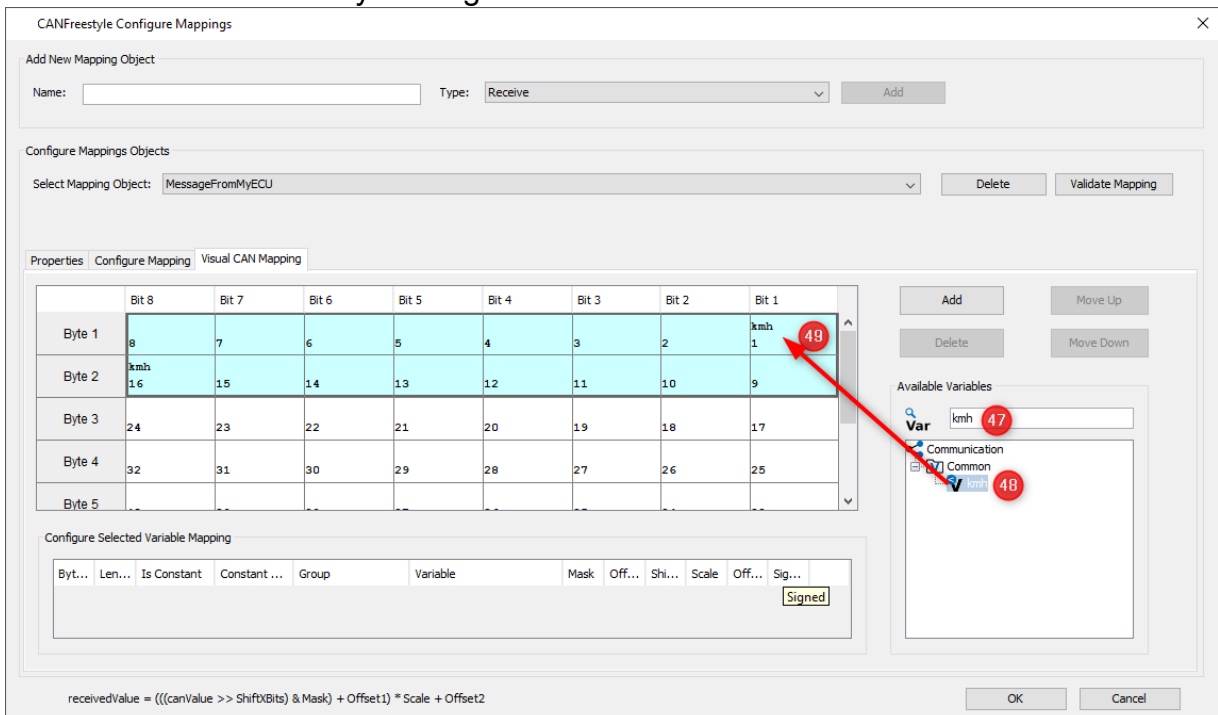
Now you have created a CAN message with CAN ID 0x4711 (Hex) that transmits 4 Byte of data. The data is used as value for variable **kmh**.

Whenever this message is received, the value of **kmh** variable is updated. Because the meter references the **kmh** variable, the meter needle will immediately show the received

value. It's important to remember that the message length has to be the same for the display and the ECU. If only parts of the message are needed, use a constant with mask set to 0 to extend the message.

Create a New Page

In a next step a new page is added. Further the navigation between the two pages is defined.

- To create a new page use the right click context menu and choose **New Page** (50).
- A new page with name **Page 1** is created (51).
- A new editor tab is opened showing the new page **Page 1** (52).
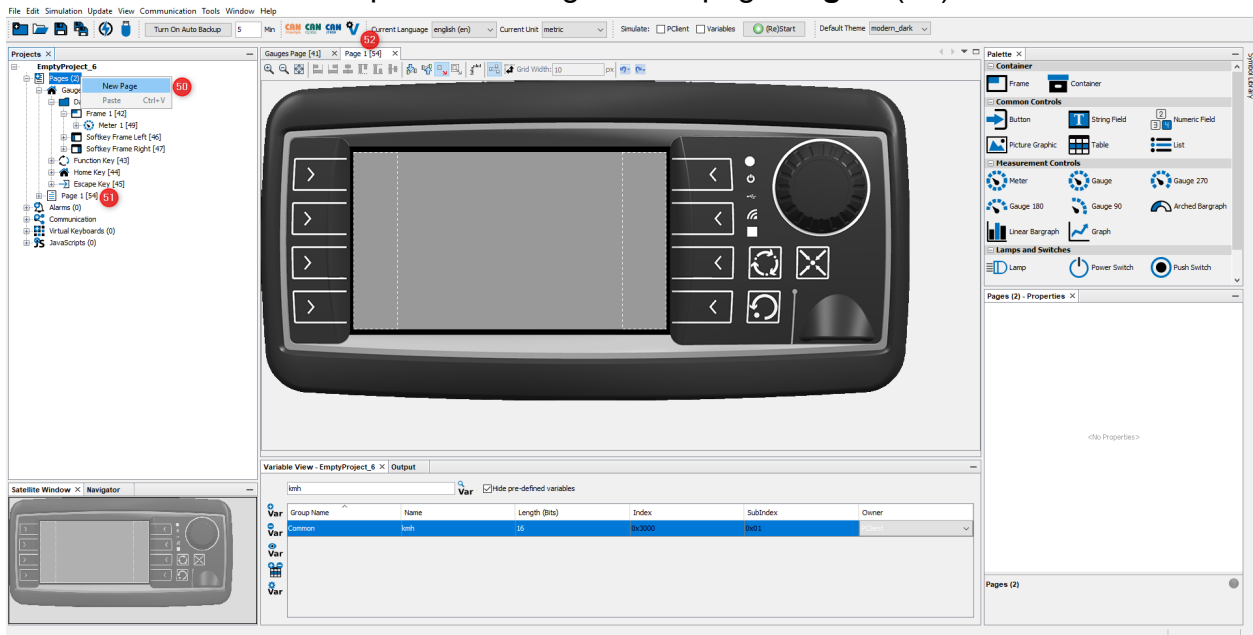
As you did in the first page[13]:
- In the **Properties** window of the page change the name into **Configuration** (53).
- Add the training background 2 image from the Image Library and add it as a background image for the frame (54).
- The new page now has the selected background picture (55).
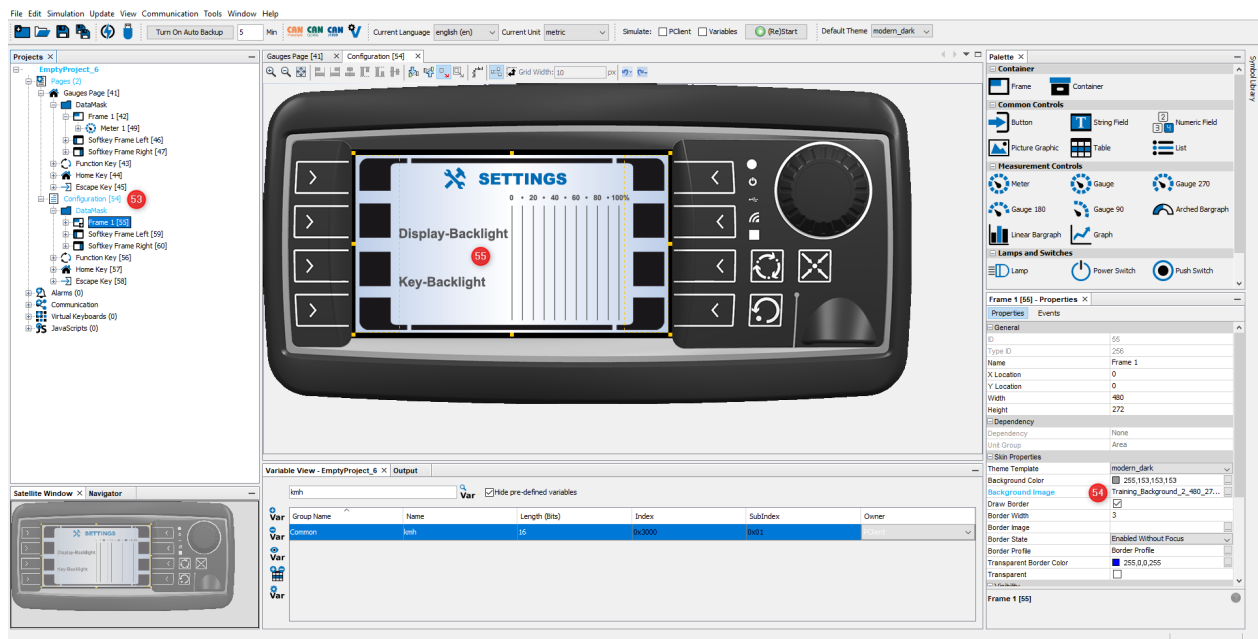
Define Navigation Between Pages

The eight keys at the left and right screen border are called softkeys. You can place icons close to the softkeys into the **Softkey Frame** on the screen to describe the function of each softkey. The function can be triggered by pressing the appropriate key or by pressing the icon on the screen, if a touch screen is available.
Softkeys will be used for the navigation between the pages.

The **Softkey Frames** for the row of softkeys at the left and right screen border were created automatically. But you still need to create a softkey level to configure the keys. We will start with the **Configuration** page:

- Open the right click context menu of **Softkey Frame Left** and choose **Create Softkey Level** (56)
- Alternatively you can right-click on the **Softkey Frame** itself and select **Create Softkey Level** (57)
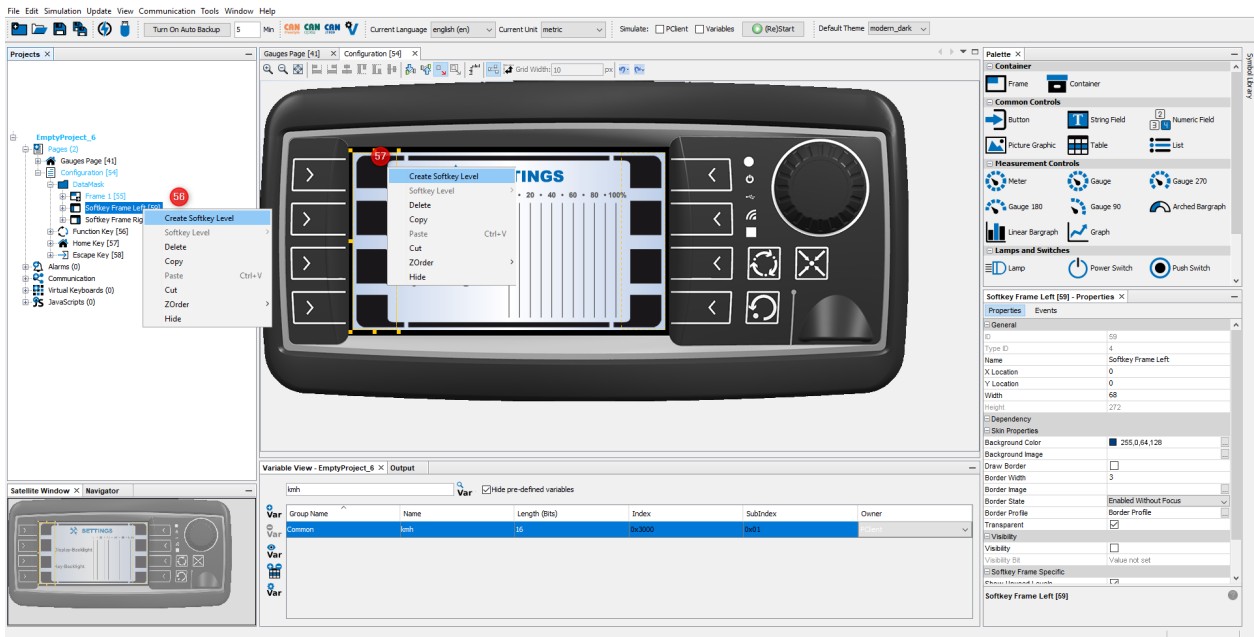
Now we can start configuring the softkeys:
- Select the first **Softkey** in the project tree (58)upper left or in the editor (59)
- Drag & drop the file **BUTTON_BACK.png** from the Image Library on the Softkey, and in the context menu, select the softkey and then **Released Image.**
- Drag & drop the file **BUTTON_BACK_pressed.png** from the Image Library on the Softkey, and in the context menu, select the softkey and then **Pressed Image.**

By switching the state of the **Pressed State** check box (62) you are able to view both softkey states in the editor. Make sure that the property is unchecked at the end.

Next step is to define a softkey event. The event should change to the **Gauges** page when the user presses (or rather, releases) the **Softkey** you just configured graphically:
- Select **Softkey 1** in the **Projects** tree (63).
- Open the **Events** tab by clicking **Events** in the **Properties** window (64).

- Click the "**...**" icon of the **OnRelease** event (65).

The **Configure Events and Actions** dialog opens.

- Un-check **No Action** (66).
- Select **Jump To Page** (67).
- Choose the **Gauges** page (67).
- Close window by clicking **OK** button (69).
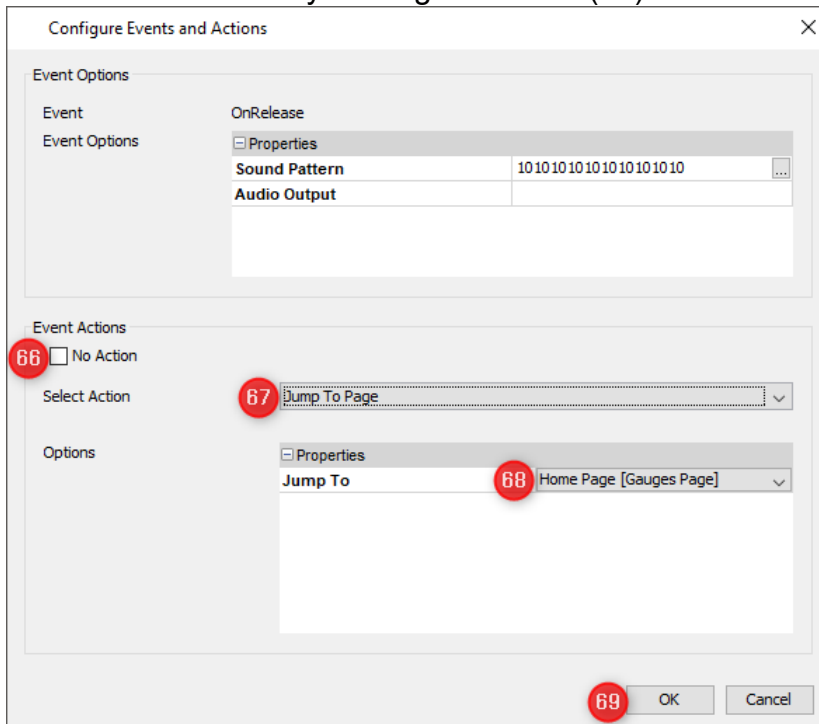
That's all for the page switch from the **Configuration Page** to the **Gauges** page.
We added two images, one for press and one for non-pressed state of the softkey. Further we defined the jump to page action for the softkey on the OnRelease event.

For the navigation from the **Gauges Page** to the **Configuration Page** please follow the same steps (56 to 69) to configure one of the softkeys of the **Configuration** page. The result should look like this:

Let the User change the value

The display user shall be able to change the display backlight intensity. Here is what needs to be done:

- Drag the **Linear Bargraph** from the **Palette** (70) to the page editor (71).
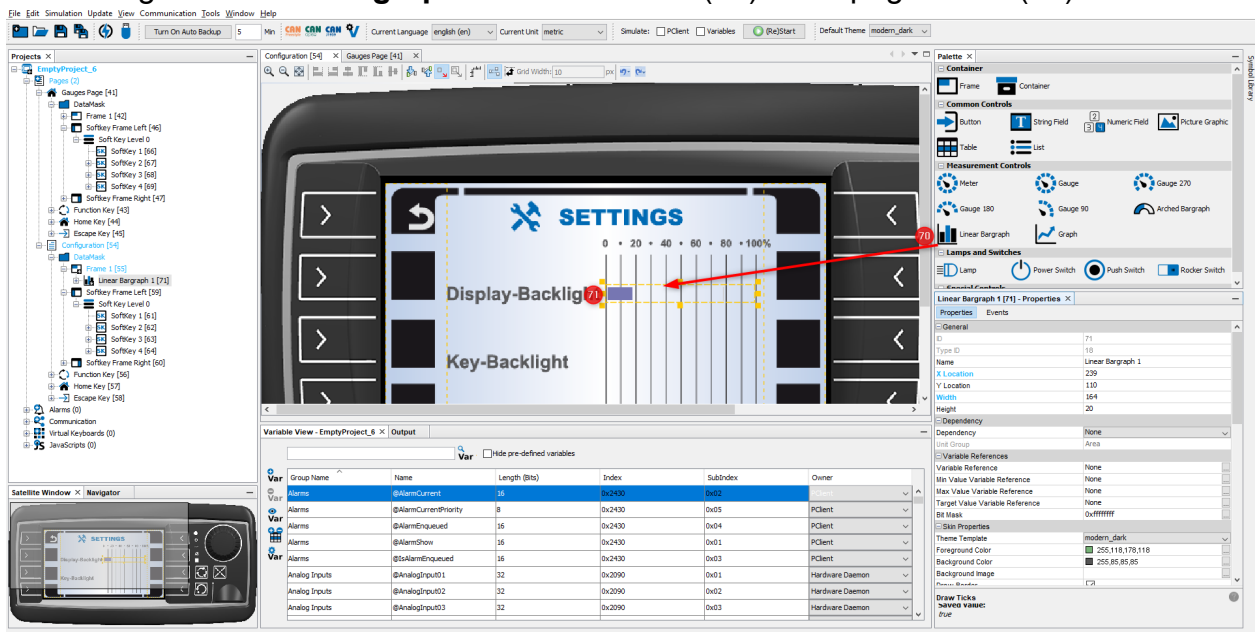
Change the following **Properties** of the **Linear Bargraph**:

- **X Location**: 239
- **Y Location**: 110
- **Width**: 164
- **Height**: 20

- Check **Transparent**
- Check **Set As Input**
- Uncheck **Draw Target Line**
- Uncheck **Draw Ticks**

To use the **Linear Bargraph** to change the display backlight intensity, we need to create a reference from the **Linear Bargraph** to the pre-defined variable **@DispBacklightIntensity**:

- Uncheck **Hide pre-defined variables** in the **Variable View** (72).
- In the **Variable View** search filed enter: **disp** (73).
- Drag the variable **@DispBacklightIntensity** from **Variable View** (74) and drop it on the **Linear Bargraph** (75). Alternatively, you could drop it on the Linear Bargraph entry in the project tree (77)
- A context menu appears. In the first level, select the Linear Bargraph 1 object. In the second level, choose **Variable Reference** (76).
- You can now check in the Properties of the object that the variable is set as the Variable Reference (78).

When running the project on a real display device, the user can use the encoder to change the display backlight intensity. The intensity is changed immediately when the user changes the value of the **Linear Bargraph**.

2.3   Simulate the Project

You can simulate the project on the PC without a device, using our PClient simulation[598] .
To do that, enable the checkboxes for both the PClient and the Variable simulation and then start by clicking the button (Re)Start.
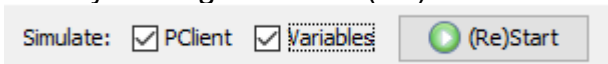
The project will be saved automatically before the simulation starts.

Open the **Gauges Page** in the **PClient** simulation, then follow these steps in the **Variable Simulation**:

- Enter **kmh** in the search filter (1).
- Select the **kmh** variable (2).
- Move the slider (3) and you will see how the meter in the **PClient** simulation will move (4).
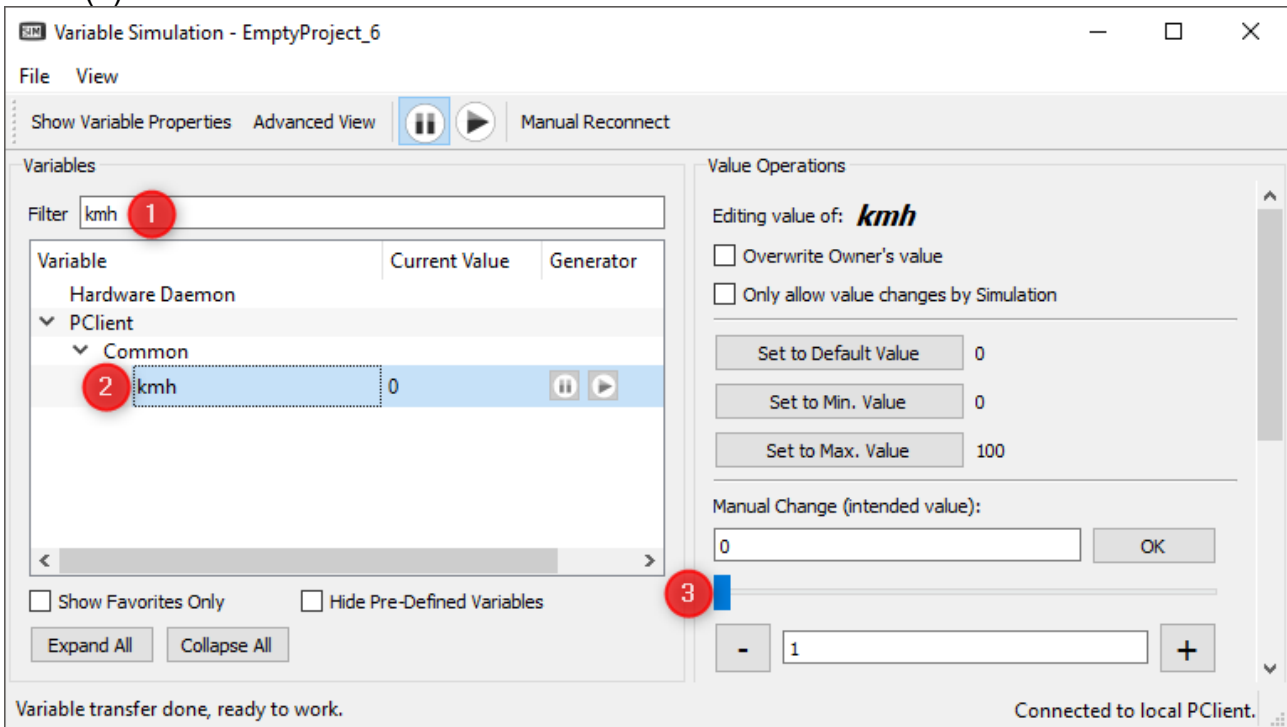
Download the project to the device

First you have to make sure that the software on the device is on the same level as the PC software. If in doubt, perform an update of OS and PClient on your device (there is a manual on how to update your device here[764]).

To download the project to the device, connect a USB stick (FAT 32 formatted) to your computer. There can be other files on the stick, they will be ignored.

Use the Project Update button in the toolbar (the project will be saved automatically), choose the drive letter of the USB stick and click "Transfer" to download the project to the USB stick.

Once the download has finished, remove the USB stick from the PC and connect it to the (running) device. The project will automatically be installed and started.

I'm done. What do I do now?

Congratulations, you have now experienced the basic work flow of this software!

Now you can browse the manual to see what other possibilities there are. Also, check out some sample projects to see what is possible. Try to think of some small tasks to get started creating solutions and to get to know the features, but also the limitations of the software. Especially take a look at the JavaScript[514] page; many advanced ideas can only be done with the help of some scripts. Get familiar with the language and the functions specific to our devices.

User Interface

3   User Interface
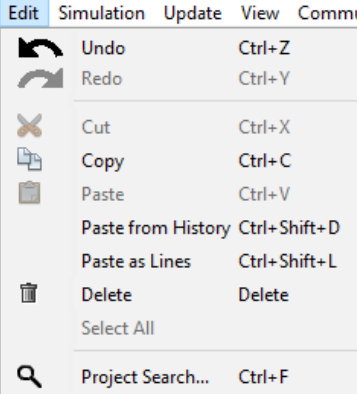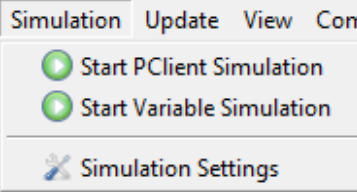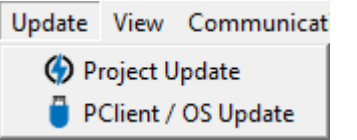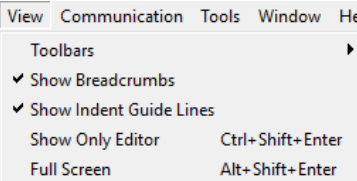
This chapter describes the main elements of the user interface:
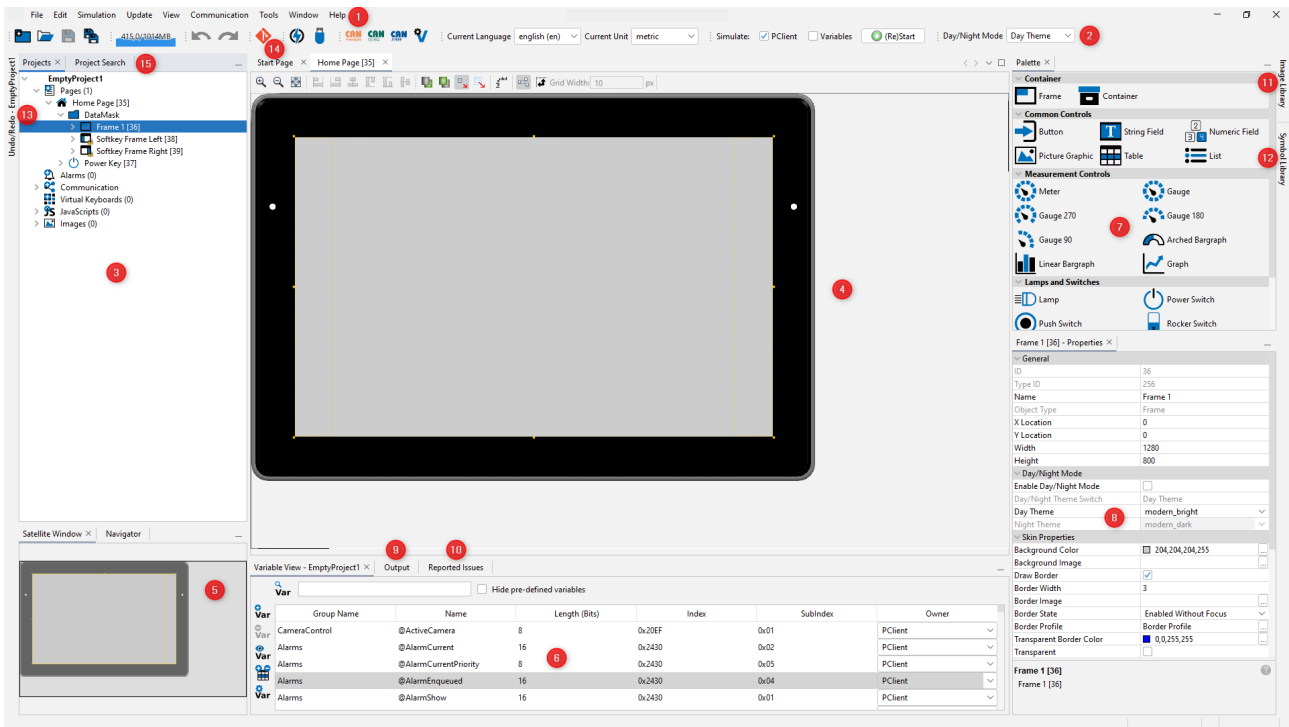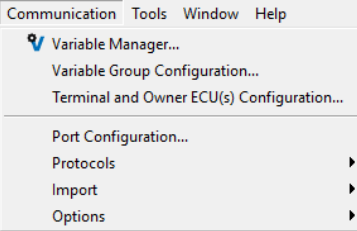
Menu Bar (1)

Menu	Description
File 	**New Project...**: Create a new project. See <u>Quick Start Tutorial</u>[11] to see how to create a new project **Open Project...**: Open an existing project. **Close Project** (project name): Close the currently selected project. **Open in Encoding...**: Opens a file in the JavaScript editor with the chosen encoding **Project Properties** (project name): Edit the <u>project properties</u>[76] of the currently selected project **Create Sample**: Create a new sample project. <u>Read more</u>[130] . **Convert Project**: Convert the current project to other display types. <u>Read more.</u>[40] **Export Project**: Export the project to a ZIP file to send it to someone. **Save Project** (project name): Save the current project. Note that the entry is disabled if no changes were made to the project. **Save As** (project name): Save the current project under a different name. **Save All**: Save all currently open projects. **Save in Encoding...**: Saves the file in the JavaScript editor with the chosen encoding

Menu	Description
	Print to HTML: Only for the JavaScript Editor. Prints the currently opened JavaScript file into an HTML file. **Exit**: Close the program.

Edit
Edit Simulation Update View Commu
↶ Undo   Ctrl+Z
↷ Redo   Ctrl+Y
✕ Cut   Ctrl+X
▤ Copy   Ctrl+C
▦ Paste   Ctrl+V
  Paste from History Ctrl+Shift+D
  Paste as Lines   Ctrl+Shift+L
🗑 Delete   Delete
  Select All
🔍 Project Search... Ctrl+F
	Undo: Reverts the last modification in the project. For a complete list of Undo/Redo steps check the Undo/Redo[142] tab.
	Redo: Re-does the last reverted modification in the project. For a complete list of Undo/Redo steps check the Undo/Redo[142] tab.
	Cut: Cut the currently selected object(s). Note that paste after cutting only works if the parent of the cut object changes.
	Copy: Copy the currently selected object(s).
	Paste -> Paste: Paste the currently selected object(s). The parent object of the source object has to be selected. Read more[42].
	Paste -> **Paste as Reference**: Paste the currently selected object(s) as references. All properties are identical. The parent object of the source object has to be selected. Read more[42].
	Paste -> **Shallow Paste**: Paste the currently selected object(s). Children objects will be reference copies. The parent object of the source object has to be selected. Read more[42].
	Paste from History: Only for the JavaScript Editor. Opens a popup with the last copied texts for pasting.
	Paste as Lines: Only for the JavaScript Editor. Pastes multi-line copied text to one line per multi-cursor.
	Delete: Delete the currently selected object(s). There will be a confirmation dialog.
	Select All: Only for the JavaScript Editor. Marks all text in the currently opened JavaScript file.
	Project Search...: Opens the project search[108] .

Simulation Simulation Update View Com ◉ Start PClient Simulation ◉ Start Variable Simulation ⚒ Simulation Settings	**Start PClient Simulation**: Starts the PClient Simulation[598] . **Start Variable Simulation**: Starts the Variable Simulation[601] . **Simulation Settings**: Settings[596] for the Simulation.
Update Update View Communicat ⚡ Project Update 🖱 PClient / OS Update	**Project Update**: Opens the Project Updater Dialog[618]. **PClient / OS Update**: Opens the PClient / OS Update Dialog[653]

View
View Communication Tools Window Hel
  Toolbars   ▸
✓ Show Breadcrumbs
✓ Show Indent Guide Lines
  Show Only Editor   Ctrl+Shift+Enter
  Full Screen   Alt+Shift+Enter
	Toolbars: Provides customization options for the toolbar.
	Show Breadcrumbs: Only for the JavaScript Editor. Enables or disables the breadcrumbs bar in the JavaScript editor (only useful if classes and functions are used).
	Show Indent Guide Lines: Only for the JavaScript Editor. Enables or disables vertical indent lines in the code view.
	Show Only Editor: Maximizes the currently selected editor window.

Menu	Description
	Full Screen: Switches to Full Screen view.
Communication 	**Variable Manager...**: Opens the [Variable Manager](#)[165] . **Variable Group Configuration...**: Opens the dialog to manage and add [variable groups](#)[165] . **Terminal and Owner ECU(s) Configuration...**: Opens the dialog to add / remove and configure CAN protocols for the project. **Port Configuration...**: Opens the dialog to configure the bus speeds of the CAN ports for the project. **Protocols** -> -**CANFreestyle**-> --**Configure Mappings**: Opens the Mapping Configuration dialog for CANFreestyle. [Read more.](#)[317] --**Transmit Sequences...**: Opens the dialog to configure Transmit Sequences. [Read more.](#)[325] --**CAN ID Masks...**: Opens the dialog to add / remove and configure CAN ID masks. [Read more.](#)[315] -**CANopen** -> --**Configure Mappings**: Opens the Mapping Configuration dialog for CANopen. [Read more.](#)[328] --**Generate EDS File**: Creates an EDS file. [Read more.](#)[328] -**J1939** -> --**Configure Mappings**: Opens the Mapping Configuration dialog for J1939. [Read more.](#)[360] --**DM1/DM2 Settings...**: Opens the dialog to configure DM1 / DM2 support. [Read more.](#)[370] **Import-Export**-> -**Import Variables**: Opens the dialog to import variables from another project or from a .CSV file. [Read more.](#)[59] -**Import DBC File**: Opens the dialog to import a DBC file. [Read more.](#)[381] -**Export DBC File**: Opens the dialog to export a DBC file. [Read more.](#)[391] **Options** -> -**Data type definitions...**: Create and manage variable data types. -**Process Configuration...**: Create and manage processes as variable owners (only for additional C/C++ applications).
Tools	**Generate Key File**: Generate a key file to obtain a license file. **Load License File**: Load a license file to unlock the full version. **Default Password**: Opens the dialog to set a default password that will be used to unlock projects with a [password](#)[76] . **Start Graph History Tool**: Starts the [Graph History Tool](#)[459] . **Reinstall Components**: Opens the dialog to reinstall different external components of the program. Only needed if one of the elements listed is missing files. -**Camera** ->

Menu	Description
Tools Window Help Generate Key File Load License File Default Password Start Graph History Tool Reinstall Components › Camera › Import Export Language CSV Plugins Revision Browser Options	-- **ISO 17215 Camera Configuration**: Opens the Ethernet Camera configuration tool[790]. **Import Export Language CSV**: Opens the dialog to export / import a language CSV file. Read more.[98] **Plugins**: Opens the plugins management dialog. Read more[660]. **Revision Browser**: Opens the project versioning[123] revision browser. **Options**: Opens the options dialog. Read more.[65]
Window Window Help Palette Properties Projects Ctrl-1 Files Ctrl-2 Navigator Output Ctrl-4 Reset Windows Close Window Ctrl-W Project Search... Image Library Symbol Library Satellite Window Variable View Undo / Redo Reported Issues	**Palette**: Opens the object palette window. **Properties**: Opens the object properties / events window. **Projects**: Opens the Projects Tree window. **Files**: Opens the project file view window. **Navigator**: Opens the JavaScript code structure window. **Output**: Opens the output information window. **Reset Windows**: Resets all windows of the program to the default sizes and positions. **Close Window**: Closes the currently selected window. **Project Search...**: Opens the Project Search window. **Image Library**: Opens the Image Library tab. **Symbol Library**: Opens the Symbol Library tab. **Satellite Window**: Opens the Satellite window. **Variable View**: Opens the Variable View window. **UndoRedo**: Opens the Undo/Redo window. **Reported Issues**: Opens the Reported Issues window. Note that if the respective window is already open, the according menu entry doesn't do anything.
Help Help Start Page Report an issue or give feedback About Help F1	**Start Page**: Opens the start page[134] with Welcome text, release notes, RSS news and recent projects. **Report an issue or give feedback**: Opens the reporting dialog[730]. **About**: Opens the version info dialog. Here you can see all software versions, get detailed information and read the list of open source licenses. **Help**: Opens this HTML help. By default, the online version of this help is loaded. If no internet connection is available, a local version of the help is loaded. Please note that the offline version is not refreshed as often as the online version.

Toolbar (2)

Icon	Description
📁	Create a new project.
📂	Open a project.
💾	Save selected project. Only active if changes have been made to the project.*
💾	Save all open projects.
↩ ↪	[Undo / Redo](#)[142]
◈	Opens the [revision browser](#)[123] .
⚡	Saves the project and opens the Project Updater Dialog. [Read more.](#)[618]
🔌	Opens the PClient / OS Update Dialog. [Read more.](#)[653]
CAN Freestyle CAN open CAN J1939	Shortcuts to mapping dialogs for [CANFreestyle](#)[308] , [CANopen](#)[328] & [J1939](#)[342] .
⚙V	Opens the [variable manager](#)[165] .
Current Language english (en) ▾	Change the [language](#)[98] shown in the [page editor](#)[73] .
Current Unit metric ▾	Change the [unit](#)[98] shown in the [page editor](#)[73] .
Simulate: ☐ PClient ☐ Variables ◉ (Re)Start	Select and start PClient and/or variable [simulation](#)[596] . The project will be saved before the simulation starts.
Day/Night Mode Day Theme ▾	Switch between Day or Night mode view in the [page editor](#)[73] .

*We are trying to be perfect, but it might be that some actions are not tracked and the save button stays disabled even though you changed something. In that case please
- contact us! Let us know what you did that wasn't tracked.
- Use the Save all button which will never be disabled.

Projects Tree (3)

The project tree shows the logical structure of the project.
Here is what you can read from the following tree:
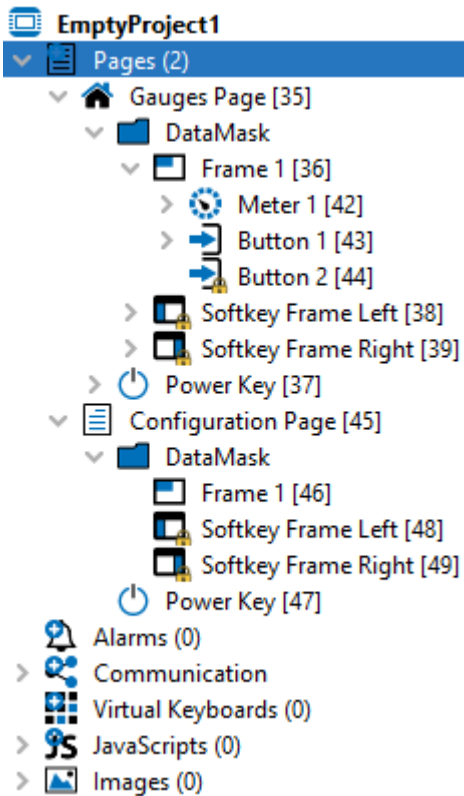
- The project name is **EmptyProject1**
- The project contains the categories **Pages**, **Alarms**[568], **Communication**[307], **Virtual Keyboards**[446], **JavaScripts**[514] and an **Images Node**[54]
- The project has two pages: **Gauges Page** and **Configuration Page**.
- The **Gauges Page** is the home page of the project (home icon).
- The **Gauges Page** has a **DataMask** (i.e. an empty drawing area) and a **Power Key**.
- In the **DataMask** (i.e. drawing area) there is a **Frame 1** and a **Left** and **Right Softkey Frame**.
- A **Meter 1** and two **Button** objects are located in the **Frame 1**.
- The second button object (ID 44) has been locked in position and size[75].

The tree shows a parent-child relationship. Example:
- The **Meter 1** is a child of the **Frame 1**.
- The **Frame 1** is the parent of the **Meter 1**.

The parent-child relationship defines how objects that are visible on screen are drawn. On the one hand, a child can never be drawn outside of its parent. I.e. the child is clipped at the parent borders. On the other hand, the order of children defines which object is in front of another object (this is called Z-order[151]). For example:
- **Softkey Frame Right** is drawn in front of **Softkey Frame Left**.
- **Softkey Frame Left** is drawn in front of **Frame 1**.

We also provide additional information through colors and icons:

```
⊟···⚠ EmptyProject_1
   ⊟···⚠  Pages (1)
      ⊟···⚠  Home Page [41]
         ⊟···⚠  DataMask
            ⊟···⚠  Frame 1 [42]
               ···⚠  String Field 1 [49]
               ⊞···   Picture Graphic 1 [54]
            ⊞···   Softkey Frame Left [47]
            ⊞···   Softkey Frame Right [48]
         ⊞···   Function Key [43]
         ⊞···   Home Key [44]
         ⊞···   Escape Key [45]
         ⊞···   Encoder [46]
   ⊞···   Alarms (0)
   ⊞···   Communication
   ⊞···   Virtual Keyboards (0)
   ⊞···   JavaScripts (0)
```
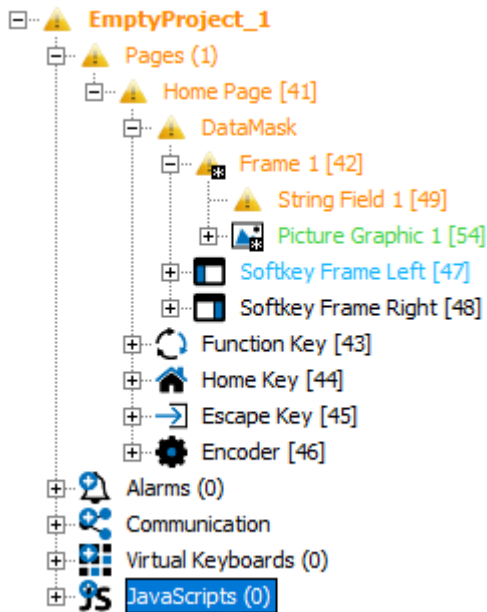
Green - This object has been created since the last save. Also indicated with a black asterisk next to the object icon.

Blue - This object has been modified since the last save (or a child object has been added). Also indicated with a black asterisk next to the object icon.

Orange - There is a problem with the properties for object. Select the object and then the colored property or event to see a description of the problem. Also indicated with the yellow warning icon.

String Field 1 [49] - Properties ✕		—
Properties	Events	
⊟Events		
OnValueChange	No Action	...
OnEnterValue	Execute Script [newEmpty_JavaSc...	...
OnEnterNewValue	No Action	...
OnEscape	No Action	...
OnOpenForInput	No Action	...
OnEnabledWithFocus	No Action	...
OnEnabledWithoutFocus	No Action	...

OnEnterValue   ⍰

Attention:
Selected Script File does not exist!
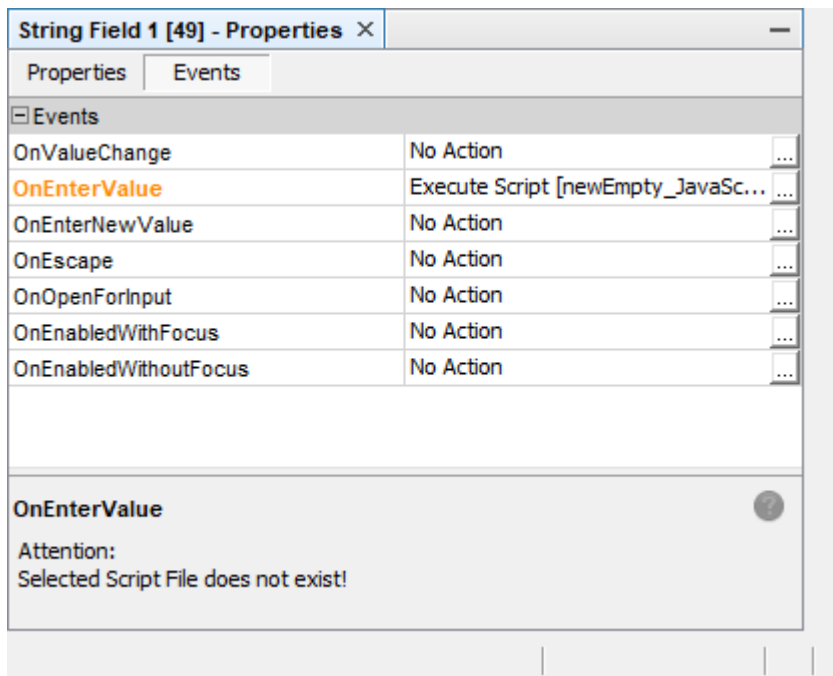
The colors will be passed to the parent objects up to the project name. Note that with a new object, only the object itself will be marked green. All parents will be marked blue, since for the parents the new object is a change, but the parents themselves are not new.

In case there is both a problem and an object modification, the problem will be shown in the project tree with higher priority.

When a project is loaded and the project contains warnings, the search tab will be opened automatically, showing all warnings in the project.
Don't discard these warnings! It might be nothing important, but more often than not something is not right in the project if warnings are shown, so please check them.

Objects can be renamed in the project tree by selecting them and pressing F2, or clicking on them again after selecting them.

Pages can be moved in the project tree to change their order for a better project overview.
With <Alt> + <PgUp> / <Alt> + <PgDn> pages can be moved one step higher / lower.
With <Alt> + <Pos1> / <Alt> + <End> pages can be moved to the top / bottom.

The **Alarms** category contains all alarms [568] in the project.
The **Communication** category contains all variables and CAN mappings sorted by owner / protocol. Note that CAN mappings can be deleted in bulk here.
The **Virtual Keyboards** category contains the virtual keyboards[446] used in the project.
The **JavaScripts** category contains all Javascript[514] files in the project. Files can be copied and deleted here as well as dragged & dropped[43] onto objects.
The **Images** category contains the Image Pool[54], all images that are used in the project.

Page Editor (4)

The Page Editor[73] allows the design of the graphical user interface.

Satellite Window (5)

The satellite window shows the whole display device and the viewing part visible in the editor. It is useful for a fast navigation when the page editor[73] is zoomed in and only shows a small piece of the whole display.

Variable View (6)

The Variable View allows viewing, creation and editing of variables. Furthermore it can be used to easily link a variable with an object in the editor by dragging & dropping the variable to the object.

Palette (7)

The palette contains all the graphical objects[412] that can be dragged & dropped into the editor.

Properties Window (8)

The property window shows all the properties[415] available for the currently selected object.

Output (9)

The output window provides information, warnings and errors that occur while using the program. They might help finding the root cause if you run into trouble.
It will also show results of J1939 DM1/DM2 translation file validations.
Also, system console outputs from the PClient simulation[598] will be printed here.

Reported Issues (10)

The reported issues tab provides an overview of the [reported issues](#)[740] from this version of the program.

Symbol Library (11)

This button opens the [symbol library](#) [135] tab.

Image Library (12)

This button opens the [image library](#) [46]tab.

Undo/Redo (13)

This button opens the [Undo/Redo](#)[142] tab.

Start Page (14)

The new [Start Page](#)[134] shows the recently opened projects, information about the current version and an RSS feed with news and update information.

Project Search (15)

This button opens the [Project Search](#)[108] tab.

3.1 Convert Project

This chapter describes how a project can be converted between different devices.
The dialog can be opened via the menu File -> Convert Project or by right-clicking the project name and choosing Convert Project in the context menu.
Please note that for both actions you need to have a project marked in the project tree.
Please note that you should save the project before the conversion to include all changes.

File	Edit Simulation Update View Communication
📁	New Project...
📂	Open Project... Ctrl+O
	Open Recent Project ▸
	Close Project (DemoProject)
	Project Properties (DemoProject)
	Create Sample
	Convert Project

DemoProject
 Pages (1)
 Home Pa...   Properties
   Data...   Close
     F...   Rename
     S...   Convert Project
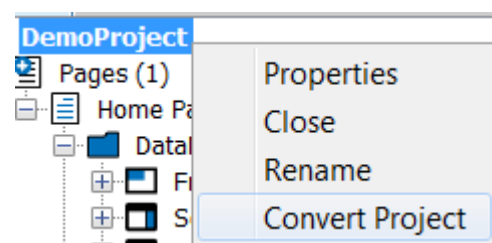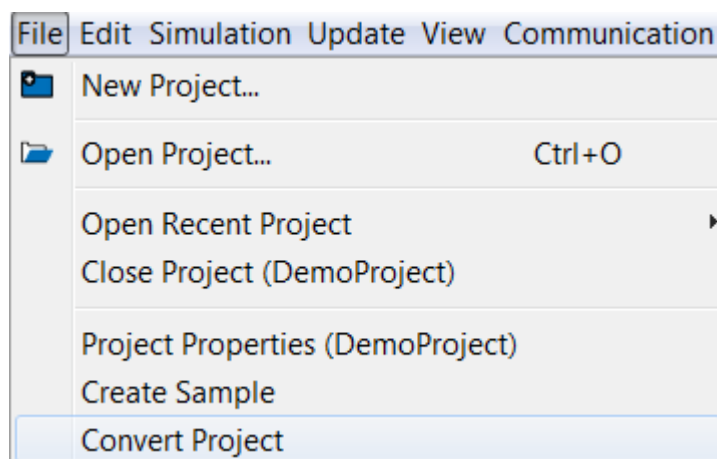
Step 1
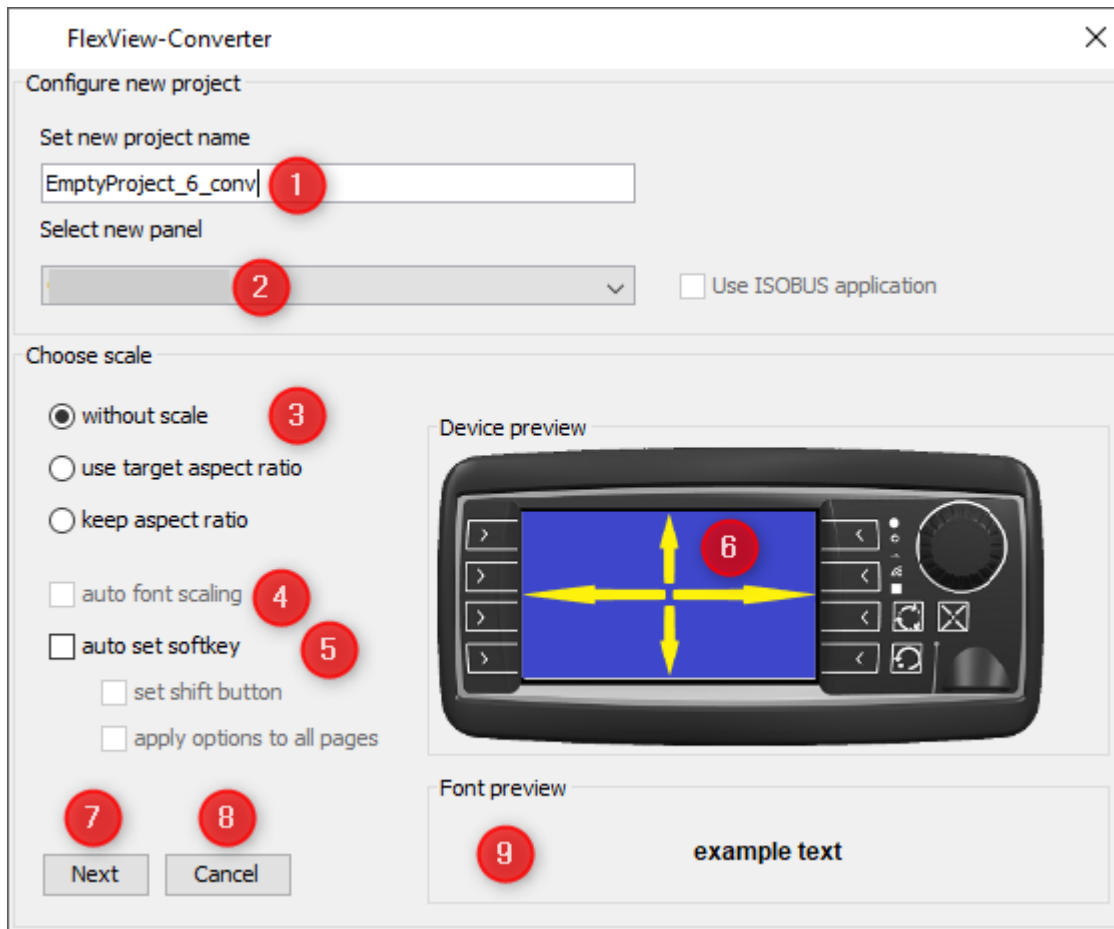
First, you can set the name of the converted project (1).
Next, you can choose the new device type that you want to convert to (2).
Below that, you have three options to scale the page contents (3):

- without scale (keep the object sizes as they are) -> only available for some device types, since content for a larger device can't be displayed on a smaller device without scaling them
- use target aspect ratio (scale objects to the full screen, independent of the existing aspect ratio -> This will make content full screen, but it might look less good because the aspect ratio changes
- keep aspect ratio (scale objects to the full screen, but keeping the existing aspect ratio -> This will make content full screen, but with a bar on the bottom or side. The aspect ratio will be kept.

You can see an example preview of how the page content will be scaled in the Device preview (6).

Next, there is the option for the auto scaling of texts. This is not available if "without scale" has been selected above. A preview of how test will be scaled can be seen in the lower right (9).

Below that are options for the handling of the soft keys (5):

- auto set soft key (soft keys will be spread to the new layout automatically) -> Please note that when converting from a device with more soft keys to a device with less soft keys, several soft key levels will be created
- set shift button (if several soft key levels are created, shift up and shift down buttons will be added to be able to shift between the soft key levels)

- apply options to all pages (the Next button (7) will change to Convert, and the project can be converted with one click, skipping Step 2.

Pressing Cancel (8) cancels the conversion process and all settings are lost.

Step 2 - Soft Keys

In the second step, softkeys can be assigned to the keys of the new device of the conversion process. The new device type with its soft keys is displayed in the center of the dialog. The soft keys of the old device are displayed to the right and left (if available) of the new device type. Soft keys can be dragged and dropped to the new device soft keys or back out to change the order. By clicking the button Next Page, the next page in the project with configured soft keys will be shown. All pages need to be set to finish the conversion process. Press Convert to go to the final step. By clicking the Back button you can go back to previous pages. When the first page is shown, clicking Back will go to the first step of the conversion. By doing that, all existing soft key mappings will be lost! Also, all soft keys that were not mapped will be lost as well. Pressing Cancel cancels the conversion process and all settings are lost.

Step 3 - Finish

In the final step, you can choose if you want to overwrite the existing project or create a new one. Pressing Cancel at this stage brings up the previous conversion step, keeping all settings.
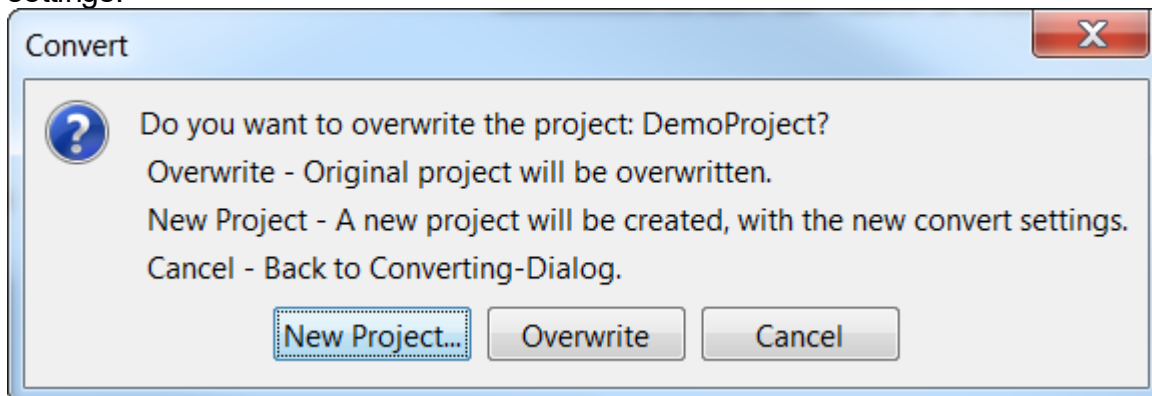
3.2   Copy and Paste

 Tutorial video about copy & paste
https://www.youtube.com/watch?v=2YMmZkwBMwI

When using Copy & Paste, the pasted object will be put on the scene with an offset of 10 px to the right and bottom if the object is put on the same parent object (e.g. on the same frame). When pasting on a different parent object, the position will be the same as with the original object.

There are three types of copy & paste in the program: Normal, reference and shallow.

Normal Paste

Objects can be copied and pasted by either right-clicking on the selected object(s) and selecting "Copy" and then right-click in the target parent object and select "Paste", or by pressing CTRL+C with the object(s) selected and pressing CTRL+V. This works multiple times which creates multiple copies of the same source object.

Please note: Pressing CTRL+C and then directly CTRL+V will not do anything for a copy between pages and when copying multiple objects at once. In this case the parent object has to be selected prior to pasting.

Please note: Direct copy & paste does NOT work for the objects Container [466] and List[477] , and also both object types cannot be pasted into each other, i.e. a list cannot be pasted into a list. Usually doing that will result in bad design, but if there is a good usecase for it, the user can paste on the parent object and move the list into the other list in the project tree.

Reference Paste

In addition to the normal paste there is also a reference paste. The difference is that with the normal paste, a new object will be created, with a new ID, its own set of properties, events etc. With reference paste, a carbon copy of the source object is created. The new object has the same ID as the source object and is identical in any way. Any change to one of the objects will reflect in the other, as well, with the exception of the position of the objects in their parent object.
This type of paste is to be used if you want the same content on several pages, e.g. a clock somewhere in a corner.

Shallow Paste

The last type of paste is called shallow paste. It only makes sense for objects that have child objects. The copied parent object will be a normal copy, whereas the child objects will be reference copies. This type of paste is useful e.g. for list objects with a set of images, like a set of warning / error icons. The list object will be a normal copy, so the copied one can show a different image than the source, but the images in the list should be identical for both lists.
If shallow paste is used with an object without children, the result will be a normal copy of the object.

3.3   Drag and Drop

This chapter describes the different ways drag and drop can be used to make workflows easier.

Objects from the Palette[412] can be dropped

* on objects in the Page Editor
* in the project tree
* in a JavaScript file (if several objects are selected when dragging, the entry "Create JavaScript ID Array" is available, which creates a JavaScript line that initializes a JavaScript array with the ID's of the selected objects as elements of the array).

A new object will be created if it can be created as a child object at that place. Otherwise a "forbidden" symbol will be displayed.

Symbols from the Symbol Library[135] can be dropped

- on objects in the Page Editor
- in the project tree

JavaScripts from the Project Tree can be dropped

- on objects in the Page Editor
- on Page nodes in the project tree
- on the Project name in the project tree

Variables from the Variable View can be dropped

- on objects in the Page Editor
- in the project tree

Images from the Image Library[46] can be dropped

- on objects in the Page Editor
- in the project tree

Images from the Image Pool[54] can be dropped

- on objects in the Page Editor
- in the project tree

For variables, symbols, images and scripts, there is a two-level context menu when they are dropped somewhere. Because of the tree-like structure of the objects, it is not always obvious where the dragged element should be placed, e.g. with a List object containing a String Field. A symbol could be placed into the String Field or into the List object.
In the first level the user can select in which object the symbol should be used. In the second level the user can specify in which way the symbol should be added to the object.
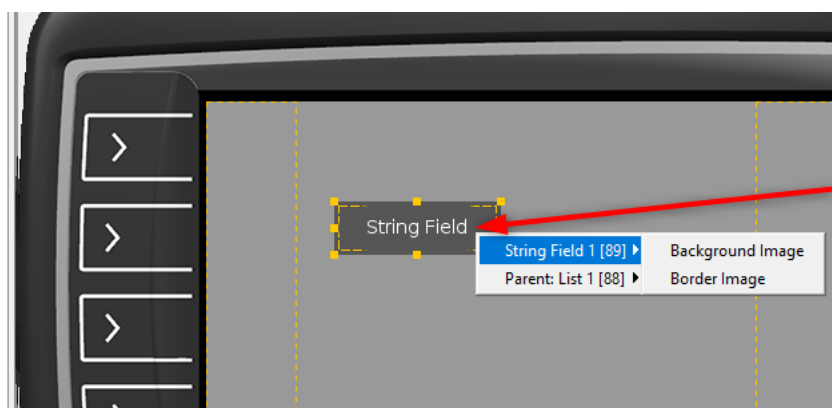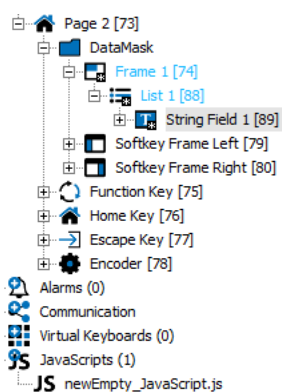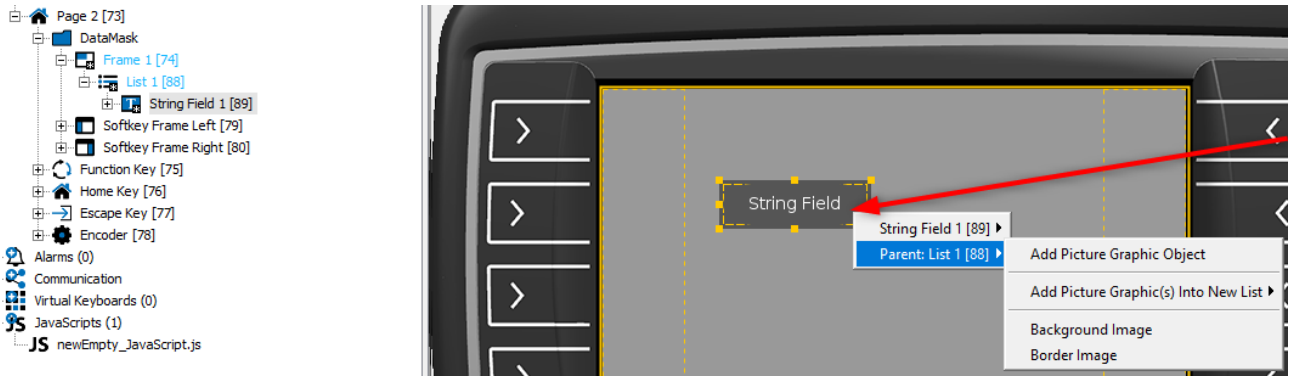
The same goes for JavaScripts. The first level of the context menu shows the object selection with the child object and, if applicable, the parent object, the second level will show the available events for the selected object.
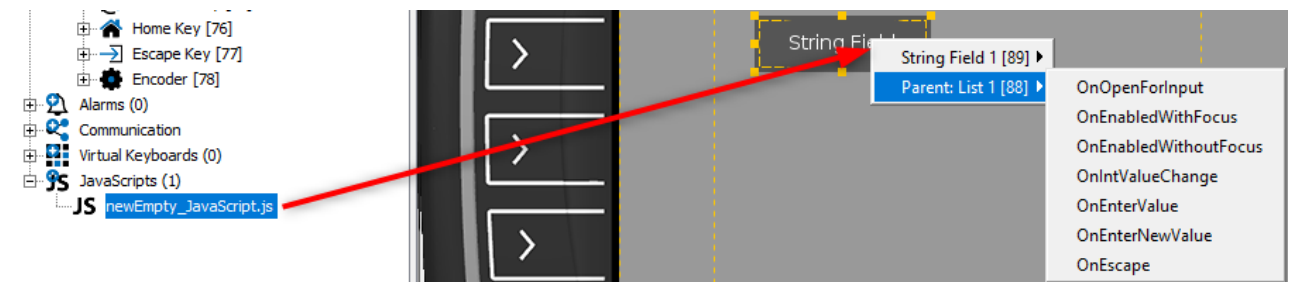
Similarly, when a variable is dropped from the Variable View, the first level of the context menu shows the object selection with the child object and, if applicable, the parent object, the second level will show the available Variable Reference options for the selected object.

Please note that the context menu only covers the the child object and the first parent, i.e. the parent of the list object itself is not included anymore.

Objects that don't have the possibility to add the dropped element, will not be displayed in the first level of the context menu.

3.4 Image Library

 Tutorial video of the Image Library
https://www.youtube.com/watch?v=IjAdGKhiNks

The image library can be used to quickly access images from folders, skipping the step to manually load images one by one with Picture Graphic [496] objects.

The image library can be accessed in the toolbar on the right side (in the default view).
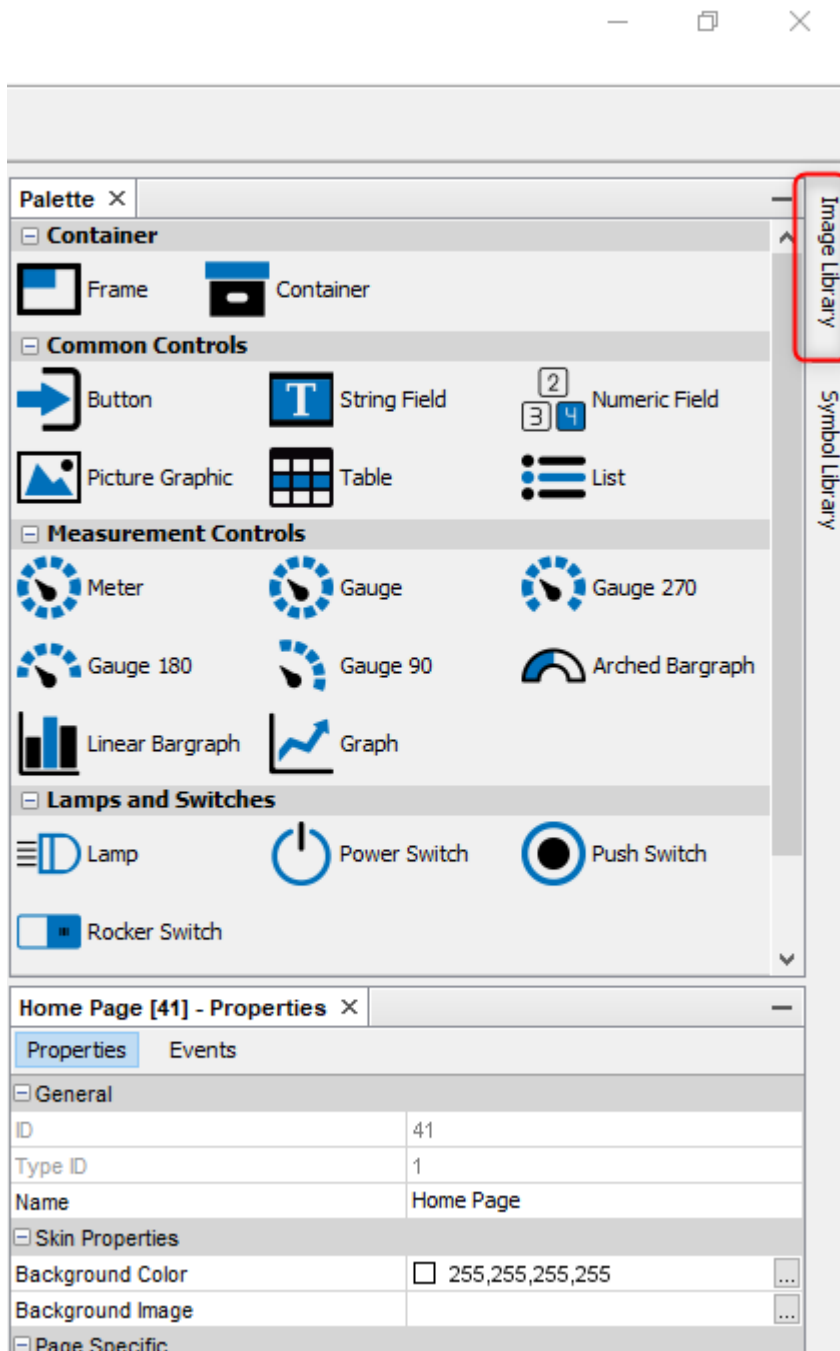
Click on the name to open the tab.
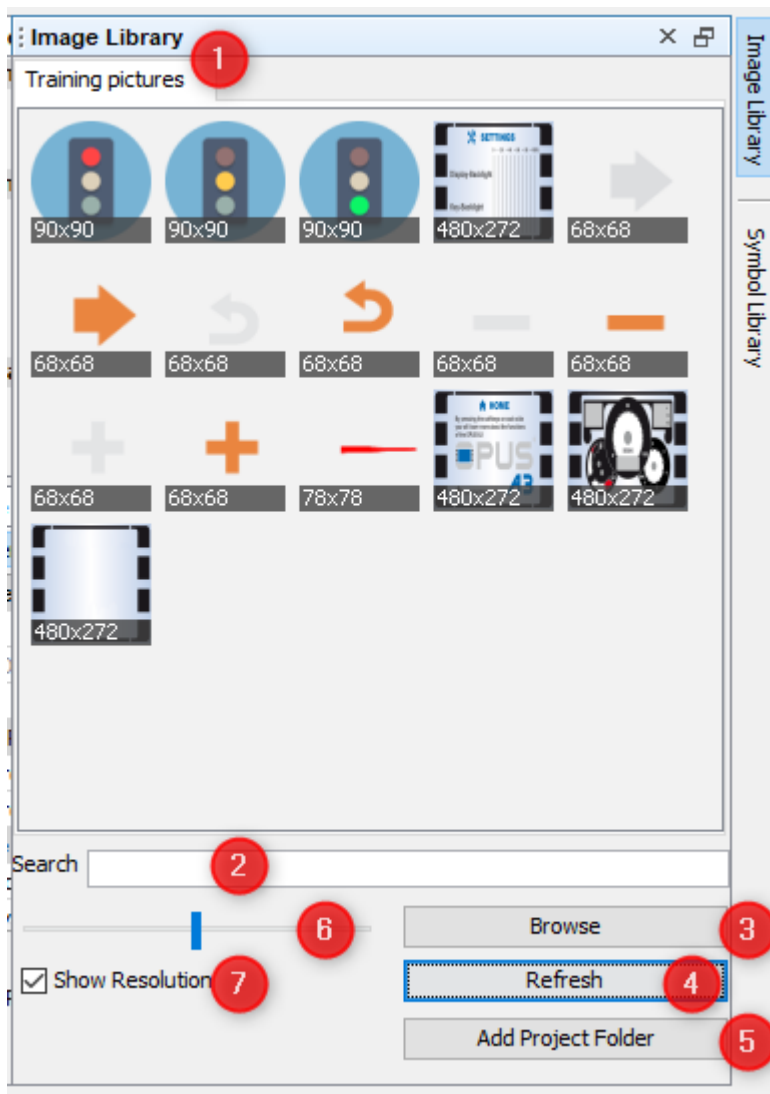
In the image above a folder named Training pictures was loaded. Several image folders can be loaded, they will be shown in tabs (1). Please note that when adding many ordners with many images (especially high-res), showing the image library for the first time will take a while to cache the preview images.
Each image will be shown in a small squared symbol.

The opened image folders can be searched (2). The search will filter live, searching for the entered term in the file name of all images in all opened image folder tabs.
Found images will be shown in an automatically created Search tab.
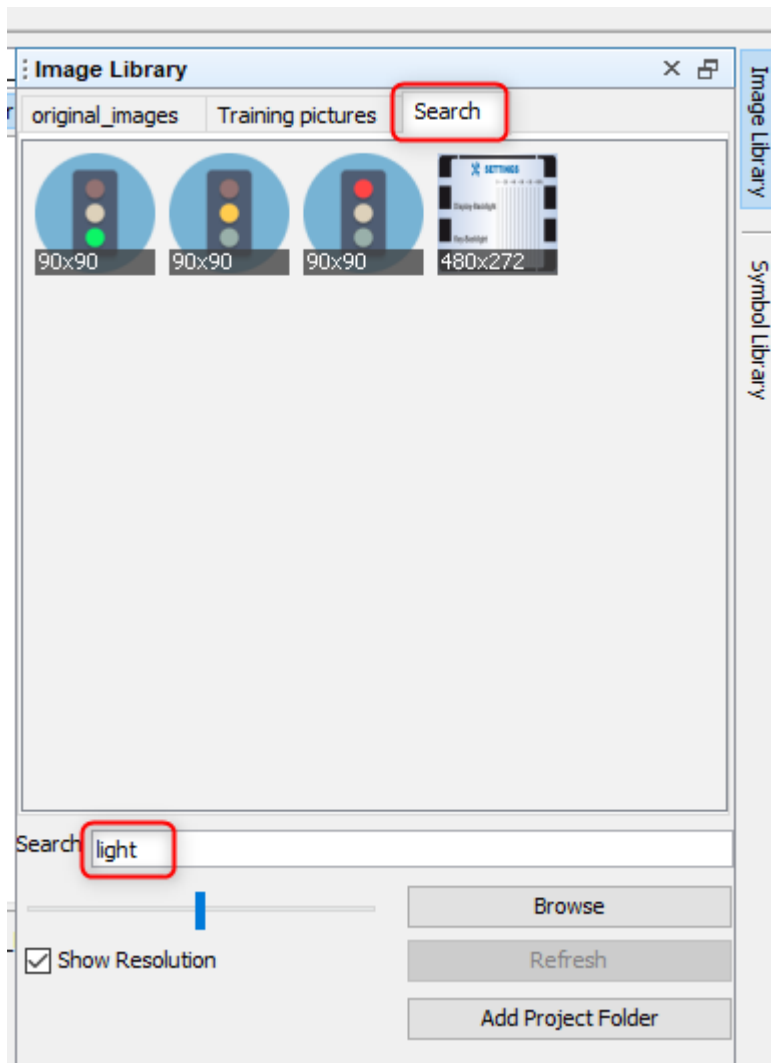
Note that the Search tab can't be closed directly. To close the Search tab, remove the search word from the Search field

A new folder can be added with the **Browse** button (3). Browse to a folder that contains images and confirm by clicking Open.
Note that folders aren't added recursively, i.e. only images directly in the selected folder will be added.
If the images in the currently selected folders have changed or new ones have been added, press the **Refresh** button (4) to refresh the preview.
To quickly add the original images folder of the currently selected project, click the **Add Project Folder** button (5).
Note that all opened folders will be visible no matter which project is selected in the program.

With the zoom slider (6) the size of the preview images can be modified. Dragging it to the left means more, but smaller preview images, dragging it to the right means fewer, but larger preview images.
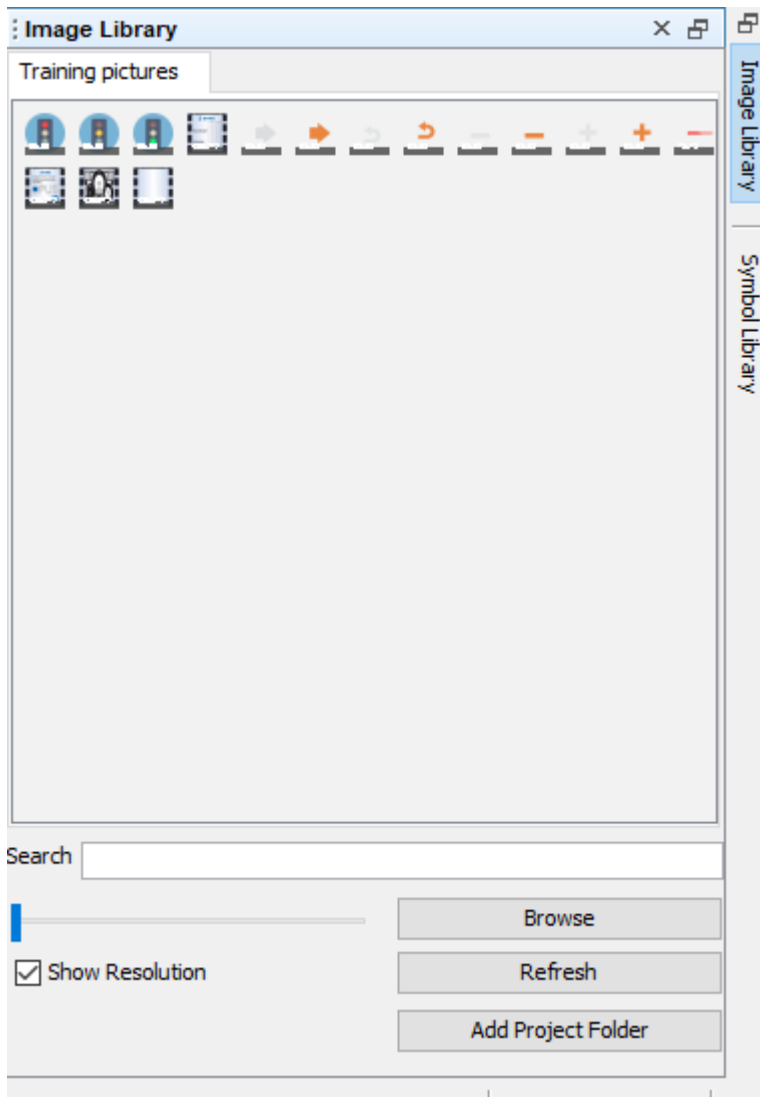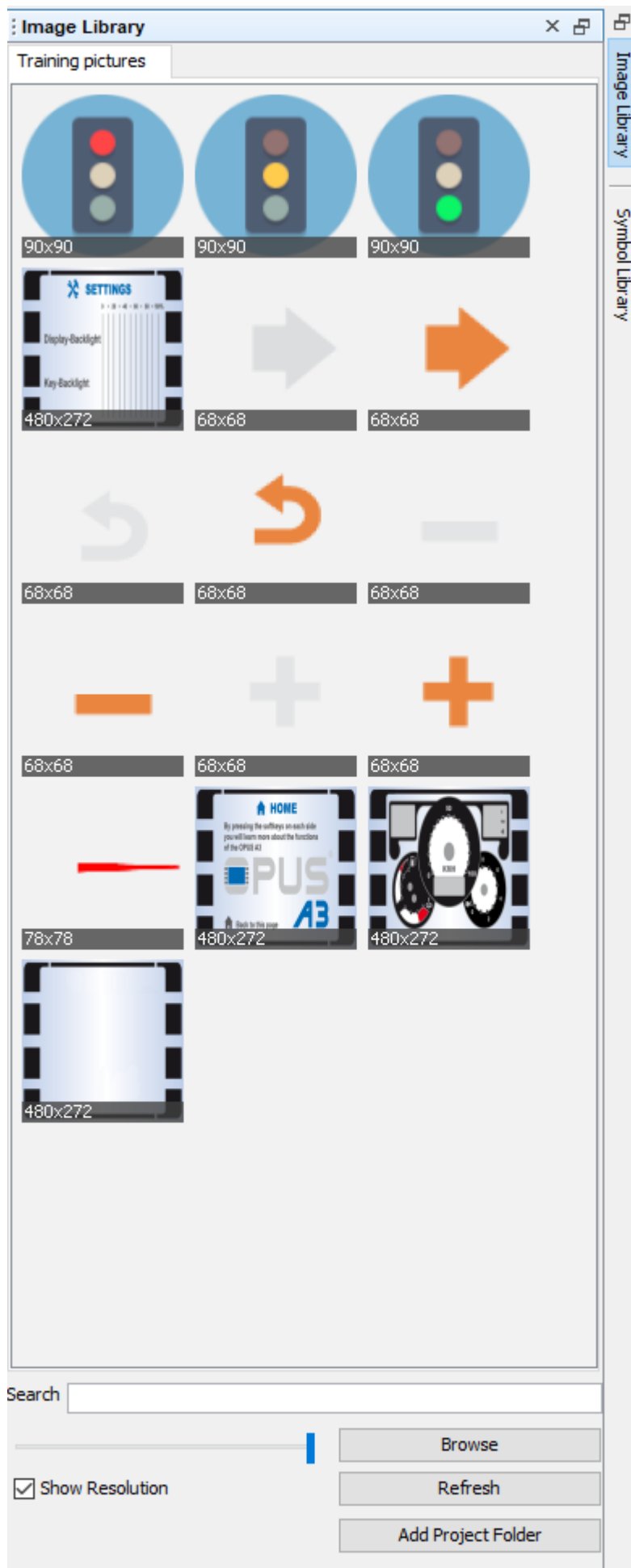
Uncheck the **Show Resolution** checkbox (7) to remove the resolution bar under the preview images.

The complete folder path will be shown as a a tip when hovering on the folder tab.
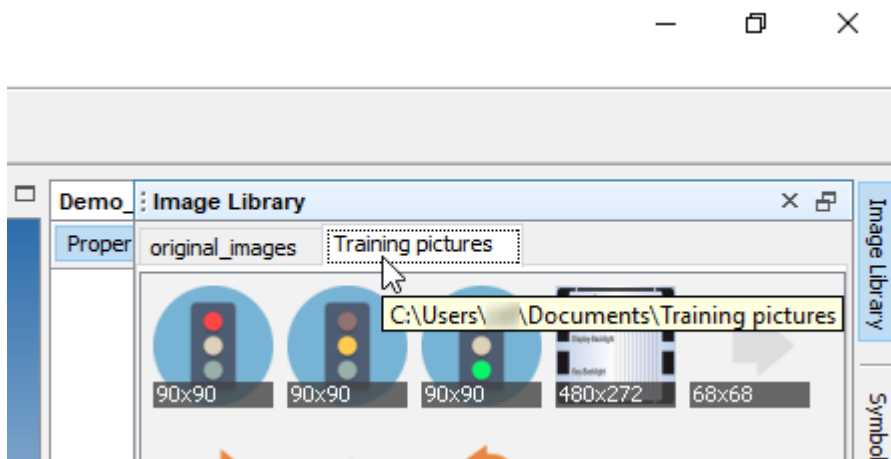
Hovering over the images will show the file name of the image.

Folder tabs can be closed by pressing the X icon next to the folder name in the tab, except for Favourites and Search tab.
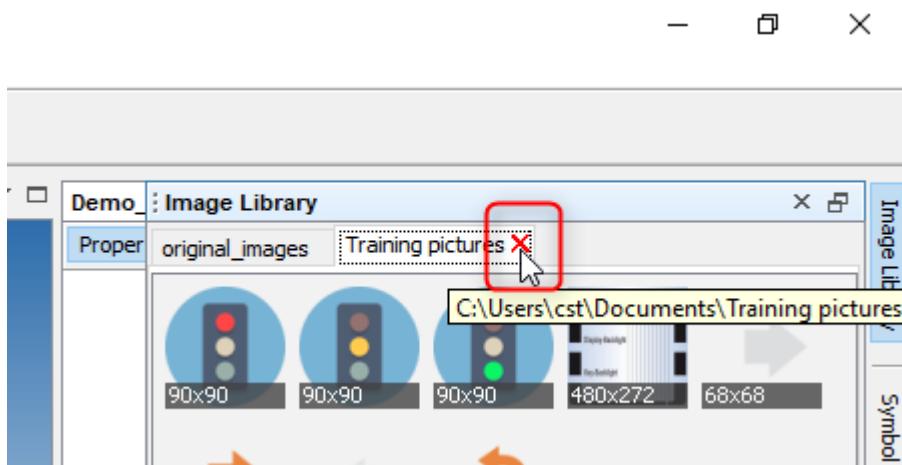
Alternatively, with a right-click on one of the tabs, a context menu with the following options opens
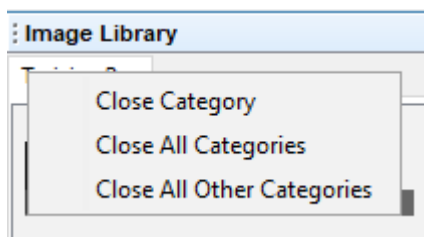
Close Category - closes the tab that was right-clicked on
Close All Categories - closes all tabs

Close All Other Categories - closes all tabs but the one that was right-clicked on

Note that the Favourites and Search Tabs cannot be closed that way. The Search tab closes when the search field is emptied, the Favourites tab is removed when no images are selected as favourites.

Images can be dragged and dropped on objects in the Page Editor or in the project tree. See for details.

The preview image symbols provide a context menu which can be accessed with a right-click.
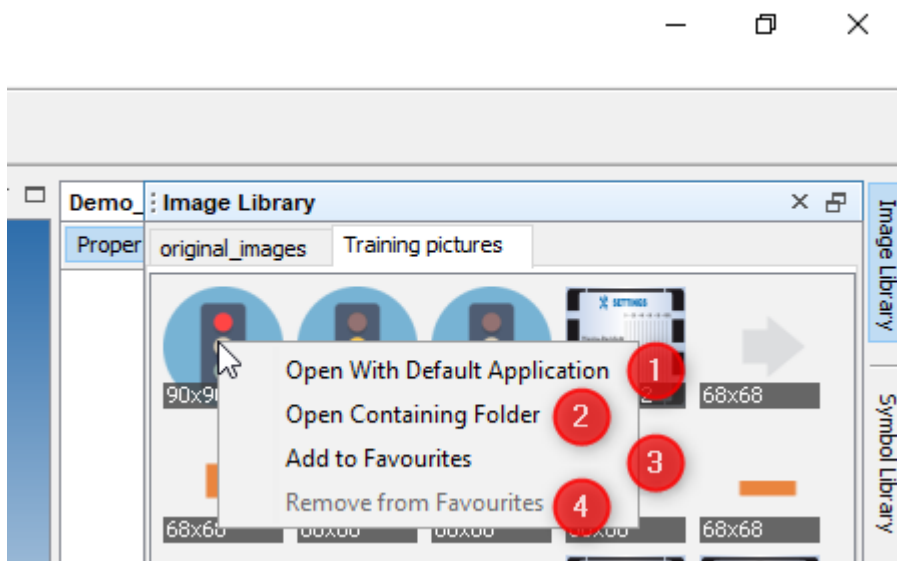
Open With Default Application (1) will open the current image with the application that is set as default in the PC operating system.
Open Containing Folder (2) will open the OS file manager with the current image folder.
Add to Favourites (3) will create a new tab called Favourites, where images can be collected that are used often.
Note that the images are not copied, there is no actual Favourites folder on the file system, the images are linked and only exist in their respective image folders.
To remove an image from the Favourites tab, right-click and select **Remove from Favourites** (4). This entry is only available in the Favourites tab.

The images in the Favourites tab and the opened folder tabs will still be available after closing and re-opening the program.

It is possible to select several images at once, to create a with them. To do this, press the CTRL key and click on the images you want to put in the list.
The images will be marked with a yellow overlay. They also get a number which represents the order in which they will be put into the list.
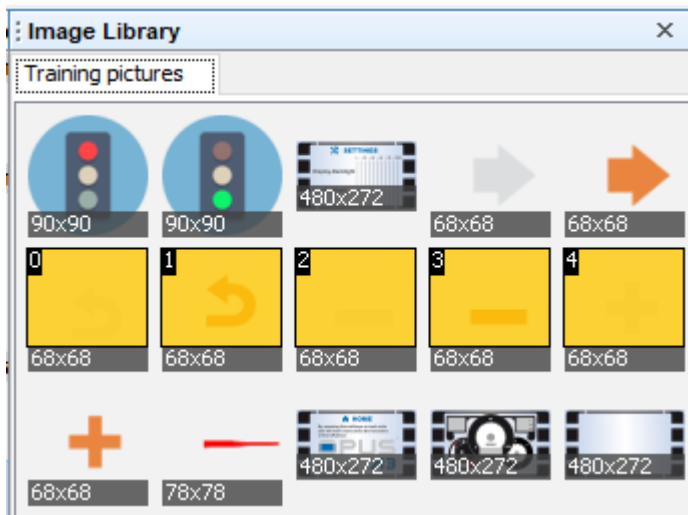
This order can be changed by scrolling the mouse wheel while hovering over one of the selected images with CRTL key being pressed.

This multi selection can also be used to select several images as favourites or to quickly empty (and thus close) the favourites tab.

3.5 Image Pool

The project tree now shows a new node called Images. This node shows all Original Images of the project as well as all Scaled Images of an Original Image.
The new Images Node is shown on the first hierarchy level below the JavaScripts Node.
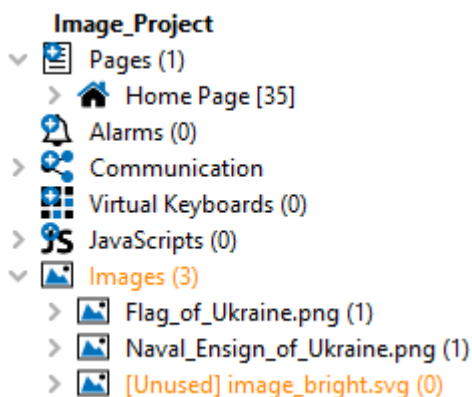
It also shows the number of images in the Image Pool in brackets (3 in the image above).

For each original image in the Image Pool, it shows the scaled images as they are used in the project as its children.
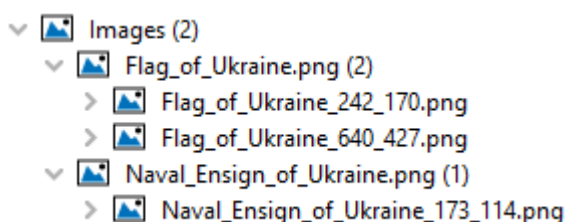
Unused Images

As you can see in the image above, the Image Pool also shows unused images. The image in the example was automatically included when a picture graphic object was added to the project, but when the image was set for the object, the default image was not used anymore.

Unused images can be deleted by right-clicking the Images (n) node and selecting Clean up Original Images...
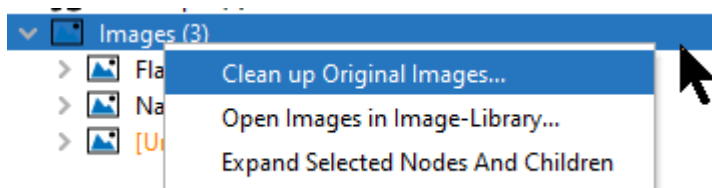
A dialog will appear asking if the following images should really be deleted.
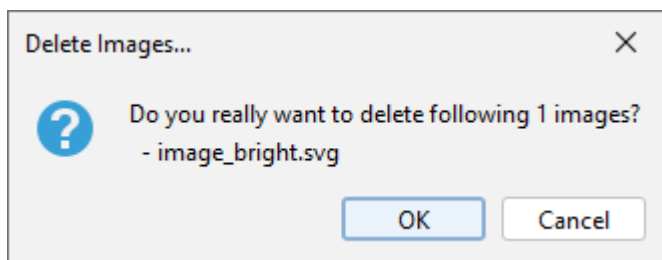
Up to 10 images will be listed in the dialog. If more than 10 images are to be cleaned, a ... indicates that there are more of them.
Press OK to remove the unused images. Press Cancel to cancel the process.

Updating Image Pool

The image pool and all its children will be updated immediately by every image change in the project.

Note that external file name changes are not tracked in the program.
The corresponding node will not work fully as expected, i.e. it cannot be used to put the image on another Picture Graphic object.
When the project is closed and opened again, the original image is reinstated and the manually renamed image will be ignored, i.e. it does not appear in the image pool.
If an image in the image pool should be renamed, it should be renamed within the program.

Renaming images

To rename an image in the program, right-click it and select Rename...
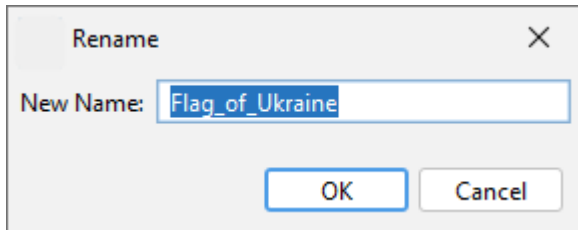
Type the new desired image in the dialog and press OK to confirm. Press Cancel to cancel the process.
All scaled images in the project will be updated.

Note that the image names are case insensitive, i.e. IMAGE.jpg and image.jpg (and Image and iMage and so on) will be treated as the same name and the renaming will not work.

Note that only alphanumerical characters A-Z (a-z), 0-9 and _ and - are allowed as image file names.

Note that only the image name itself can be changed, not the file type suffix.

Note that if the image name is used in JavaScripts, the references are not updated, this has to be done manually.

Updating and Replacing Images

If an original image should be replaced by a different one, i.e. all instances of the original image get the new image, there are three ways to do that.

1) Drag & drop from Windows Explorer
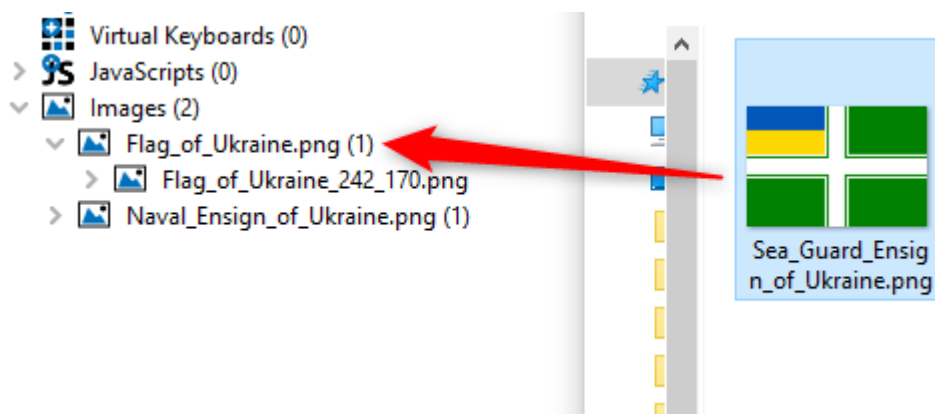
A dialog will be opened asking for confirmation.
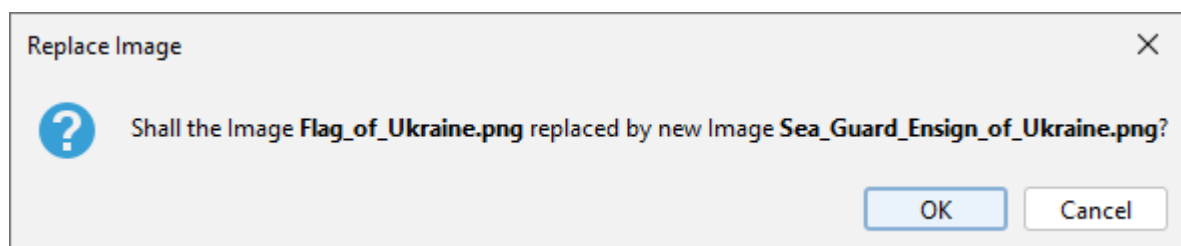
OK confirms the choice, Cancel cancels the process.

2) Drag & drop from the Image Library
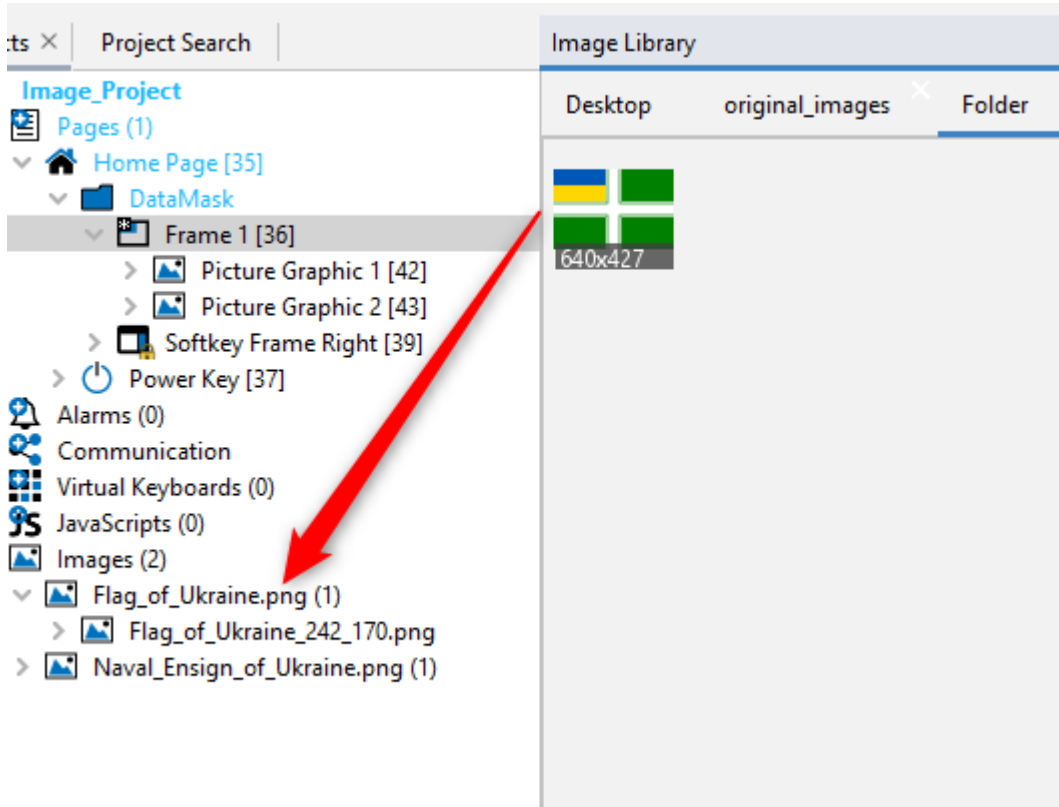
A dialog will be opened asking for confirmation.
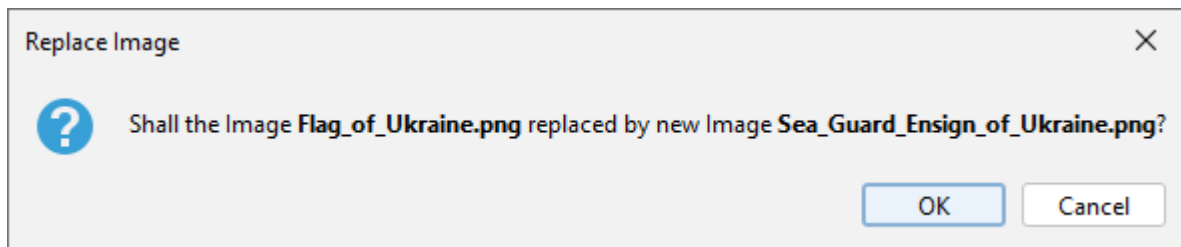
OK confirms the choice, Cancel cancels the process.

3) Context menu

Right-click on the original image and select Replace Image...
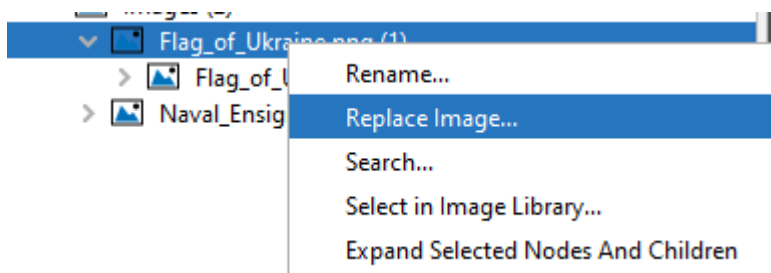
Then select a new image in the file browsing dialog.

Undo / Redo

All operations in the Image Pool can be undone and redone.

Tips & Tricks

If a double click is done on the Images (n) node, or a right-click with the selection Open Images in Image Library..., the original_images folder from the project will be added to the Image Library (which will be opened).

If a double click is done on an original image, or a right-click with the selection Select in Image Library..., the image will be selected in the Image Library.

If a double click is done on a scaled image, or a right-click with the selection Search..., the search is started for this scaled image, so each object that references this scaled image will be listed.

Note that a right-click selection Search... is also possible for an original image. Here the search is started for this original image, so each object that references one of the scaled image of this original image will be listed.

FAQ

What is the difference between the Image Pool and the Image Library?

The Image Library[46] can show the image content from folders on the computer. Images can be dragged & dropped from the Image Library onto objects to use them quickly.
The Image Pool only shows images that are or were already used in the project. It does not show images that are not or were not part of the project.
So an image from the Image Library, that is included in the project one way or another, will put the image into the Image Pool.

If I open the original_images folder in the Image Library, isn't that basically the same as the Image Pool?

No! The original_images folder can still contain images that are not part of the project. Those will be displayed in the original_images tab in the Image Library, but not in the Image Pool.

What can I actually do with the Image Pool?

You can use it as an easy overview of the images in your project. It not only shows you all images you are actually using, but also points to images you are not using.
And it shows you the exact usage of each image, i.e. where it is used in the project and at which resolution(s).

Can I just leave the Image Pool be and not use it at all?

Sure. You will probably never have the necessity to use the Image Pool in any way. It is a tool for convenience, but you can build projects and do everything you need without using it. We hope, you will use it, though.

I have the same image with two different file names - why do I only see one?

The Image Pool calculates a check sum of the image. It will find that the two images are identical and it will only show the image that was first in the Image Pool.
On the other hand, since the images are the same anyway, it wouldn't make sense to have two of them in the project.

3.6   Import Variables

This chapter describes the process to import variables. The dialog can be reached via the menu Communication -> Import Export Variables -> Import Variables

Open File For Variable Import
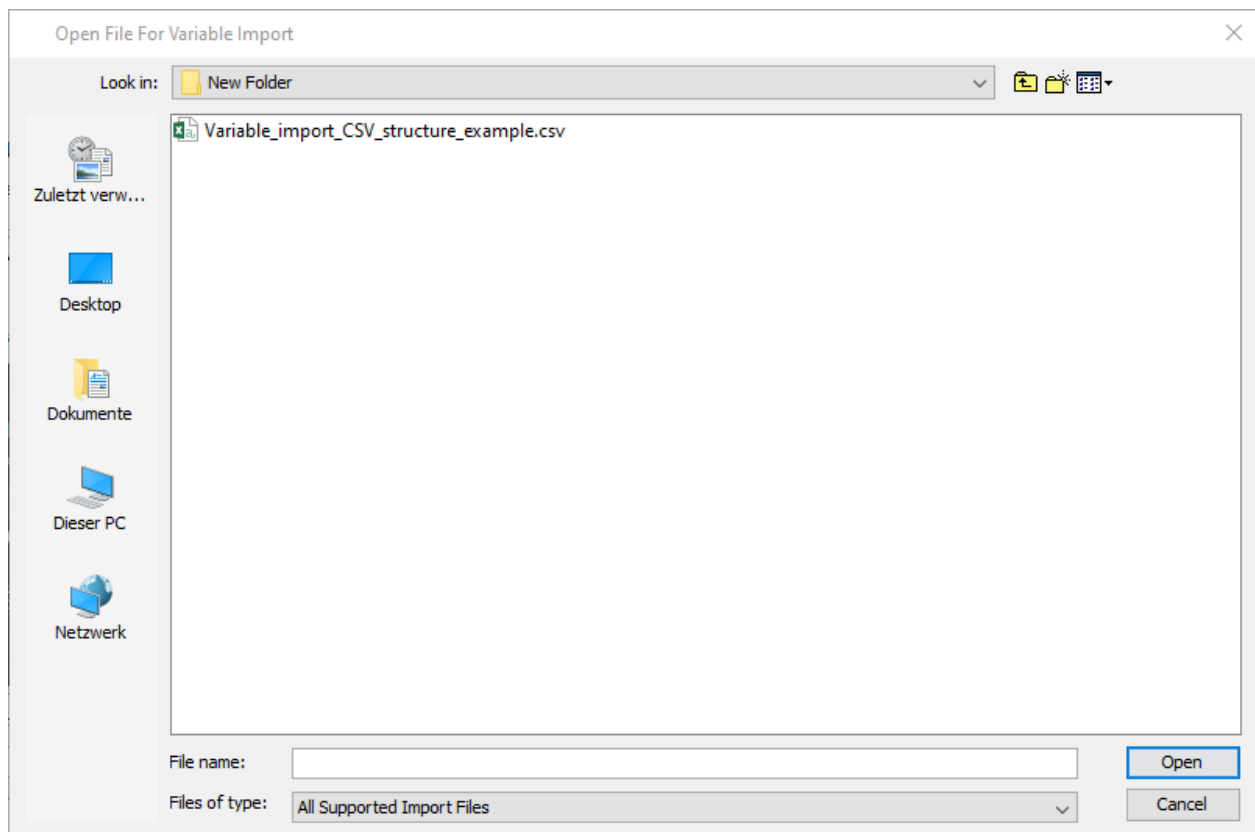
In the first dialog, a file can be chosen to import variables from. This can be
- a project.xml file from another project (located in the sub-folder \terminal_files of the project folder)
- a .CSV file containing variable information

Project.xml import

When variables are imported from a project.xml file, the Imported Variables step will be next.

CSV import

In this dialog the structure of the CSV file has to be specified.
- Enter row no. to start: Select the row in the .CSV file where the variable information starts.
- Header Row is available at 1: If the .CSV file contains a header, check this box. The row to start should include the header row
- Select delimiter: Select the delimiter used in the .CSV file.
- Map Columns: Match the columns of the .CSV file with the necessary variable properties. Press Default to revert the mapping to the default setting.

The default columns for the .CSV file are:
Variable;Index;Sub-Index;Owner;Group;Property_Description;Data-Type;Length;Access Type;Min Val;Max Value;Default Value;Initialize Value As Invalid;Remanant;Force writing

Press the button Next... to continue. Press Cancel to stop the process.

Imported Variables

In this dialog it can be specified which variables from the source should be imported.
In the Tree view you can select the variables or variable groups for import.
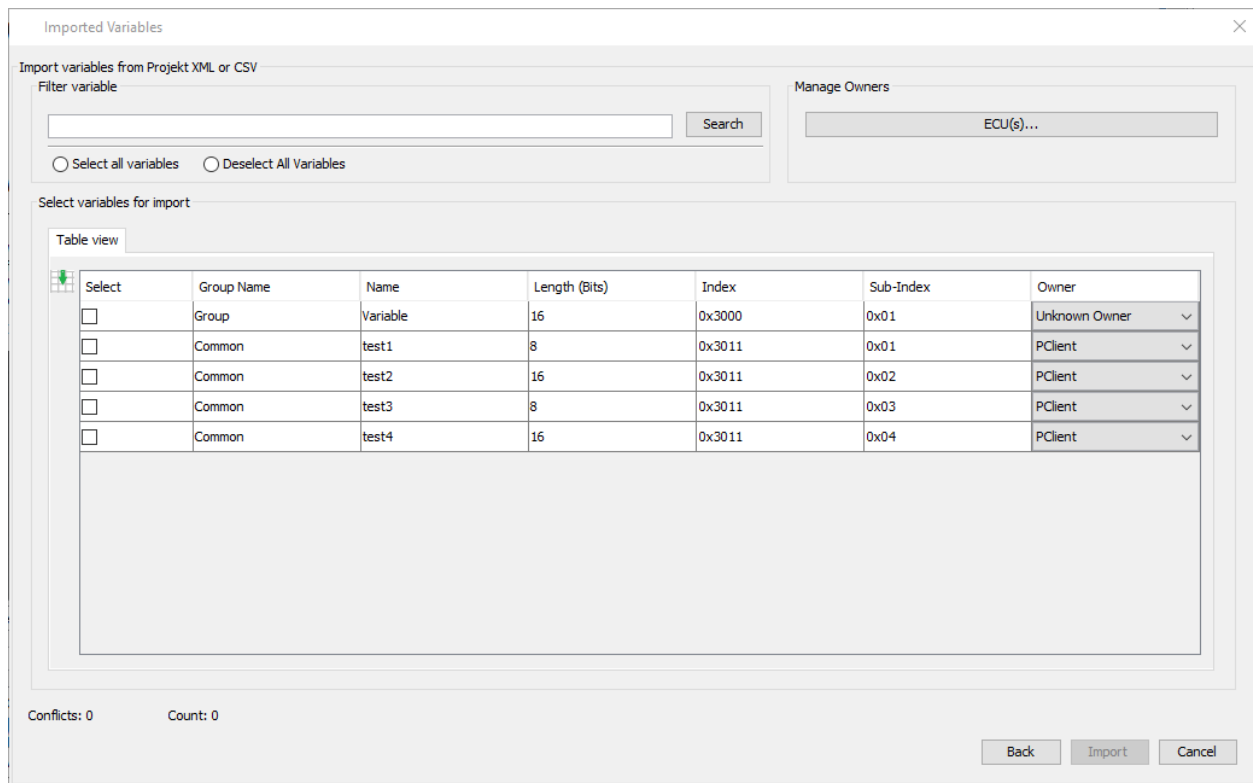
In the Table view tab you can see an overview of the variables with the most important properties.

In the Filter variable section you can search for variable names live.
With the "ECU(s)..." button you can open the dialog to add ECUs / owners to the project.

Additionally you can modify the owner that the variables are assigned to.

You can select variables by checking the according checkboxes.
Click the "Select all variables" radio button to select all available variables for import.
Click the "Deselect all variables" radio button to deselect all variables for import.

By clicking the "Import" button the variable(s) will be imported.
If there are conflicts with existing variables in the project, those conflicts will be shown in a dialog and the import process will be stopped. Solve the conflicts and restart the import process.

3.7   Launcher

Before the program starts, a launcher is started which gives a number of options for the program start.
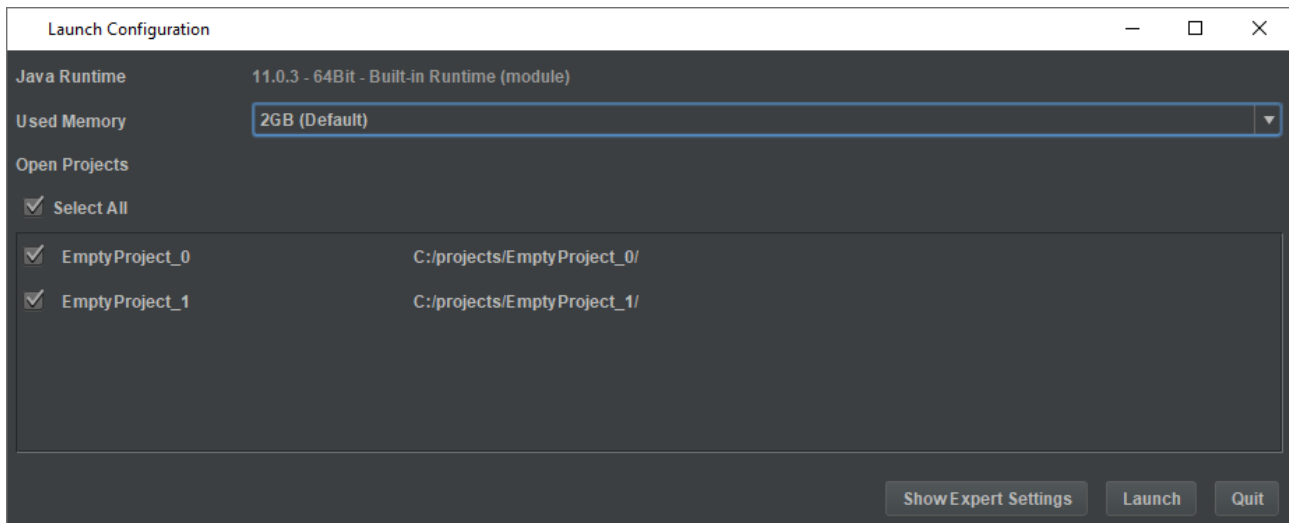
If nothing is done, the launcher automatically starts the program after a countdown of 5 seconds. Click anywhere in the launcher to stop that countdown.

In the first line, the used Java Runtime is displayed. This cannot and should never have to be changed.

In the second line, some heap memory settings for Java can be selected.
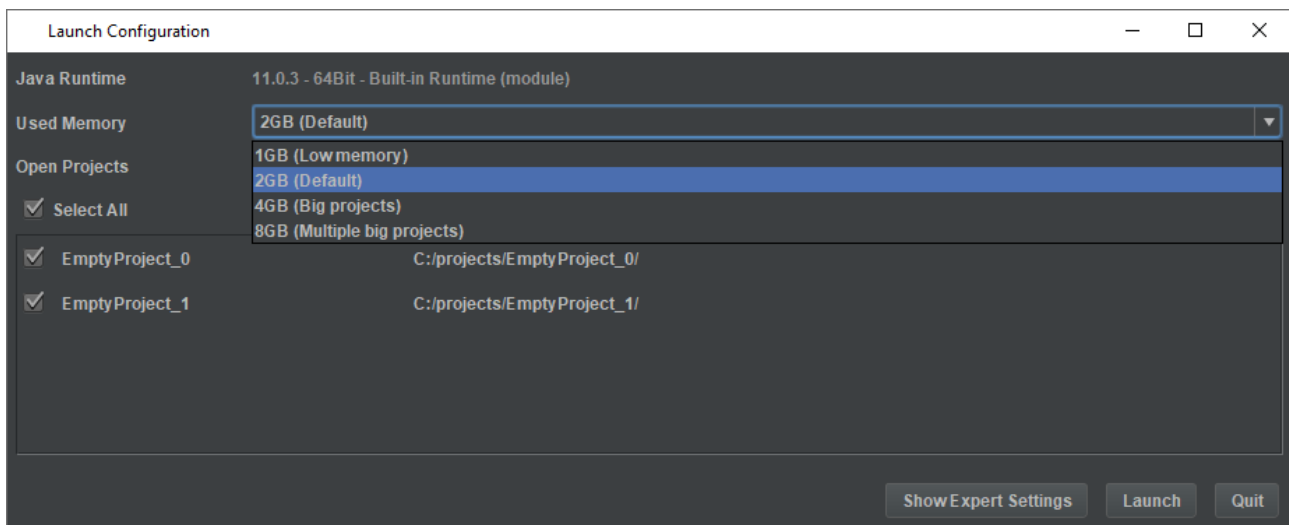
The default setting of 2GB of RAM memory is sufficient for most cases. If limited RAM is available, 1 GB can be selected. This can be problematic with large projects, through.
For larger and multiple large projects, especially when upgrading them from older versions, 4 and 8 GB can be selected. Please note that enough RAM memory needs to be available for those options to make sense.

The projects that were opened last time the program was running, is listed next. All projects are selected by default. Projects that should not be opened again can be deselected. With the Select All box all projects in the list can be selected or deselected. Please note that once the program starts with projects deselected they will not appear in this list again. Instead, they will be listed in the recent projects pane in the [start page](#)[134].
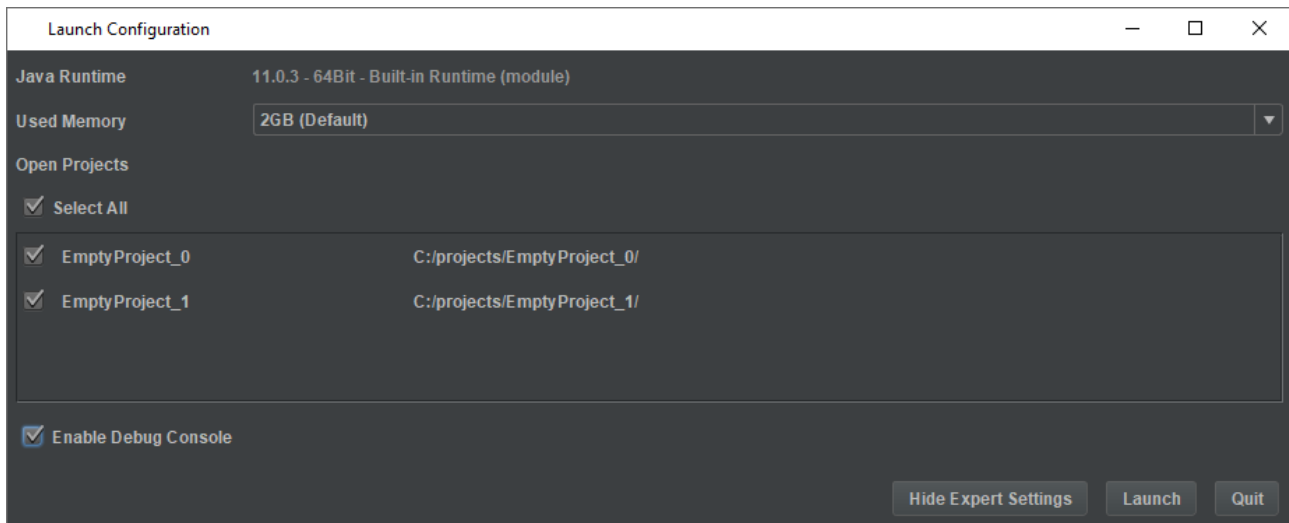
The Show Expert Settings button shows a check box to run the program with a debug console. This is mostly helpful if something goes wrong repeatedly to get more information on an exception or crash.

3.8   Licensing

 Tutorial video about how to get a license
https://www.youtube.com/watch?v=LsO0v44J6YE

The program has to be licensed to be fully usable.

In the current version, the demo has the following limitations:
- Only up to 2 pages can be created
- Only up to 5 variables can be created

Projects that exceed these limitations can be opened, modified and saved, but they cannot be downloaded to the device / USB stick.

The license can also unlock all >2400 ISO symbols [135] to be used in the program, the DBC file import feature [381] and the Ethernet Camera Tool [790]. These licenses are sold separately.

To get a license, three steps are necessary:

1. Create a key file

Open the menu Tools -> Generate Key File
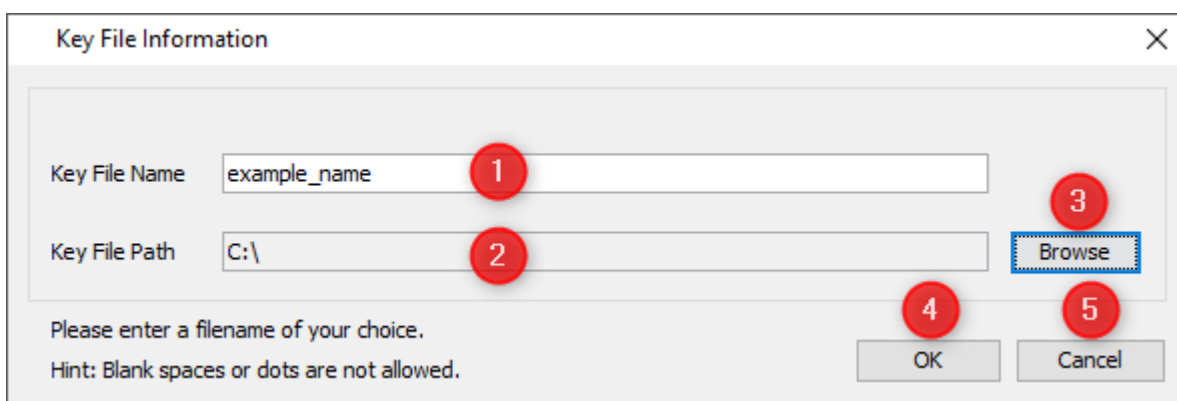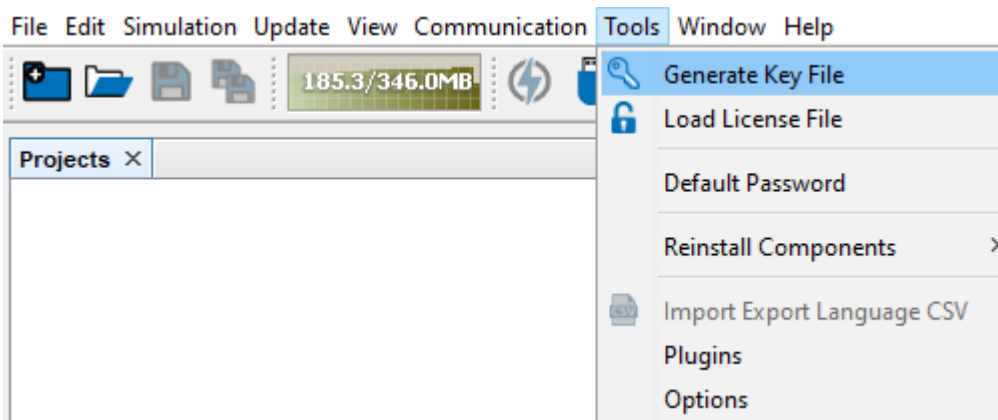
In the dialog, choose a filename (1) and select where the key file should be stored (2) by pressing the Browse button (3) and choosing a path.
Confirm and close the dialog to create a key file with OK (4). Cancel (5) cancels the process and closes the dialog.

2. Get a license file

To get a license file, first the license needs to be purchased from your vendor. Then you can send the key file to your vendor and they will send you back a license file.

3. Load the license file

After the license file was received, open the menu Tools -> Load License File
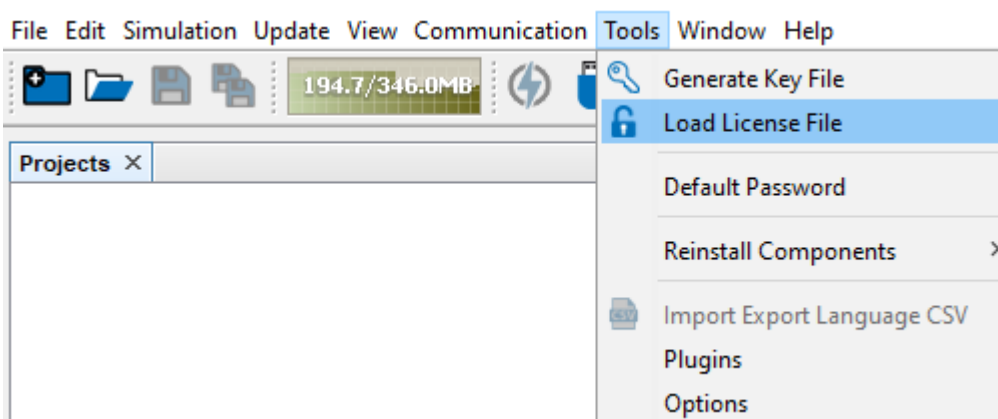
In the dialog, browse to the location of the license file, select it and confirm by clicking Select License File.

A confirmation dialog is shown. The dialog will also inform if one or more of the additional features were unlocked.
Close the dialog by clicking OK. Now the program is fully licensed.The title of the program in the title bar should have changed from Demo to Licensed.

3.9   Options

This chapter describes the available options that can be reached via the menu Tools -> Options.

General
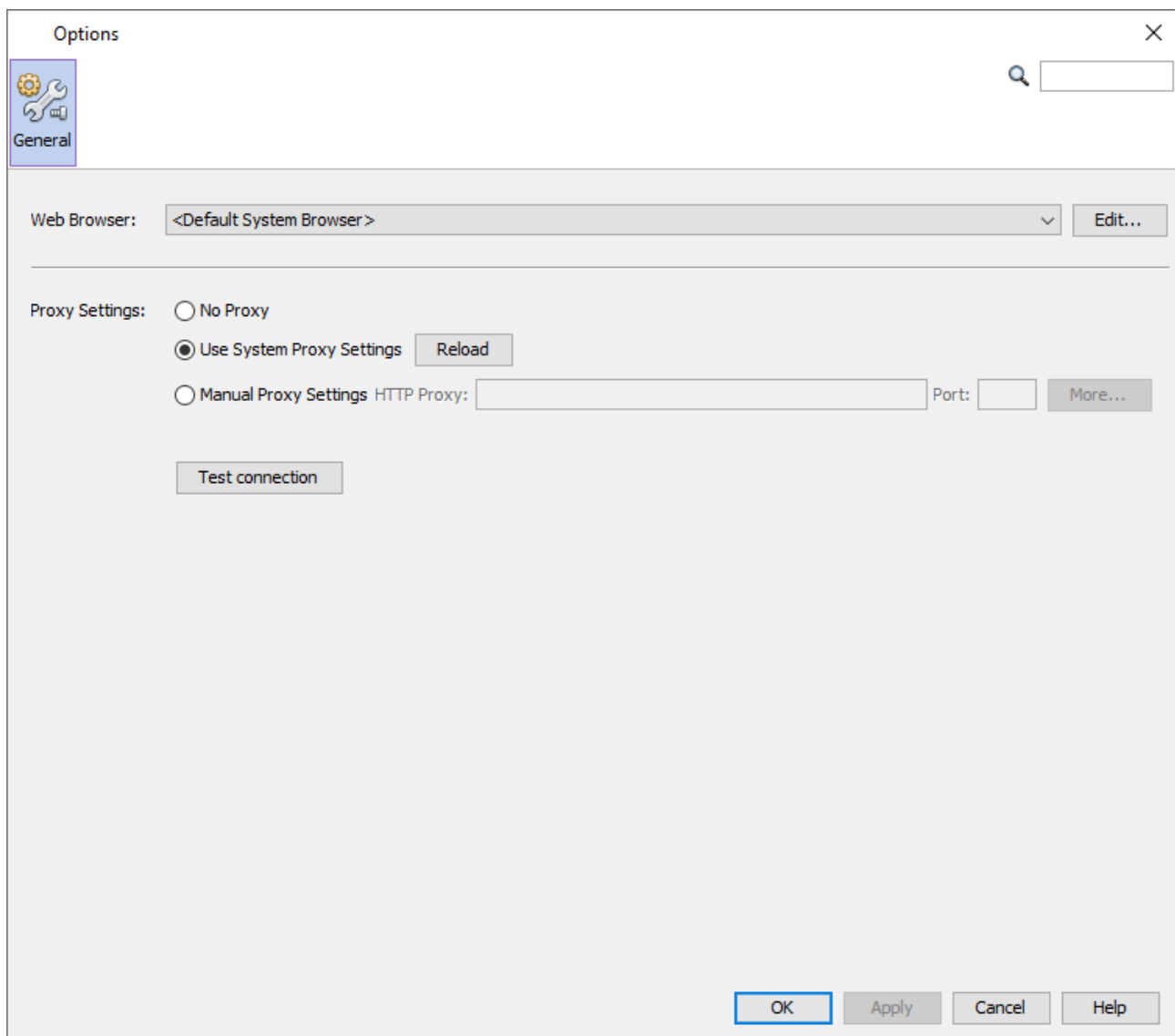
In the General tab the Web Browser used to open this HTML help can be selected. Additionally, proxy settings can be made for the online update[660] and the report [730] connection.

Editor
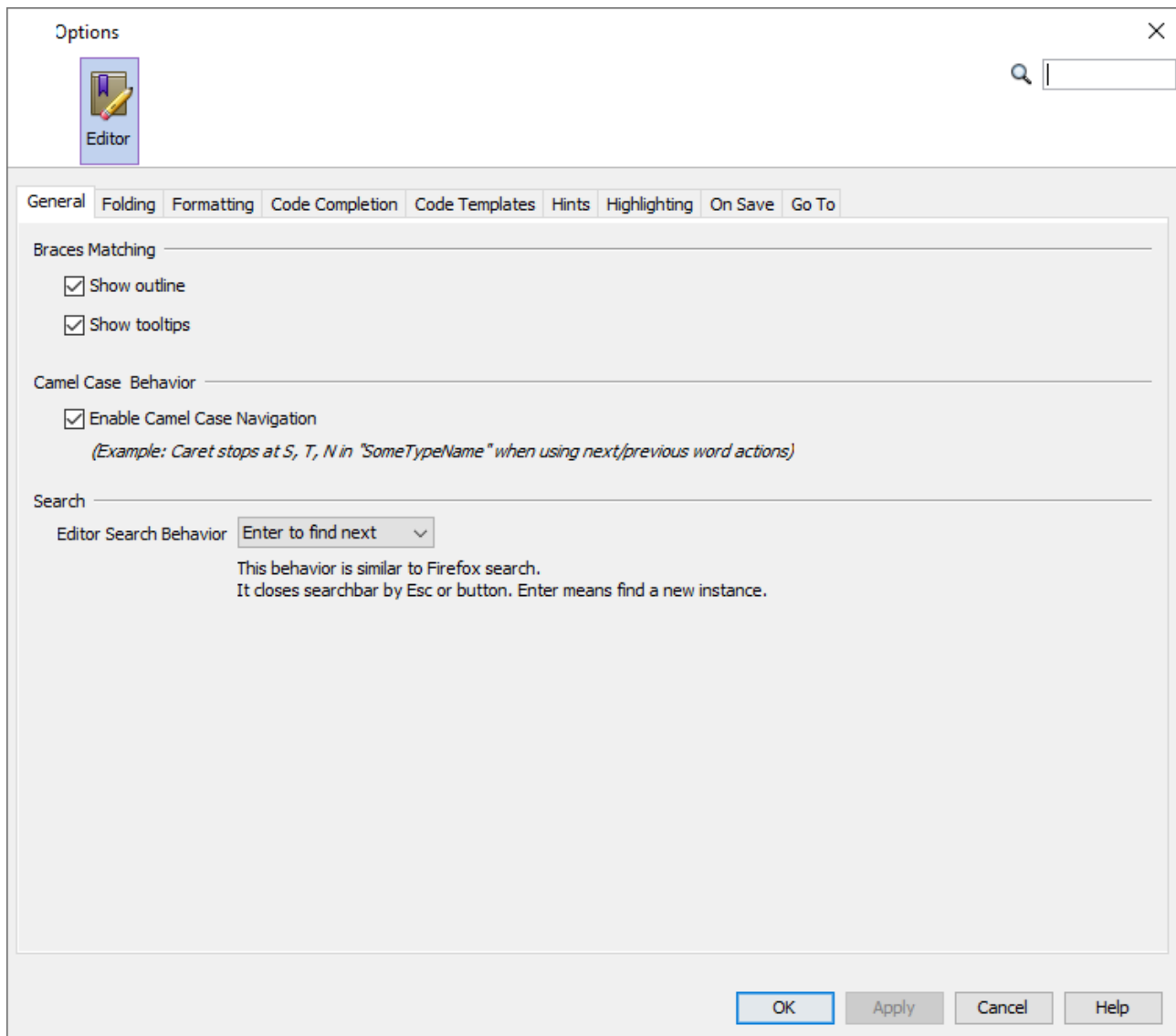
In the Editor tab the behavior of the JavaScript editor can be modified in great detail. Please refer to a Netbeans documentation for details, e.g. here.

Fonts & Colors
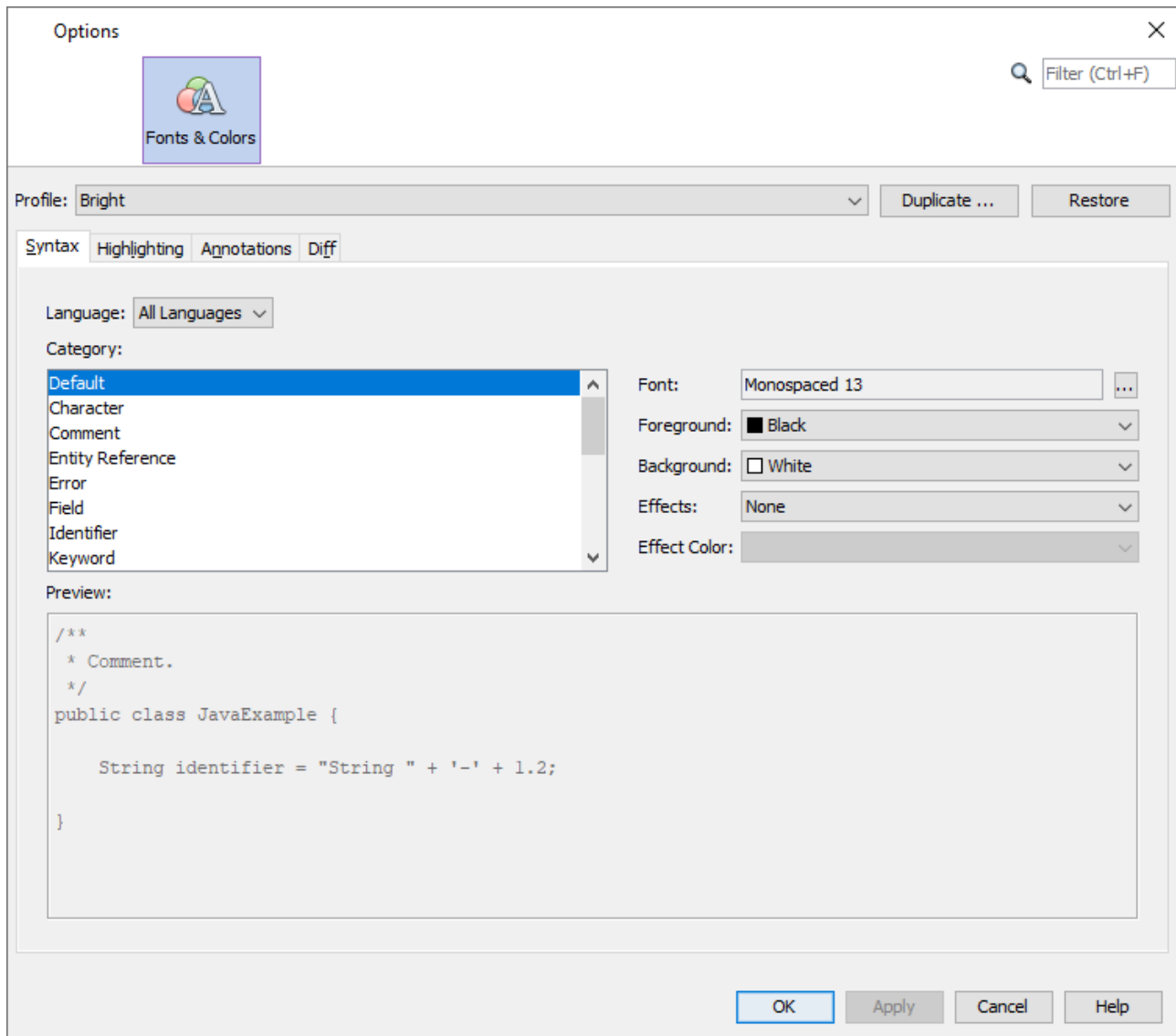
In the Fonts & Colors tab the appearance of the JavaScript editor can be modified in detail. Please refer to a Netbeans documentation for details.

Scene
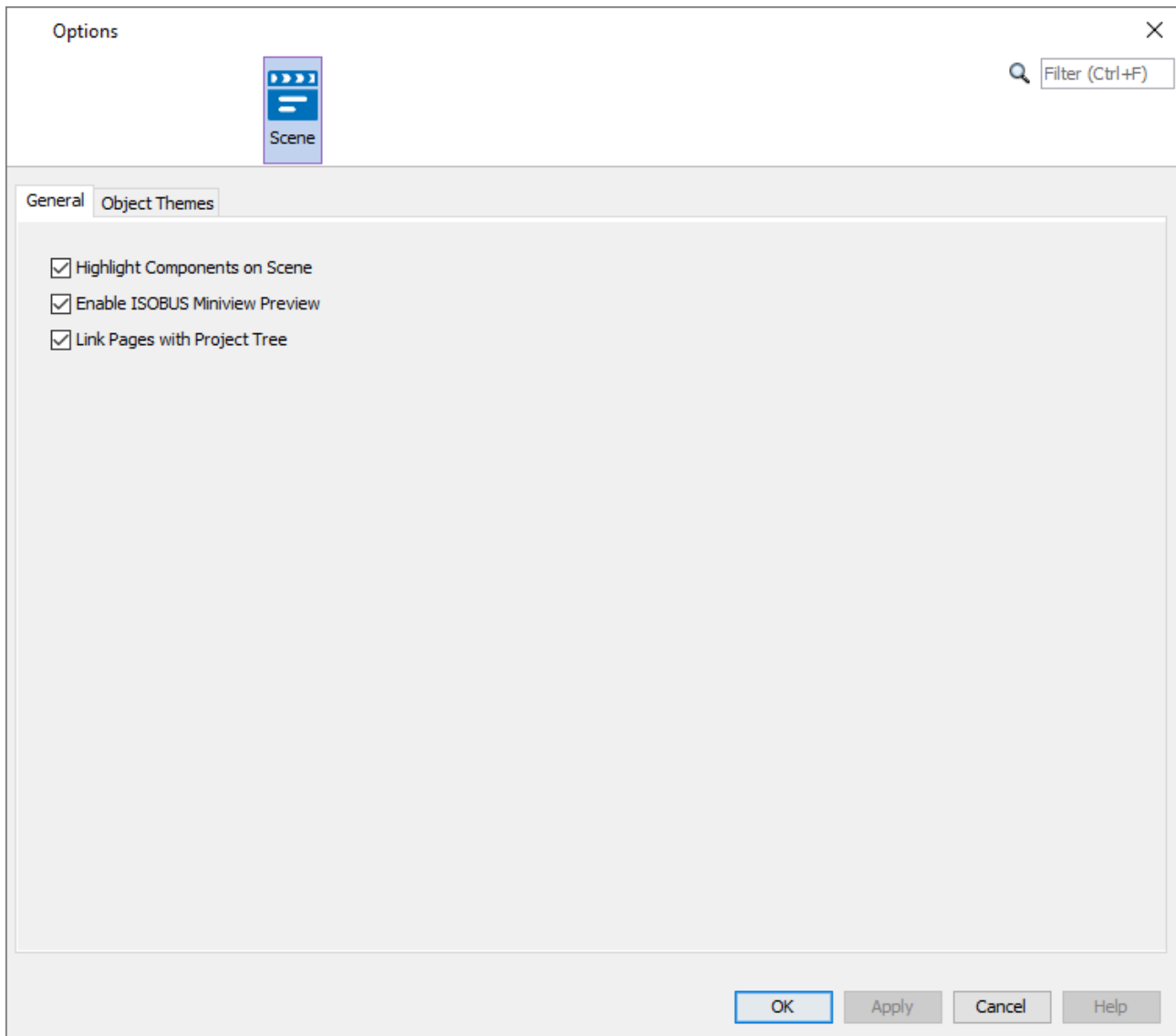
Highlight Components on Scene - Option to highlight components in the page editor when the mouse hovers over them.

Enable ISOBUS Miniview Preview - Miniview Preview in pages for ISOBUS projects.

Link Pages with Project Tree - When a tab is closed, the next open page will be selected in the tree with this option active.

For a description of the Object Themes please refer to the Object Themes [580] chapter.

Appearance
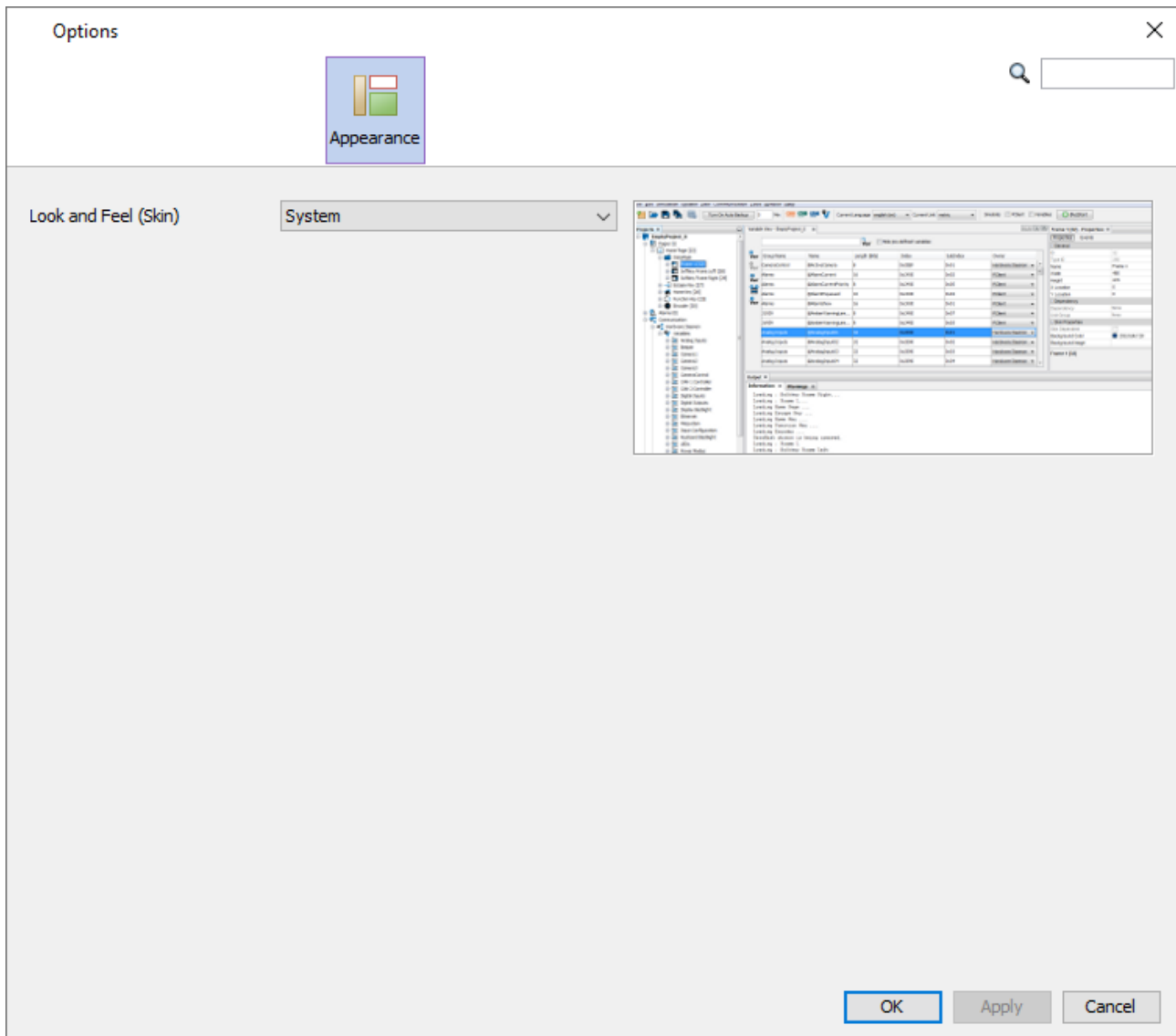
In the Appearance tab one of the available skins can be selected. The skins will be shown with a preview image.
Note that the program has to be restarted for the new skin to be activated.

Java Runtime
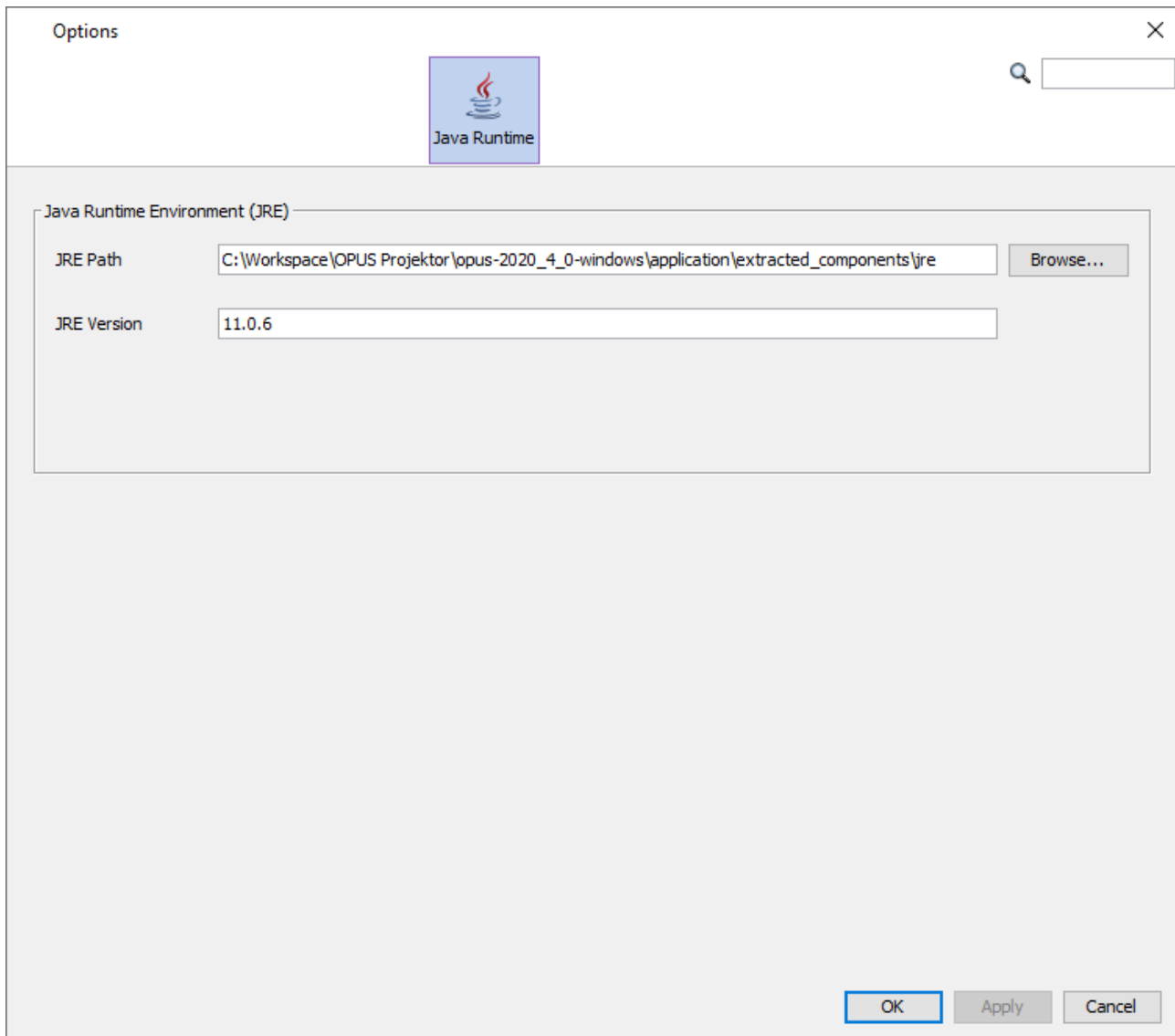
In the Java Runtime tab the Java Runtime that will be used by the program can be selected. Normally this never has to be changed, as the program comes with its own Java runtime.

Projects
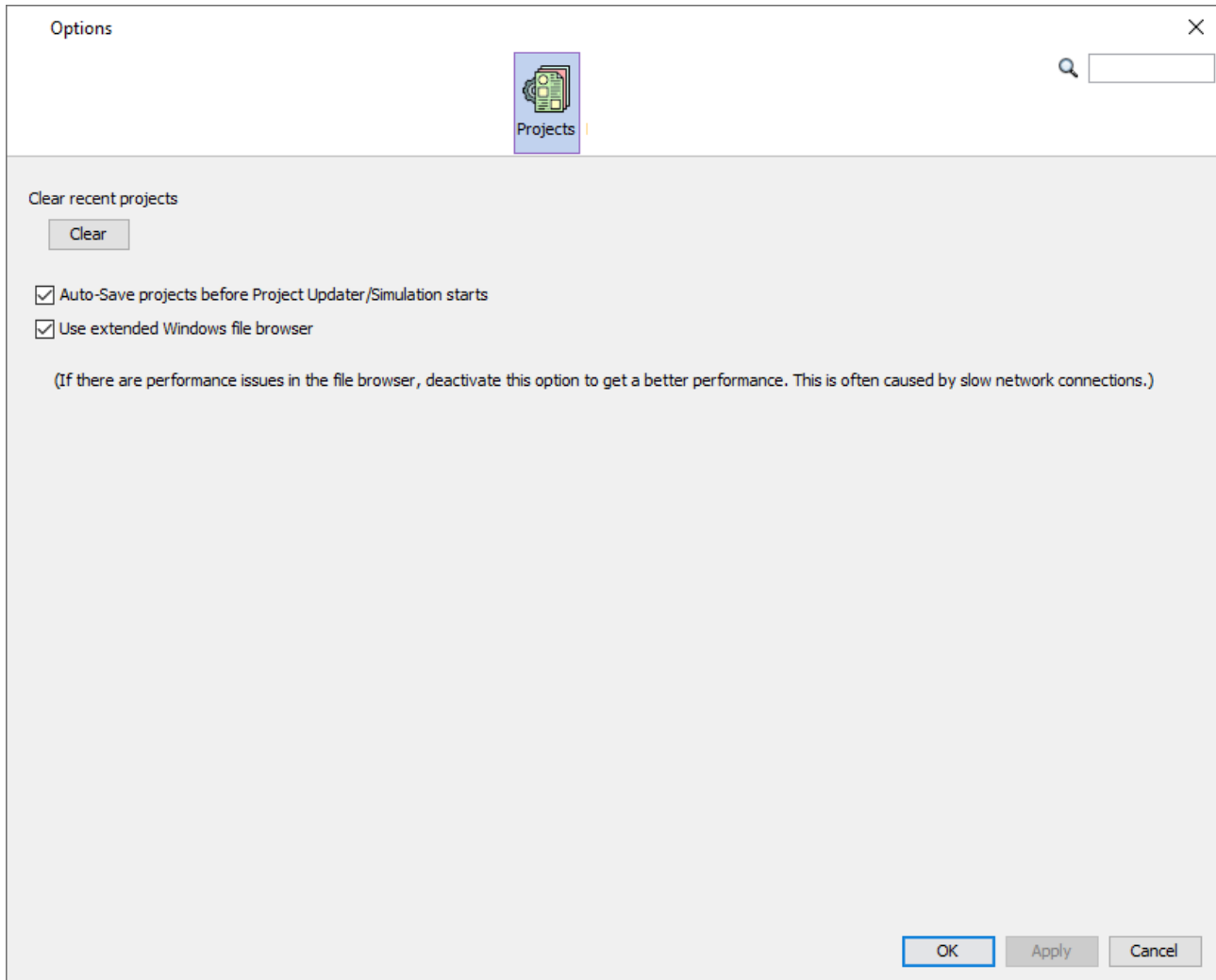
In the Projects tab the list of recent projects in the Welcome Page can be cleared.

It is also possible to enable/disable the auto-save of the current project when the Updater or the PClient simulation is started.

Lastly, the extended Windows file browser can be disabled. This will limit the view in file browsers to "My computer" and the current directory path. With the option enabled, Windows Explorer favorites and all drives including network drives will be shown. Disabling the option might help with a sluggish folder navigation.

Reporting
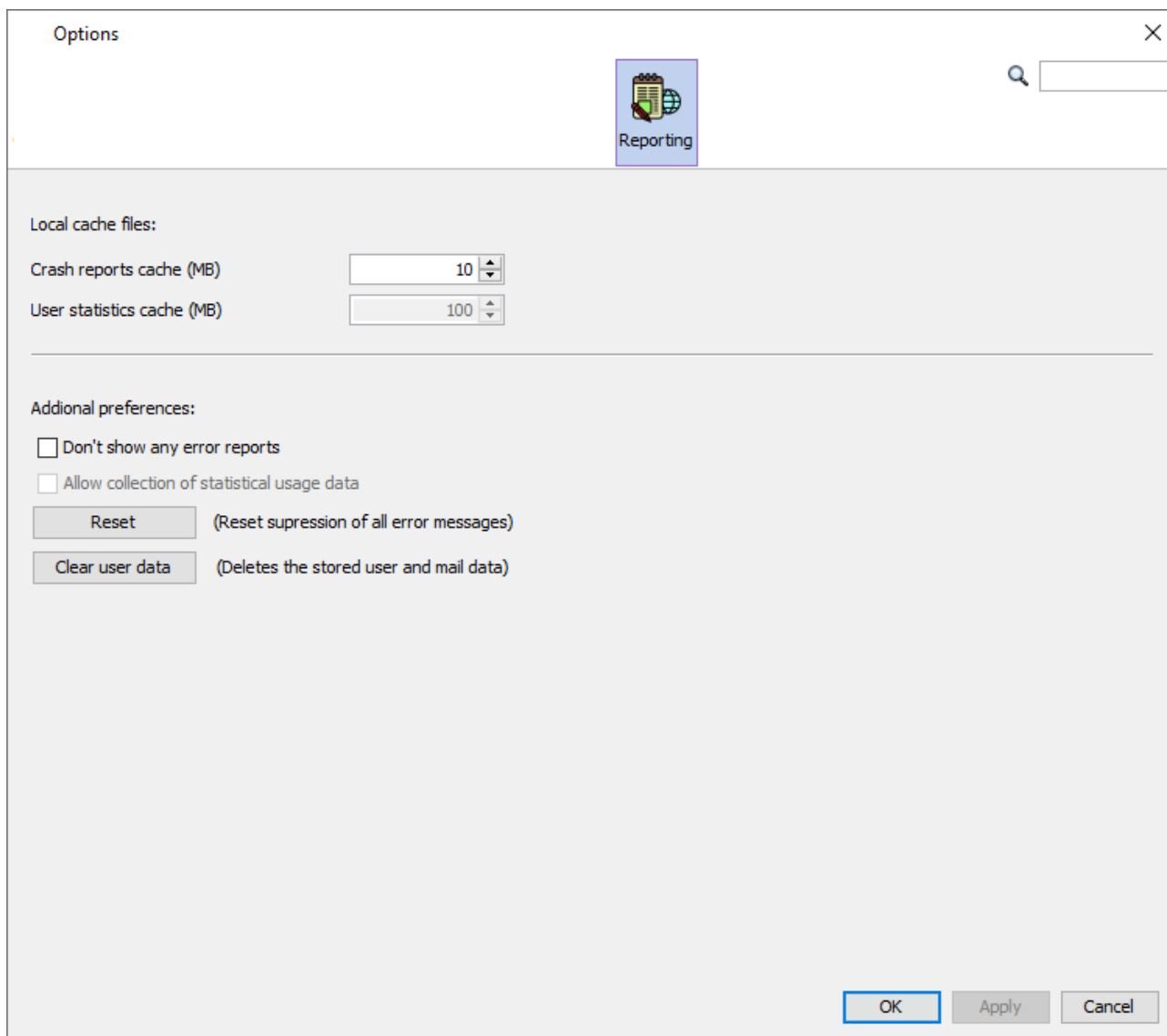
In the Reporting tab settings regarding the reporting feature[730] can be made.
The cache size for unsent reports can be set.
A general setting to not send any reports can be made here (which we don't recommend).
Also, the supression of single exceptions which can be set in the crash report dialog, can be reset here.
Lastly, the stored user data for reports, in form of the name and email, can be cleared.
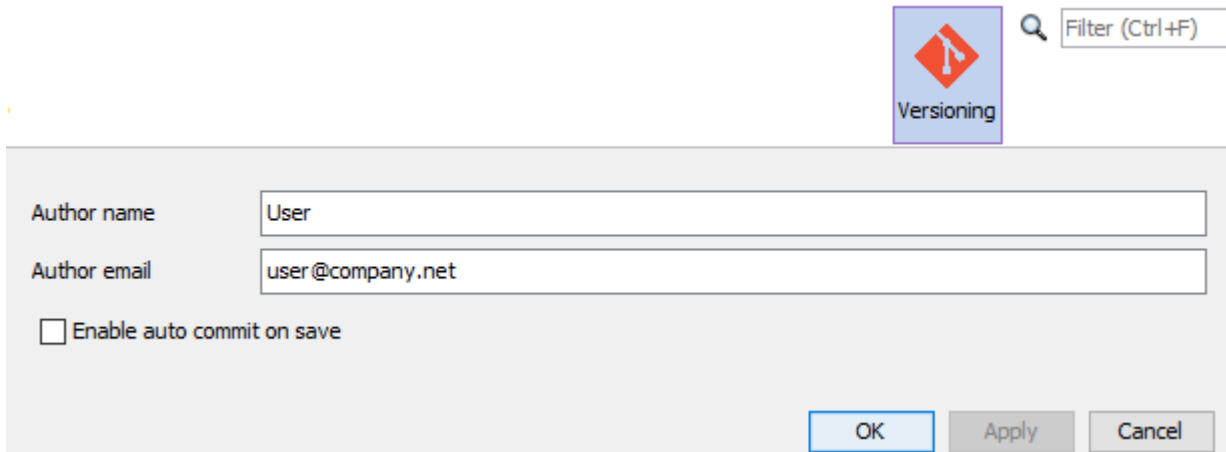
Reporting

The author fields will be used as committer information for each project commit.
Enable auto commit on save means that the commit dialog[127] isn't shown when saving the project.

3.10   Page Editor

The Page Editor is the main window to design graphical content for the project.
It has its own toolbar:

Menu	Description
	Enlarge view in editor.
	Reduce view in editor.
	100% scaled view in editor.
	Left alignment of multiple selected objects.
	Right alignment of multiple selected objects.
	Horizontally centered alignment of multiple selected objects.
	Top alignment of multiple selected objects.
	Bottom alignment of multiple selected objects.
	Vertically centered alignment of multiple selected objects.
	Move the selected object one step closer to the background (changes z-order).
	Move the selected object one step closer to the foreground (changes z-order).
	When resizing an object, child objects are not resized.
	When resizing an object, children are resized too.
	Switch "Snap to object" on / off
	Shows/hides a horizontal and a vertical ruler.
Grid Width: 10 px	Allows configuration of the grid width. Furthermore snap to grid can be enabled and disabled.
	Shows a list of the last undo steps. Disabled for this version.
	Shows a list of the possible redo steps. Disabled for this version.

Objects can be created by dragging and dropping from the [object palette](#)[412] or by using [copy & paste](#)[42] from other pages or projects.

Objects can be moved by modifying the X and Y position properties or by moving them with the mouse.
Object swim lanes help positioning objects at the correct distances:
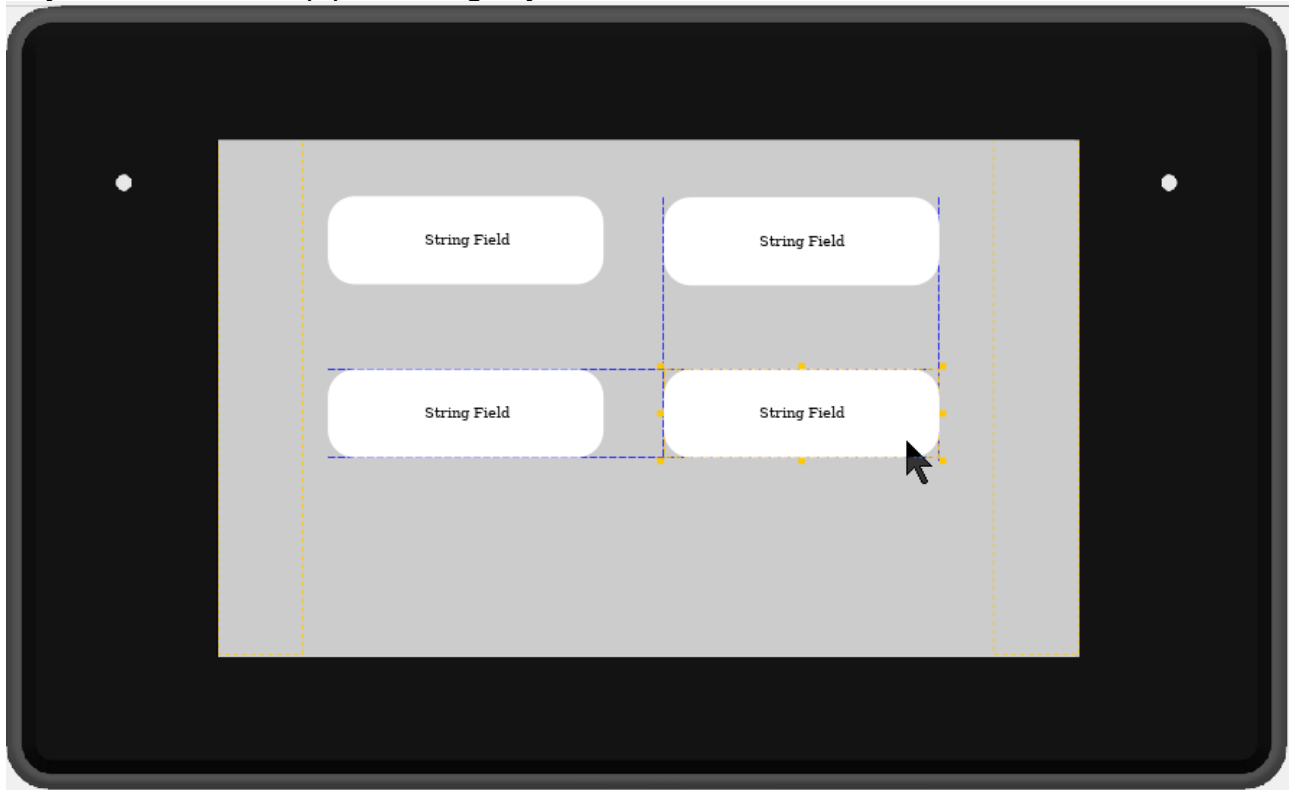
Those can be turned off with the Snap to Object switch in the Page Editor toolbar.

Objects can be resized by modifying the Width and Height properties or by using the orange squares in the corners and edges of the object.
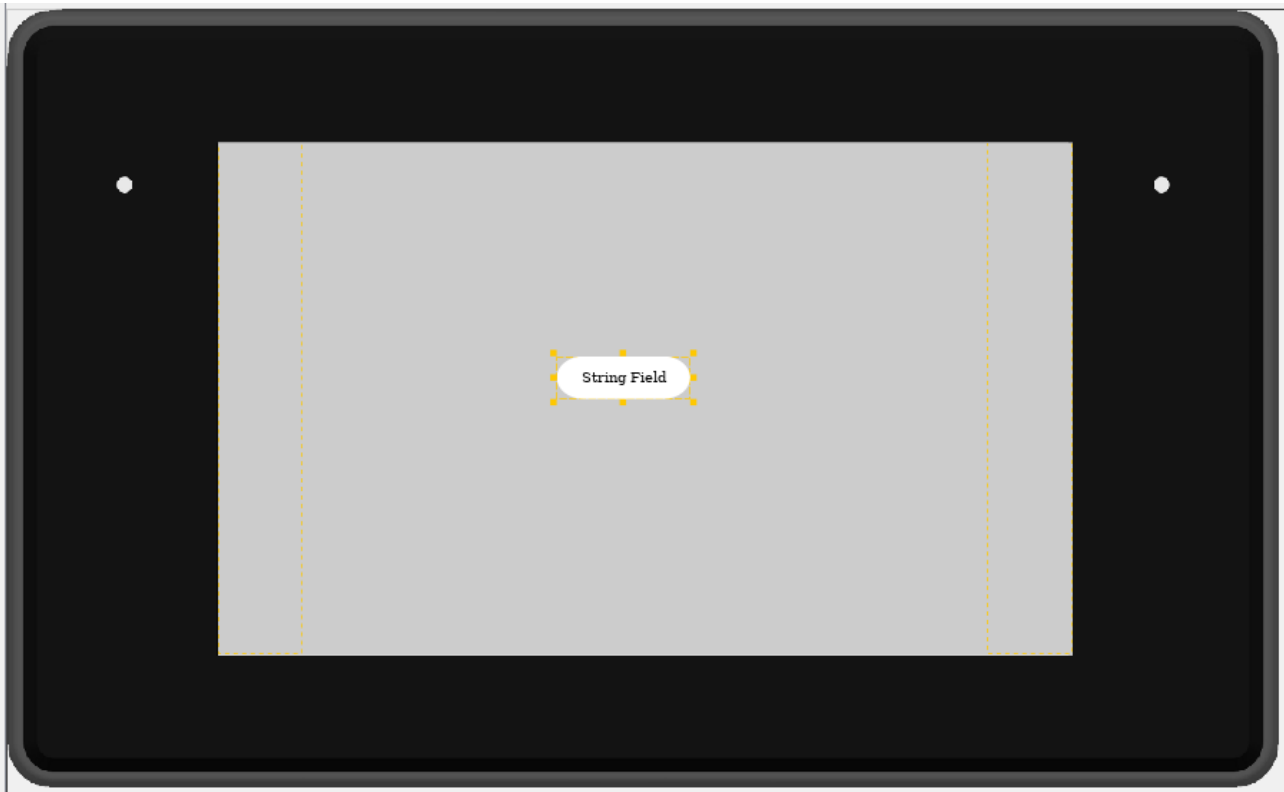
To resize an object and keep the aspect ratio (especially interesting for images), press the Shift key before clicking one of the corner squares or while resizing.

Objects can be locked in place to avoid accidental modifications with the mouse while editing other objects.
To lock an object, right-click on the object and select *Lock Position and Size -> Lock Object.*
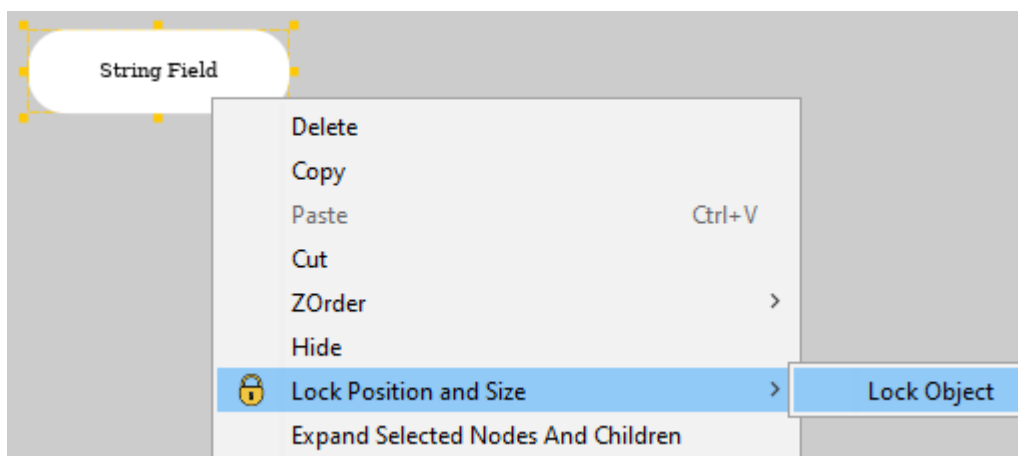
If the object is already locked, the menu will show *Unlock Object* instead.

Objects can also be locked / unlocked in the project tree.
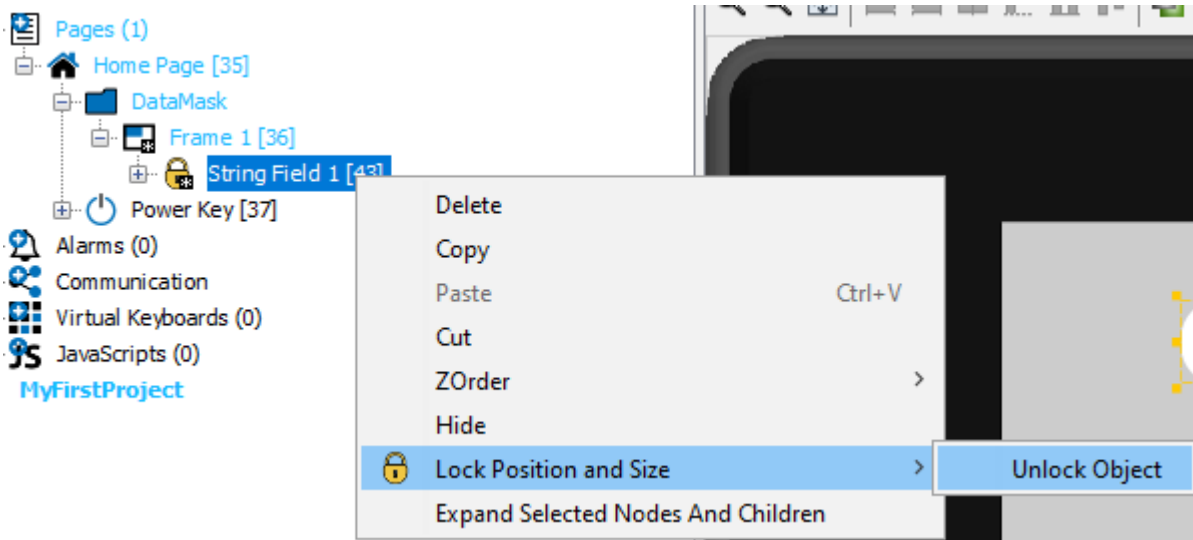
3.11 Project Properties

This chapter describes the various properties of a project.
The project properties can be opened in two ways:

- Using the **Properties** entry of the right click context menu of the project node in the **Projects** tree window.
- Using the **Project Properties** entry in the **File menu.**

Please note that not all properties are available at all times. Only the properties available for the device in the currently selected project are shown.

General Properties
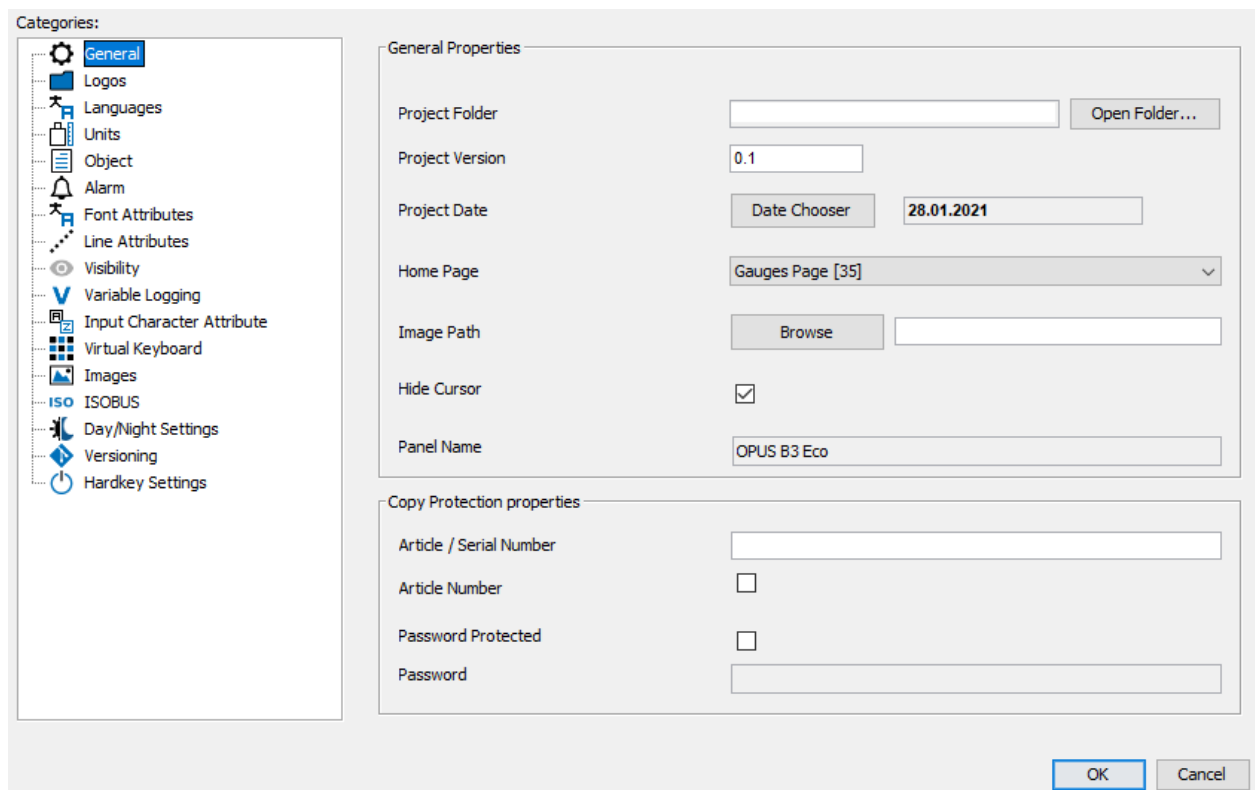
Project Folder

Shows the absolute path to the project directory.
Please don't change the project directory name outside of the program. This will not change the project name. Use "Save As" or "Rename Project" to save the project under a different name. This will also change the project path to the same name. Next to the path is an "Open Folder..." button which opens the OS file explorer at the location of th eproject.

Project Version

The version number of the project can be set here. The version can be displayed in the project with the variable @ProjektVersion[174] .

Project Date

The project date (e.g. release date) can be set here. This date can be displayed in the project with the variable @ProjektReleaseDate[174]

Home Page

The selected page will be used as project home page. The project home page is the page which opens when the display starts. The Home Page can also be set via context menu in the project tree.

Image Path

Here the default path for images can be set. Whenever you open an image browser for opening an image, the browser starts using this path.

Hide Cursor

On display devices running X-Server, the mouse cursor is hidden when this property is enabled.

Panel Name

The panel name of the project is displayed.

Copy Protection Properties

Note: This feature only works on supported devices!
The copy protection feature protects your project from being copied from one device to another. It prevents hackers from running your project on other devices.
The copy protection uses the serial or article number of the display to determine if a project is allowed to run on the display.

Article / Serial Number

Enter the serial or article number your project is allowed to run on.
Note: The serial number is device specific. When using a serial number, you have to update your project for each device the project shall run on. Using the article number, the project runs on all devices having the same article number. Note that this only makes

sense if you have a specific article number, otherwise the project will run on all devices with the same article number.

Article Number

If you enable this property, the [Article / Serial Number](#)[76] is treated as an article number. When disabled, it is treated as a serial number.

Select Copyright Protection Image

You can customize the image that is shown if a copyright violation is detected.
- Select an image file for using a customized image.
- Deselect the customized image and use the default copy protection warning.

Password Protected

To prohibit "hackers" extracting your project from a device and changing it (e.g. remove the copy protection), you can enable the password protection.
If password protection is enabled, you have to enter the password whenever you try to open the project.

Note: It is highly recommended to use password protection when copy protection is used!

Password

If Password Protected is enabled, you have to enter the password to use here.

Logos Properties
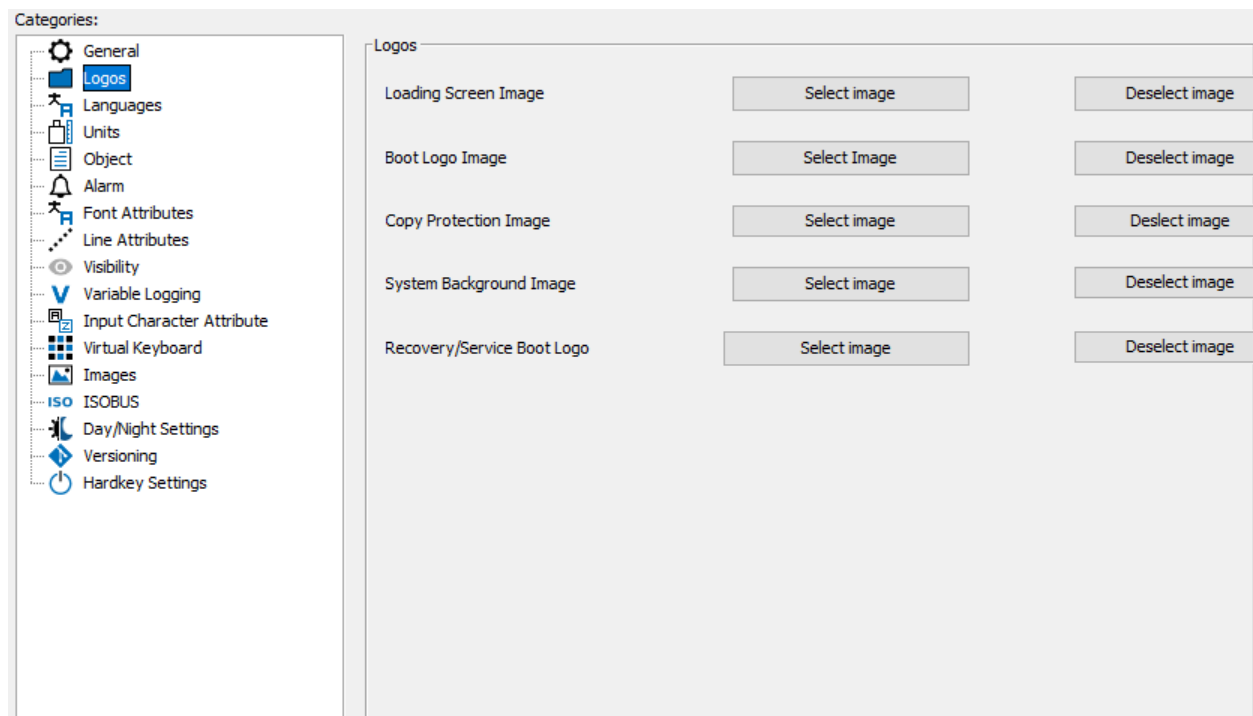
Note that not all image settings are available on all devices.

Loading Screen Image

The loading screen image is an image that will be displayed while the project is being loaded on the terminal. Showing a progress bar is possible.
The loading screen image should have the same resolution as the target device display and must be an image in PNG file format.
When selecting an image, the image will be renamed to 'loading_screen.png' and placed in the terminal_files/additional_files directory in the project folder.

How to Show a Progress Bar

To create a progress bar you only need to make parts of the loading screen transparent. the progress bar will become visible automatically. The only constraint is that the progress bar runs from the left to right of the display. It starts at 10% of the display resolution and ends at 90% of the display resolution. Example:

If your screen resolution is 480x272, the progress bar starts at
X = 48 = 480 * 0.1
and ends at
X = 432 = 480 * 0.9
You are free to define any height and vertical position of the progress bar on the screen.
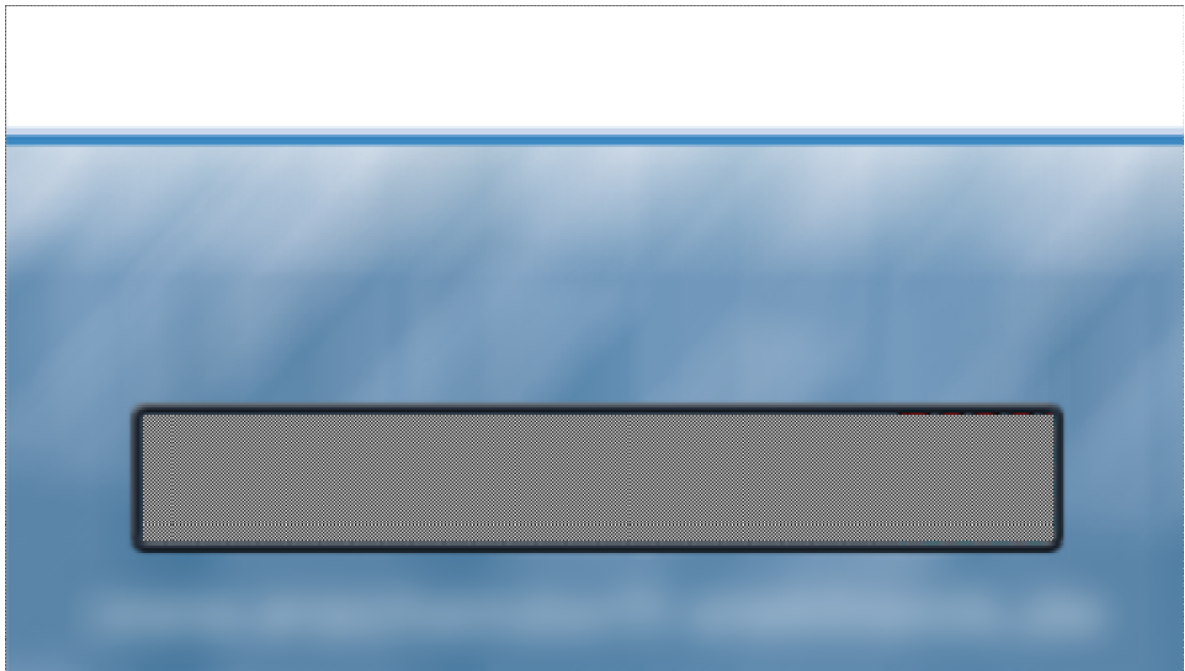
The image above shows an example loading screen image. The shaded area is transparent in the image and will be filled with a loading bar when the project loads. It provides a positive feedback for the end user that something is happening.

The progress bar will be drawn in black, filling a white area. So the colors of the loading screen image should be chosen accordingly.
The following image shows the progress bar using the above loading screen image.
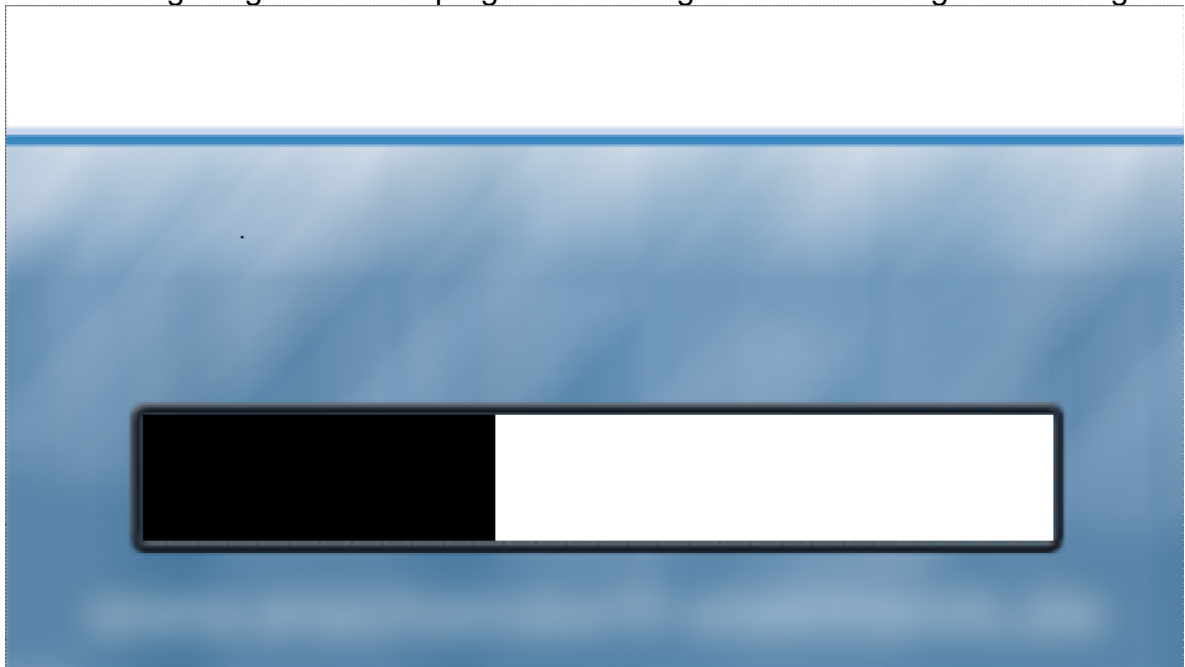
Here is another example:
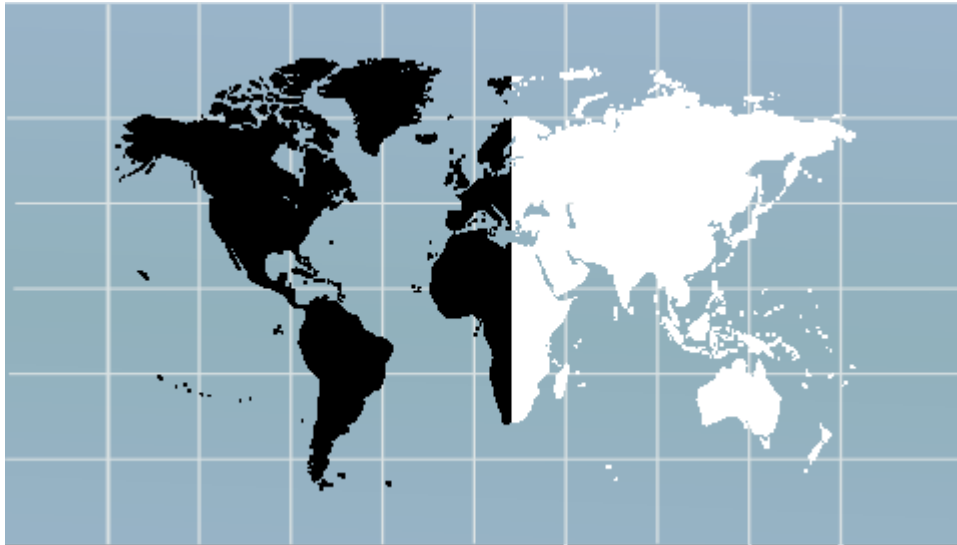
By Pressing on "Deselect image" , the image will be removed.

Boot Logo Image

The selected image will be directly displayed after the cold start of the display after the project is installed.
By pressing on "Deselect logo", the image will be removed.
Note that the deselected boot logo will not be removed from the device upon the next installation of the project.

System Background Image

The selected image will be displayed while the PClient starts and while the PClient shuts down.
By pressing on "Deselect image", the image will be removed.
Note that the deselected image will not be removed from the device upon the next installation of the

Recovery / Service Boot Logo

The selected image will be directly displayed after the cold start of the display in service mode after the project is installed.
By pressing on "Deselect logo", the image will be removed.
Note that the deselected boot logo will not be removed from the device upon the next installation of the project.

Project Icon Image

The project icon will be used on the applications running on the target device e.g. sunrise or
for PClient- and variables simulation applications running on windows operating system.
By pressing on "Deselect icon", the selected image will be removed.

Language Properties

The [Languages](#)□98 chapter describes these properties.

Unit Properties

The [Units](#)□98 chapter describes these properties.

Object Properties

"Send Value" Default

This property changes the default setting for accepting variable changes. There are two possibilities available:
- Send Value Directly
- Send Value on Enter

Here is an example that describes the difference between them:

Lets assume your project uses an numeric field. The value of this numeric field is 1. The variable attached to the numeric field is called **FanSpeed**. The user uses the encoder to navigate to the numeric filed. Then he presses the encoder to open the numeric filed for editing. Now he rotates the encoder. The value shown in the numeric changes ..2..3..4..5..6..7..8..9..10. When reaching value 10, the user ends editing of the value by pressing encoder again.

Now the difference between the two possibilities is:

Send Value Directly: The variable value of variable **FanSpeed** changes nine times: 2, 3, 4, 5, 6, 7, 8, 9, 10. If the variable was mapped to a transmit CAN message which is sent on any value change, this message would be sent nine times, too. Another example: If the variable **@DispBacklightIntensity** would have been connected to the numeric field, the user could see the brightness changes while finding the proper value.
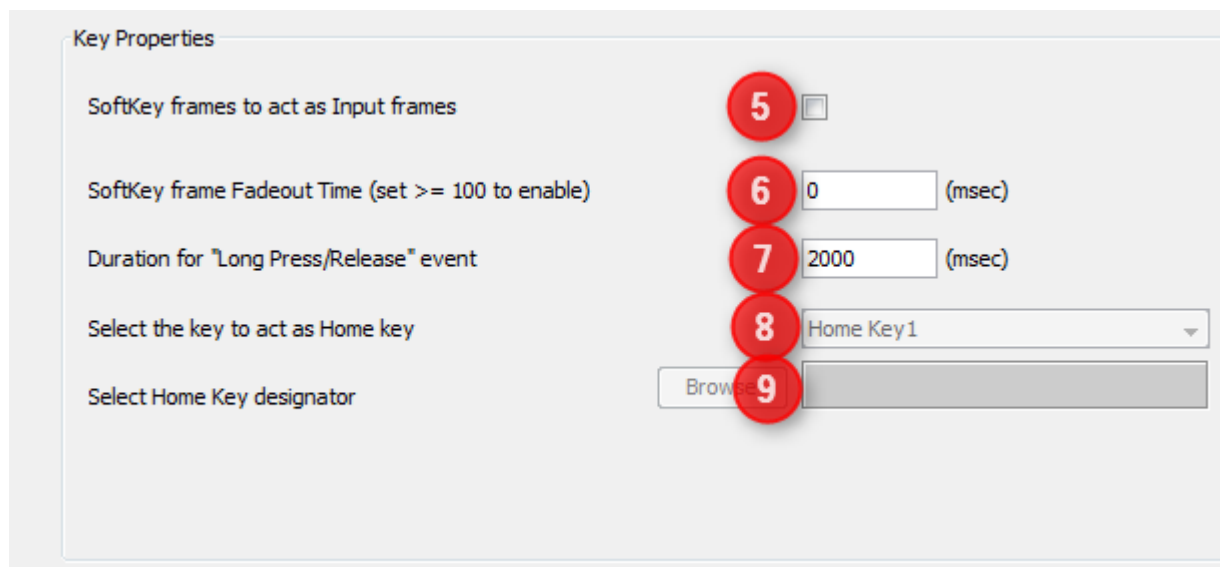
Send Value on Enter: the variable value of variable **FanSpeed** changes only one time (when the user accepts the value by pressing encoder): 10. If the variable is mapped to a transmit CAN message which is send on any value change, this message would be send only one time. Another example: If variable **@DispBacklightIntensity** would have

been connected to the numeric field, the user wouldn't see any brightness change until he presses the encoder to accept the value.

Softkey Frame

Here you can change the default softkey frame width for landscape display orientation or height for portrait display orientation. whenever you create a new page, the default width / height is used for the softkey frames.

Key Properties

Softkey frames to act as Input Frames (5)

When you enable this property, the user can use the encoder to navigate through the softkeys in the softkey frames.

Softkey frame Fadeout Time (6)

This value defines the delay time until the softkey frames are faded out (i.e. hidden). Softkey frames are shown again, when the user presses a softkey or if a java script shows them again.
A value of zero means: Disable fade out.

Duration for "Long Press / Release" Event

This time defines the time how long a user needs to press a key or touch the screen fo triggering a long press event. The long release event will be executed immediately whe a long press event was triggered and the user releases a key.

Select the Key to Act as Home Key (8)

Unused.

Select Home Key Designator (9)

Unused.

Border Properties (4)
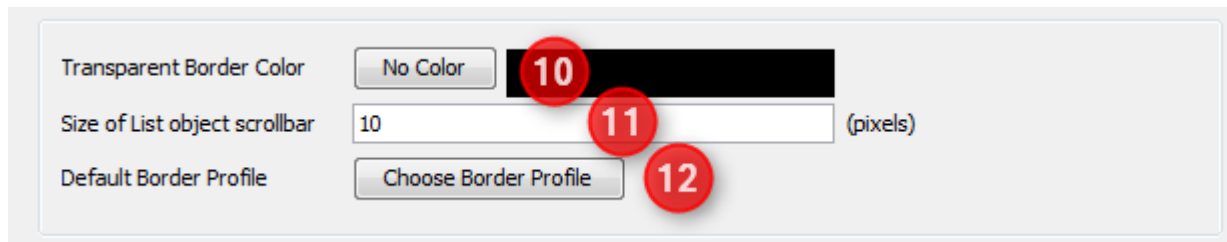
Transparent Border Color (10)

Allows changing the transparent border color. The color which is selected here will not be drawn on the screen, it will be drawn transparent.
It's also possible to disable transparent border color by clicking **No Color** button. Note: To get rid of the blue border around all objects that are created on the screen, set this property to blue (0,0,255).

Size of List Object Scrollbar (11)

When a list object contains more elements than can be displayed at once, a scrollbar appears. This property defines the scrollbar width.

Default Border Profile (12)

This property allows changing the border color profile. By clicking the button, the following window appears:
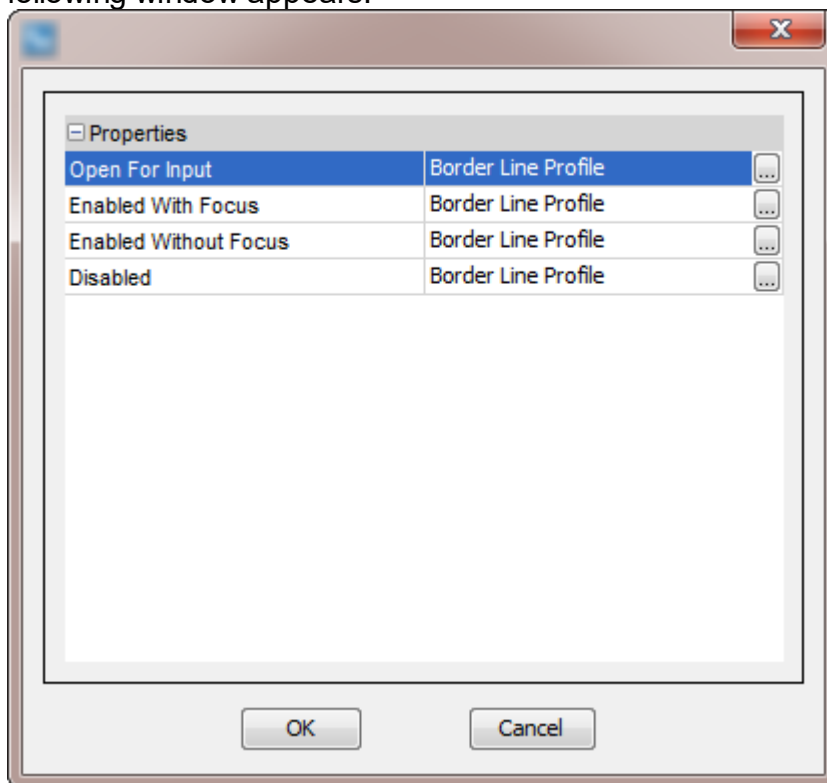
The following values can be configured:

Property	Description
Open For Input	The border color visible when the user is editing a value.
Enabled with Focus	The border color visible when the user has selected an object, but not started editing the value.
Enabled without Focus	The border color visible when the user has not selected an object.
Disabled	The border color visible when the value of an object is invalid. **Note:** This profile is obsolete and not used in this version.

Alarm Properties

The [Alarms]□568 chapter describes these properties.

Font Attribute Properties

 Tutorial video about fonts and font attributes
https://www.youtube.com/watch?v=tfkXFy-AQzM
Using different font styles in a project is different than in common text writing applications. Each **Font Attribute** is a combination of font size, font style (italic and / or bold) and font color. Only the combinations you have created as **Font Attributes** are available when creating your project pages. When designing your project pages you can use all the font attributes you have defined and combine them with the available fonts. Whenever you need a new combination of size, style and color, you need to create a new **Font Attribute**. This allows you to quickly change e.g. the size of the text for all String Fields that use the same Font Attribute.
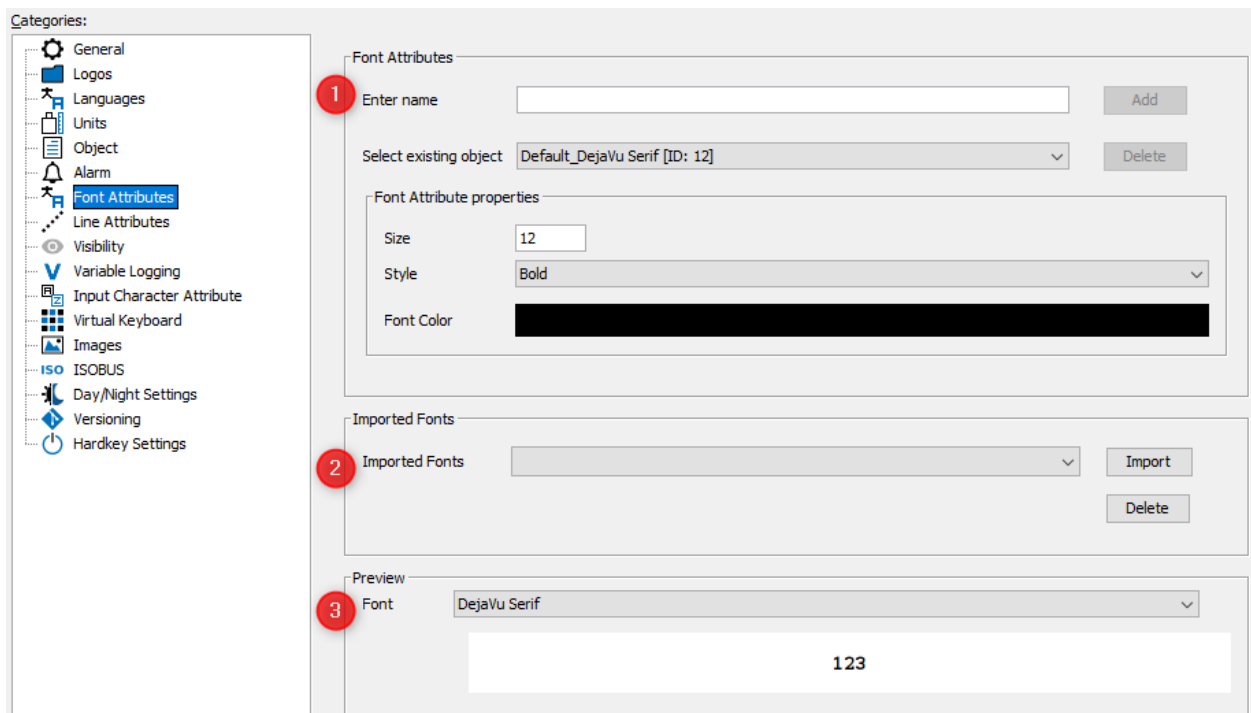
Font Attributes (1)
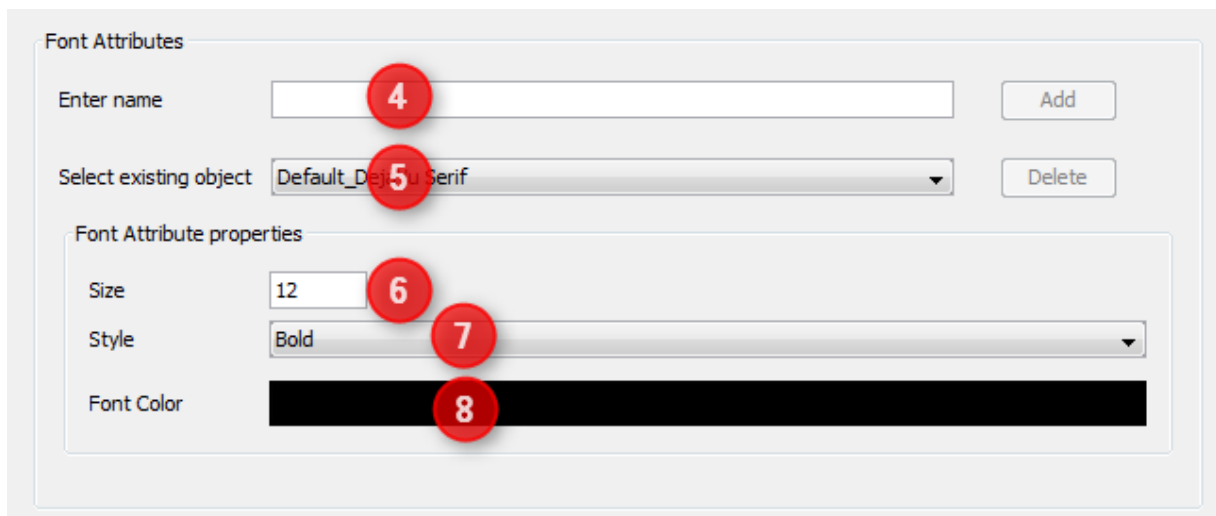
To create a new **Font Attribute**, enter a new font attribute name in (4) and click the **Add** button. The name entered here will be used to select a **Font Attribute** while creating the project pages. Afterwards you can configure the new **Font Attribute** as described below.
To change the properties of an existing **Font Attribute** select the **Font Attribute** (5). Then change the size (6), style (7) and color (8) according to your needs.
To delete a **Font Attribute** select the **Font Attribute** to delete (5) and click the **Delete** button.

Imported Fonts (2)

To make new font styles available you can import True Type Fonts here.
You also can delete fonts, because each font increases the size of the projects.
Note: It is highly recommended to delete unused custom fonts.

Note: Please be aware of the copyright situation of fonts.
Note: Font files have a lot of name metadata. The font file will only be used successfully if the font name is the same in as many places as possible. You can check and change these settings with a font editor like FontForge:

Bad example, doesn't work:
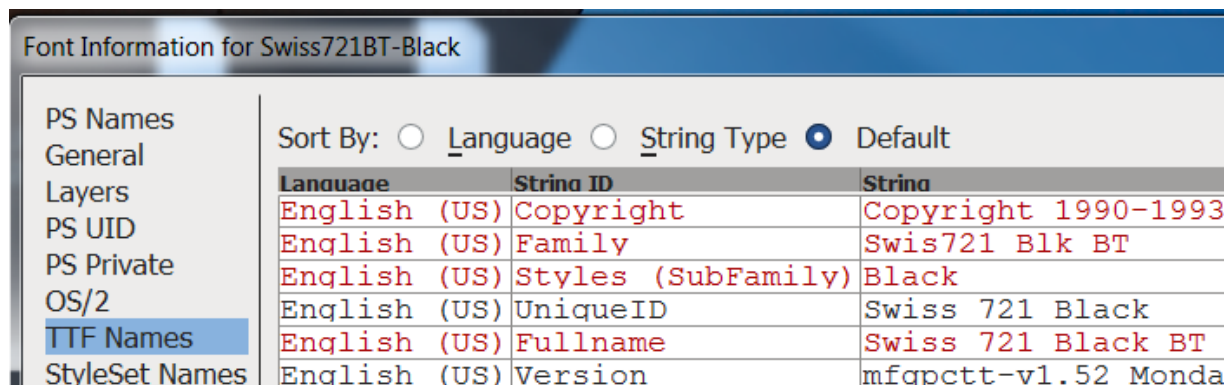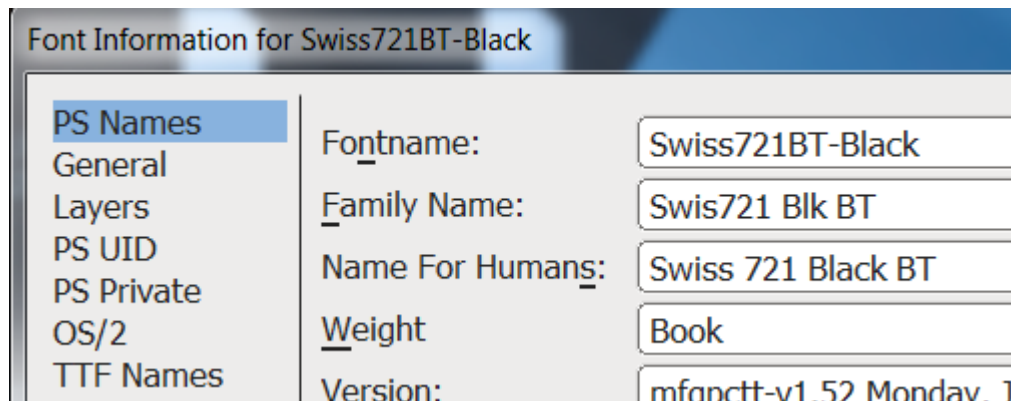
Good example, works:
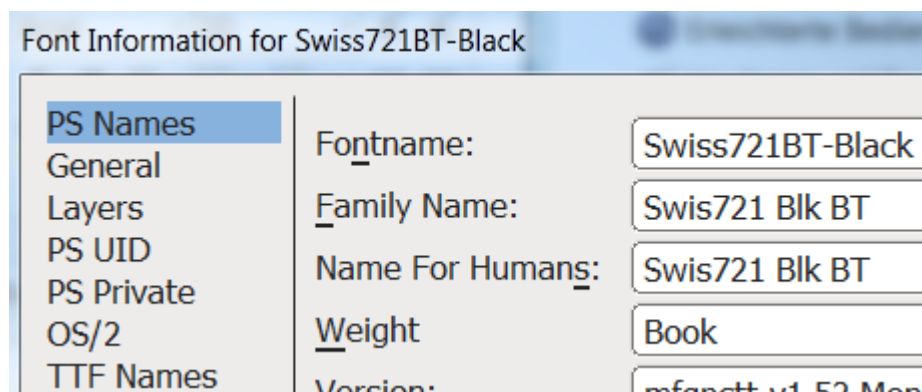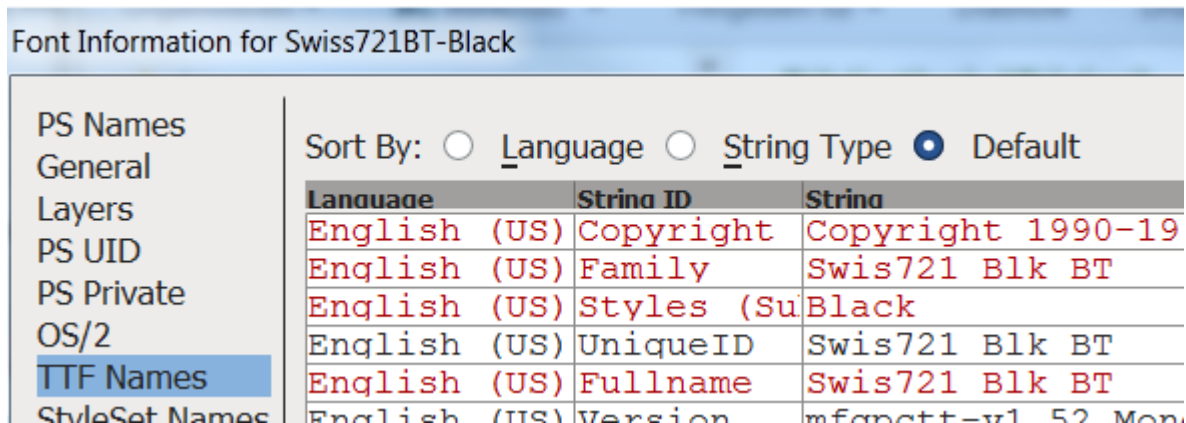
Preview (3)

You can use the preview to see how the selected font looks with the selected font attribute. The background will change color between black and white depending on the brightness of the set font color so it is always visible.

Line Attribute Properties
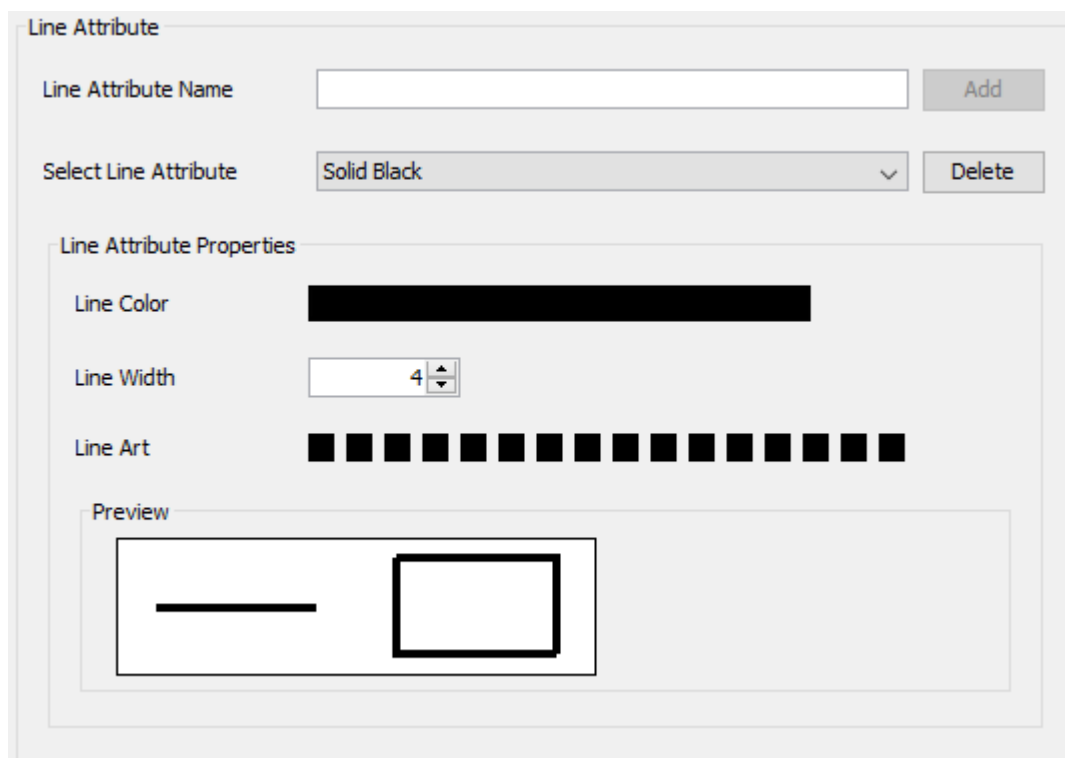
In this category line attributes for the 2D Graph[451] can be created and modified.
To create a new line attribute, enter a name in the Line Attribute Name field and press Add.
A line attribute consists of:
- a color - click on the color bar and select a color
- a width - enter a width or change the value with the up/down buttons. Line widths between 1 and 255 pixels are allowed.
- a "Line Art", i.e. a dash pattern - click on the squares to change the dash pattern
There is a preview field showing the result of the settings made.

Visibility Properties

The Visibility[588] chapter describes these properties.

Variable Logging Properties

The Variable Logging chapter describes these properties.

Input Character Attribute Properties

When you offer the user the possibility to enter values with the encoder, you most likely want to restrict the characters he is able to use.
Example: When your project asks the user to enter his name, you might want to prohibit the possibility to enter special characters like !?_ and so on.
This is what the **Input character Attribute Properties** are used for. You can store a selection of allowed characters as an **Input Character Attribute**. While creating your project, you can assign these **Input Character Attributes** to string objects.
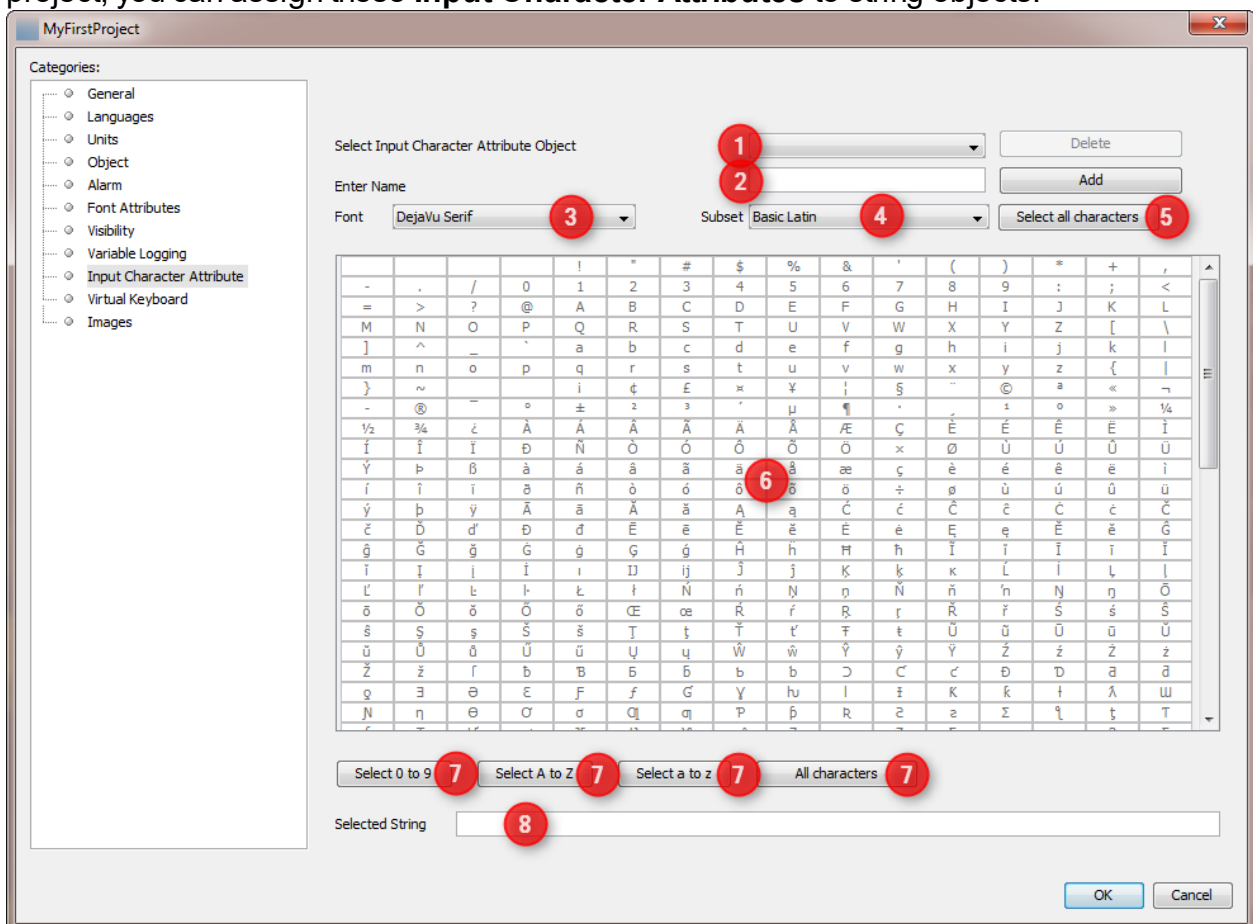
To delete an existing **Input Character Attribute** choose the one to delete (1) and click **Delete** button.
To create a new **Input Character Attribute** enter a name for the new attribute and click **Add** button. The name is used in the editor while creating project pages to make a string object use the **Input Character Attribute**.
To change the characters available for an **Input Character Attribute** choose the attribute to change (1). Then you can choose any of the fonts defined in the tool (3).

Now you can start selecting characters by clicking them in the table (6). The table shows the characters in Unicode (UTF-8 encoding form) code pages.
For faster selection of code pages, you can select the subset (4) (it contains a list of character blocks) and click **Select all characters** button. Using the same procedure, you can also unselect code pages by clicking the **Select all characters** button again.
Furthermore there are buttons available for fast selection of typical character ranges (7). They can also be used to unselect characters by pressing the buttons again.
The selected characters are further shown in the bottom of the window (8).
Note: For a complete listing of the normative character blocks in the Unicode Standard, see here.

Virtual Keyboard

When your project uses virtual keyboards for entering values you should configure, which virtual keyboards have to be used for which language. Because the character set used by languages are very different, you most likely need to create one virtual keyboard per language used in your project.
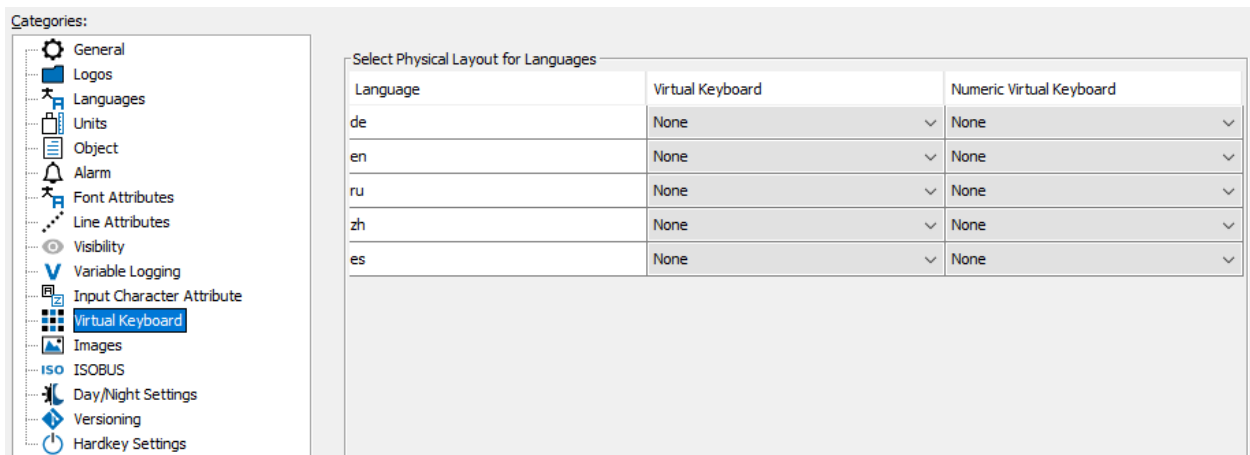
First you need to create the virtual keyboards. Then open the **virtual Keyboard** section of the project properties and select the virtual keyboards that have to be used for each language.

Images
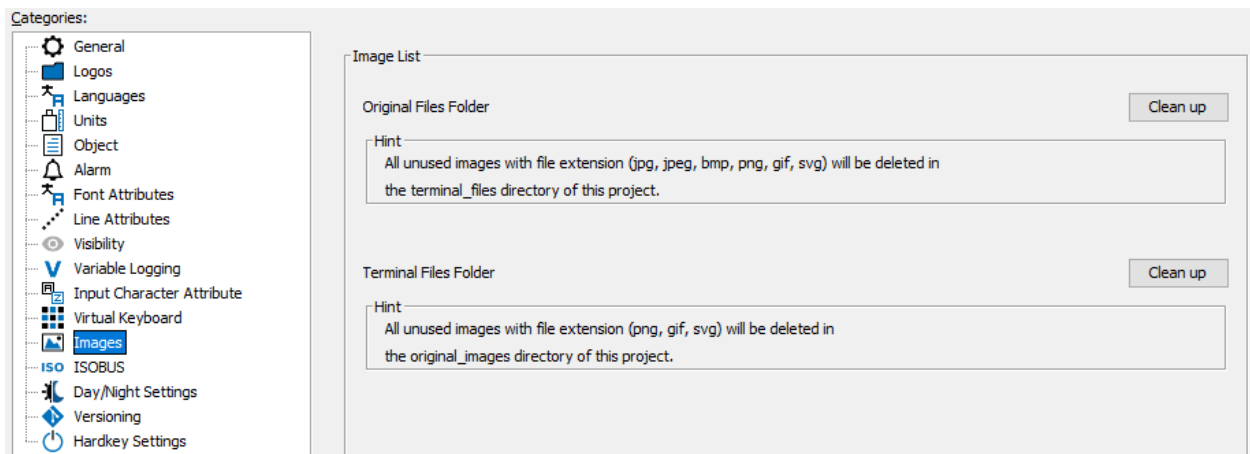
Here it is possible to clean up the folders "original_images" and "terminal_files" to delete unused images.

1. If an image is added to a project the original image will be added to "original_files" folder. Additionally the scaled or modified image is added to "terminal_files" folder.
When the project is updated (objects are deleted) it could be that images will be removed from "terminal_files" folder and the original image is not used anymore.
In this case it could be helpful to delete unused original images.

2. Normally the "terminal_files" folder is automatically cleaned up. But if images have been manually copied to this folder, the user has the possibility to remove them.

NOTE: Please never manually copy images to "terminal_files" folder. This folder is completely maintained by the application.
If you do anyway please be aware that the files will not be used by the application and disk space is lost on the target device when project is downloaded.

HINT:
Original Files: On clicking "Clean up" (1) all unused images with file extension (jpg, jpeg, bmp, png, gif, svg) will be deleted in "original_images" folder.
Terminal Files: On clicking "Clean up" (2) all unused images with file extension (png, gif, svg) will be deleted.
For more information about "Image Handling" please refer to the Image Handling[92] Chapter.

ISOBUS
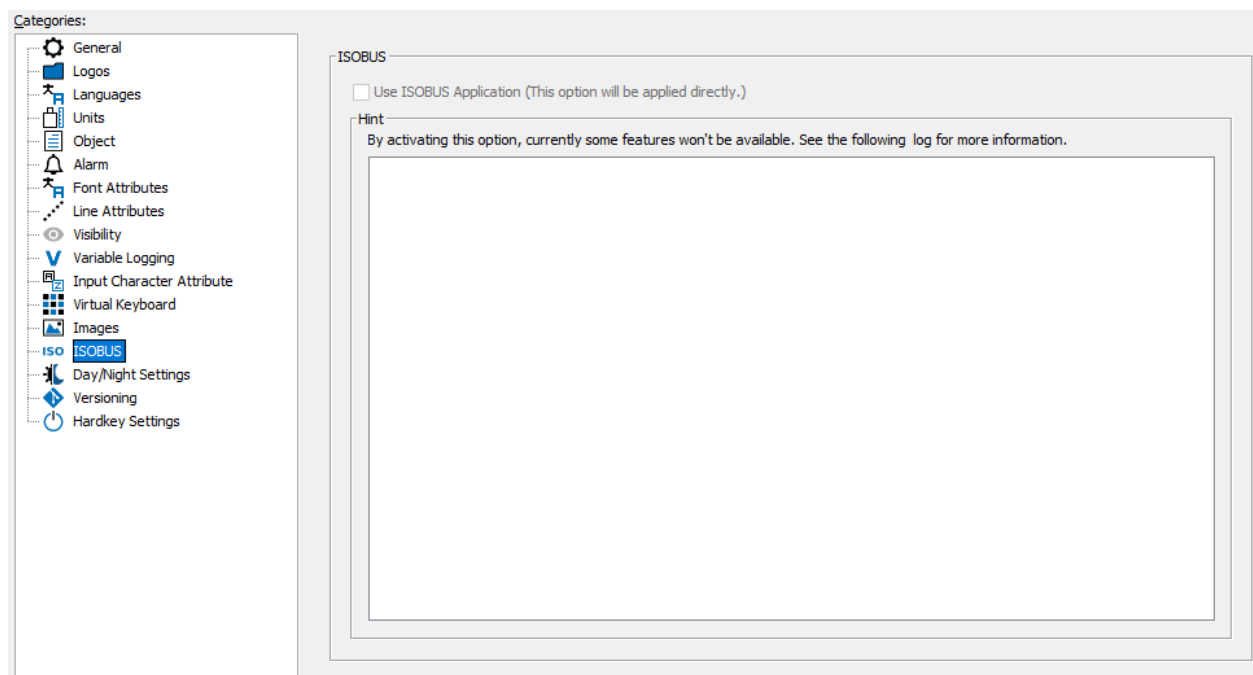
For OPUS A6G2, OPUS A8, OPUS B3, X18 and X40, ISOBUS projects can be created, see here[576] .
The ISOBUS functionality can be viewed in this category.
The media player and variables that are not available for ISOBUS devices will be deactivated when an existing project is converted to an ISOBUS project.

Day/Night Settings

The [Day/Night](#) [571]chapter describes these settings.

Versioning

The [Project Versioning](#) [123]chapter describes these settings.

Hardkey settings
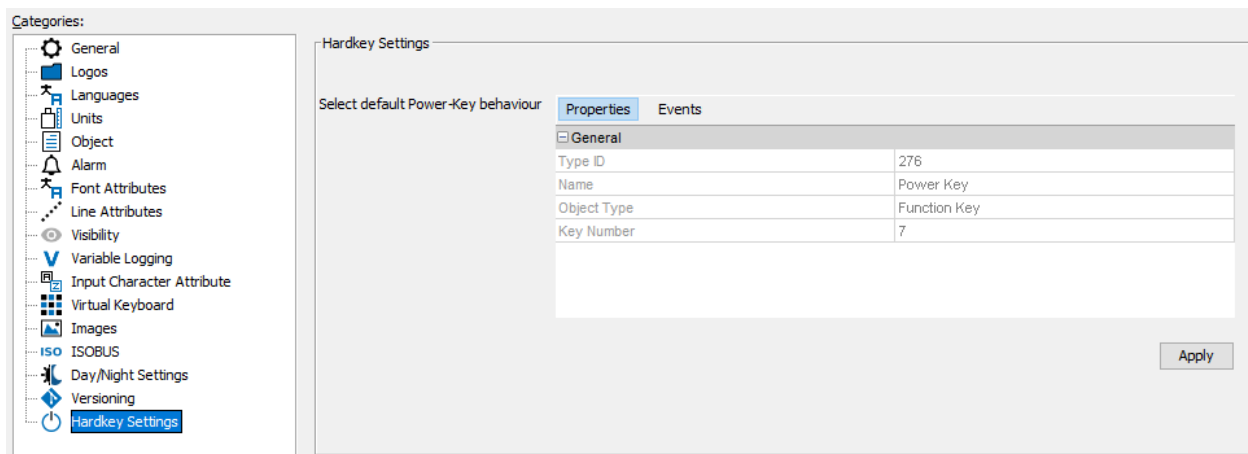
For devices with a power button, the project-wide behavior of that button can be defined.
This behavior can be overridden by a per-page setting on each page.
On each page the same events can be configured specifically for that page.
Please note that if ANY event is configured for a specific page, NO action from the project
properties will be executed while on that page, i.e. if you want a specific action OnPress
but the general action OnRelease, you need to put that action into the page event
configuration for the Power Key, as well.
On pages that have no event configured, the actions from the project properties will be
executed.

Please note that a hard shut-off is configured for a press longer than 5 seconds.

3.11.1   Image Handling

This chapter describes how images are handled by the tool.
Thereby two definitions are very important to understand the **Image Handling**:

- Original Image
- Scaled Image

"Original Image" means the image which is added to the project via "Image chooser"
dialog. This image will be copied to "original_images" folder and will never be modified by
the tool.
"Scaled Image" means the image which is created from "Original Image" with e.g. modified
height and witdh. This image is saved in "terminal_files" folder.

Overview

For the **Image Handling** there is a separat file "image_list.xml" which stores all informations about the images in a project. The file is saved in the project folder.
During working on the project the Tool creates an temporary file called "image_list.tmp" which saves all image changes in the project. When project is saved the "image_list.tmp" file is saved to "image_list.xml" and the temporary file will be deleted.
All original and scaled image infomrations are stored in these file. Each image name is stored with the according MD5 checksum. The MD5 checksum is used to recognize if there are duplicate images. Additionally the "image_list.xml" file stores the assignment between image property and Object.

Purpose

The **Image Handling** shall help to avoid duplicate images in the project. The device has restricted resources and they should not be wasted with duplicate images.
This helps to keep the performance on the device (by loading the project) and save restricted disk space.

Image Naming

This chapter shall describe how the naming of images works in the Tool.

⊞Original Image Naming

If an image is added to an Object, the image is added to "orginal_images" folder. The name of the original image is normally not modfied except of 2 cases.
Case 1: The folder "original_images" contains already a file with same name, but different content.
The new image is added to "original_images" with an extension.
Example:
Original_images folder contains an image with name **"test.png"**.
A new image is added with same name but different content. The new image will be renamed to **"test_1.png"**.
Case 2: The folder "original_images" contains an image with same content, but different name.
The new image is **NOT** copied to "original_files" folder, because there is already an image with same content. For the scaled image the name of the already existing image will be used.
Example:
Original_images folder contains an image with name **"test.png"**.
A new image with name **"myImage.png"** is added with same content. The image "myImage.png" is **NOT** added to "original_images" folder.

Scaled Image Naming

The scaled image contains the name of the original image with the extension of "Width" and "Height" and optional "Color of Transparency".
Example:
Original Image Name: **test.png**

Scaled Image Name: **test_100_80.png** (without transparency color)
Scaled Image Name: **test_100_80_0xffffffff.png** (with transparency color "white")

NOTE:
1. For original image files with extension JPG/JPEG and BMP the scaled images will be converted to PNG.
Example:
Original Image Name: **test.jpg**

Scaled Image Name: **test_100_80.png** (without transparency color)
Scaled Image Name: **test_100_80_0xffffffff.png** (with transparency color(argb) "white")

2. For original images files with extension GIF or SVG the SAME image will be used for original and scaled images. For these image formats the "Transparency Color" is **NOT** available.
Example:
Original Image Name: **test.svg**
Scaled Image Name: **test.svg**
ATTENTION:

The name of the images will be modified only when project is saved. That means if an image (test.png) is assigned to an Picture Graphic Object with Width = 100 and Height = 80 the Scaled Image Name is "test_100_80.png".
The name is immediately updated. **BUT** if now the size of the Picture Graphic is updated. e.g. to Width = 60 and Height = 40 the image name will not be updated to "test_60_40.png" until project is saved.

Adding an Image Example

(1) After Dragging and dropping the object on the canvas,
click on Image Chooser Button to configure a new image.

⊟ Picturegraphic Specific		
Image To Display	default_icon_100_100.png	...
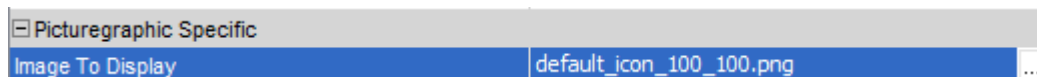
(2) The Image Chooser dialog is open with the currently configured image.
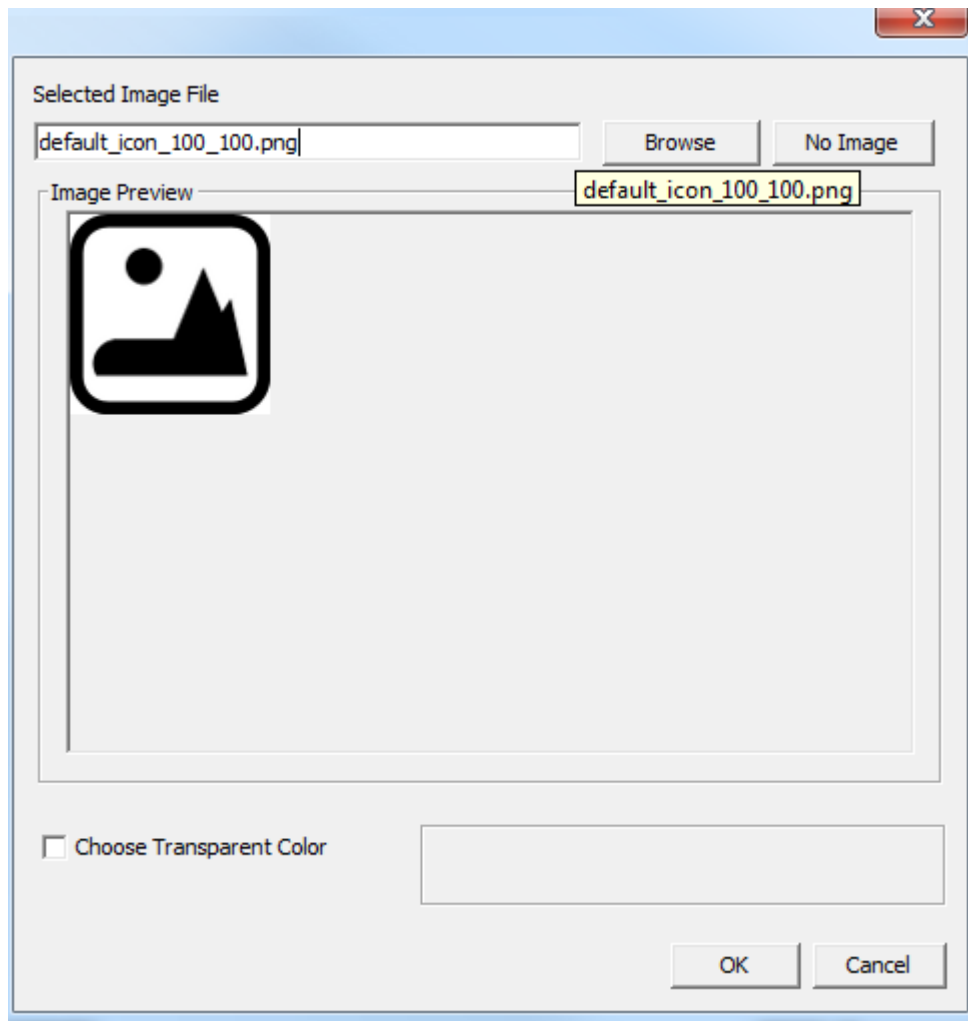
(3) Click on "Browse" to select a new image.
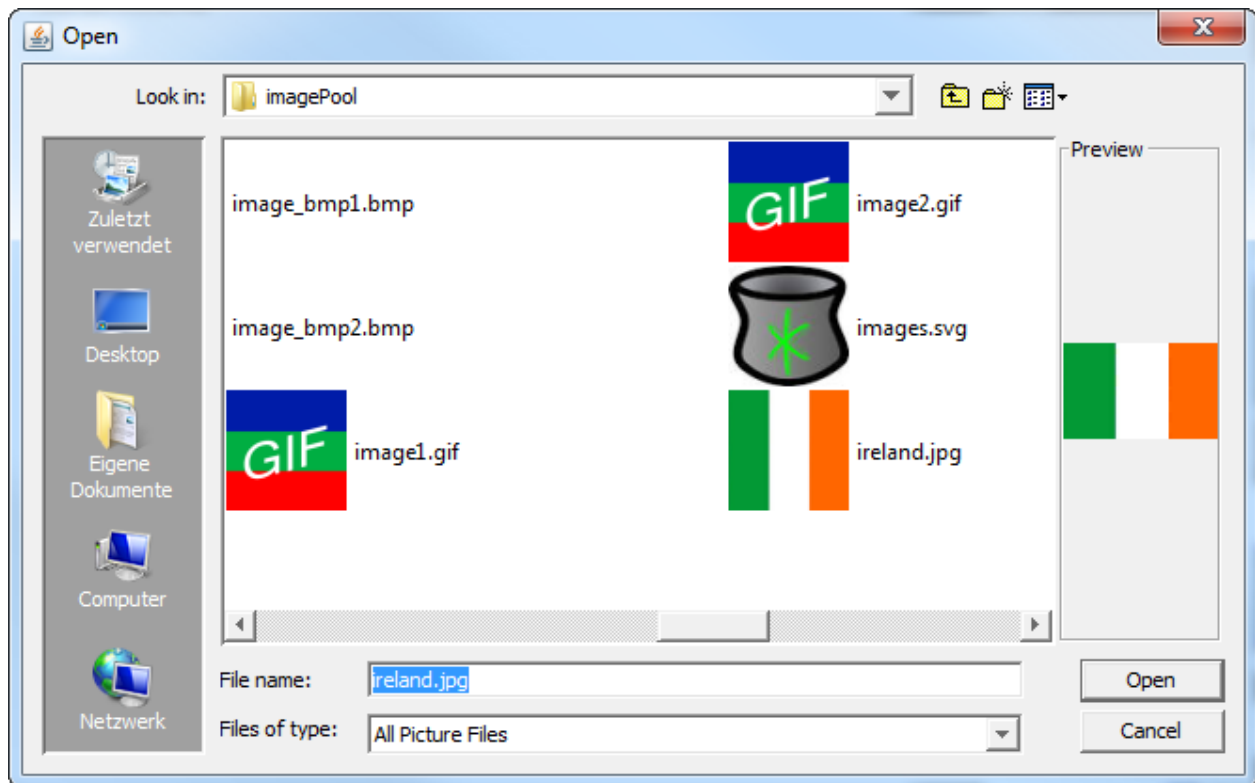
(4) Select a new image
(5) Click on "Open" to add new image.
(6) Original Image name with path of new image
(7) Preview of new Image
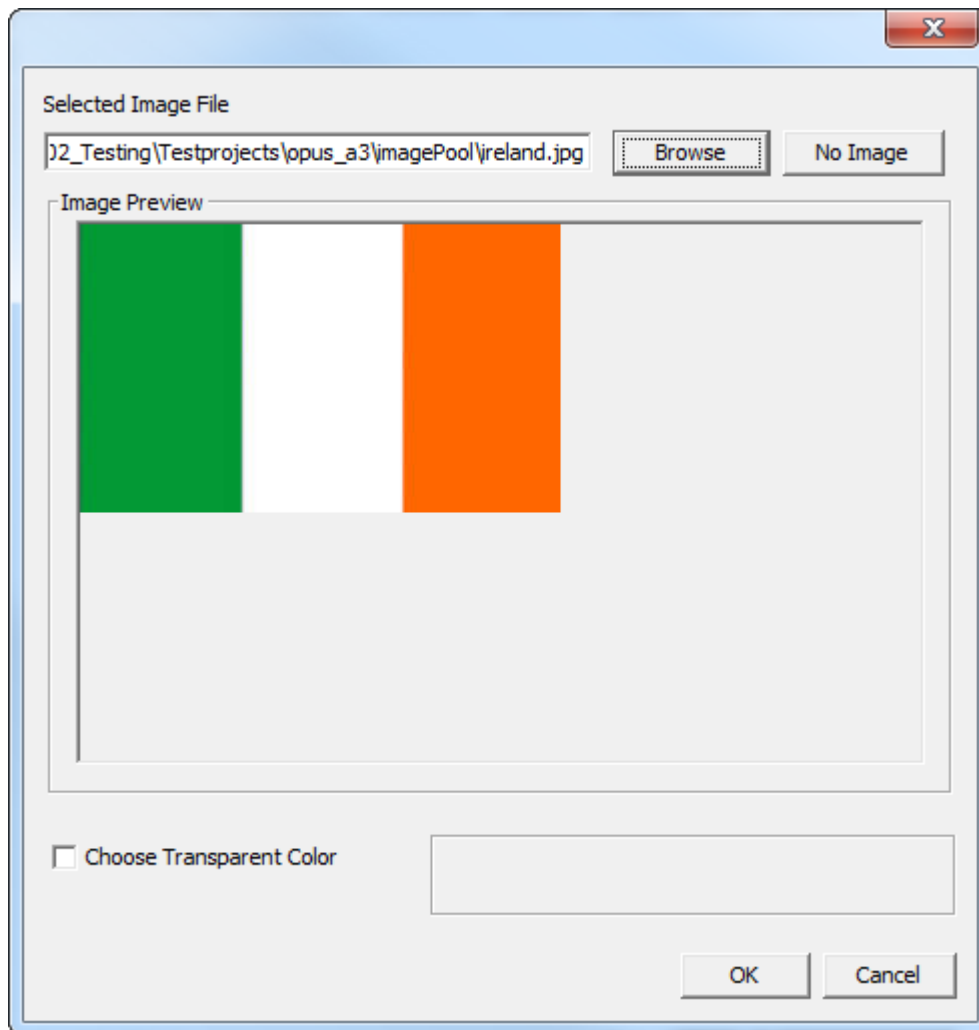
(8) Click on "Ok" to set new image.
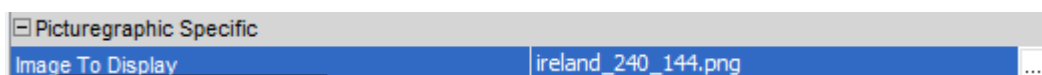
(9) New image is shown in Designer for "Picture Graphic" object.
(10) The property "Image To Display" shows the name of new created scaled image "ireland_240_144.png"

(11) When property "Height" and "Width" are changed, a new scaled image is created with new name "ireland_100_100.png".
NOTE: The image name of property "Image To Display" is not changed until "Save" is performed.

Now, if a new Picture Graphic object is created with same "Height" and "Width" and the same original image is assigned again, the **Image Handling** recognizes that the scaled image is already used and doesn't create it again. The already existing scaled image is also used for the new Picture Graphic object.

Conversion of Old Projects (<1.9.x)

If the tool Version > 1.9.x is used and a project with a lower version (< 1.9.x) is opened the project is automatically converted for new **Image Handling**.
After the conversion of the project, all scaled images will be **recreated** and **renamed**. This step is necessary to avoid duplicate images.
For recreating the scaled images the original images must be available. By missing of the original images, the old scaled images are used but they will be renamed in any cases to new format according rules described in [Image Naming]^{93} Chapter.
Note:
If it is not desired that the scaled images are recreated, you can delete all images in the "original_images" folder before converting the project. Then all old scaled images are not recreated.
For performance reason, this step it is not recommended!

Consistency Check

The new **Image Handling** has a "Consistency Check" which is performed after every "Save" and "Load" actions. This Check ensures that all images which are used in the project are stored in "terminal_files" folder. If images are missing they will be created again. But the "Consistency Check" can only be performed properly if the original images are available in "original_images" folder. If a scaled image is not available in "terminal_files" folder, it will not be visible in the tool.

Supported Image Files

The supported image formats are listed [here]^{497} .

3.11.2 Languages

3.11.3 Units

This feature is always a little bit misleading. The tool and the run time on the device do not offer any pre-defined calculations like Celsius to Fahrenheit and vice versa. You can see the groups of units in the project properties in the Unit category.
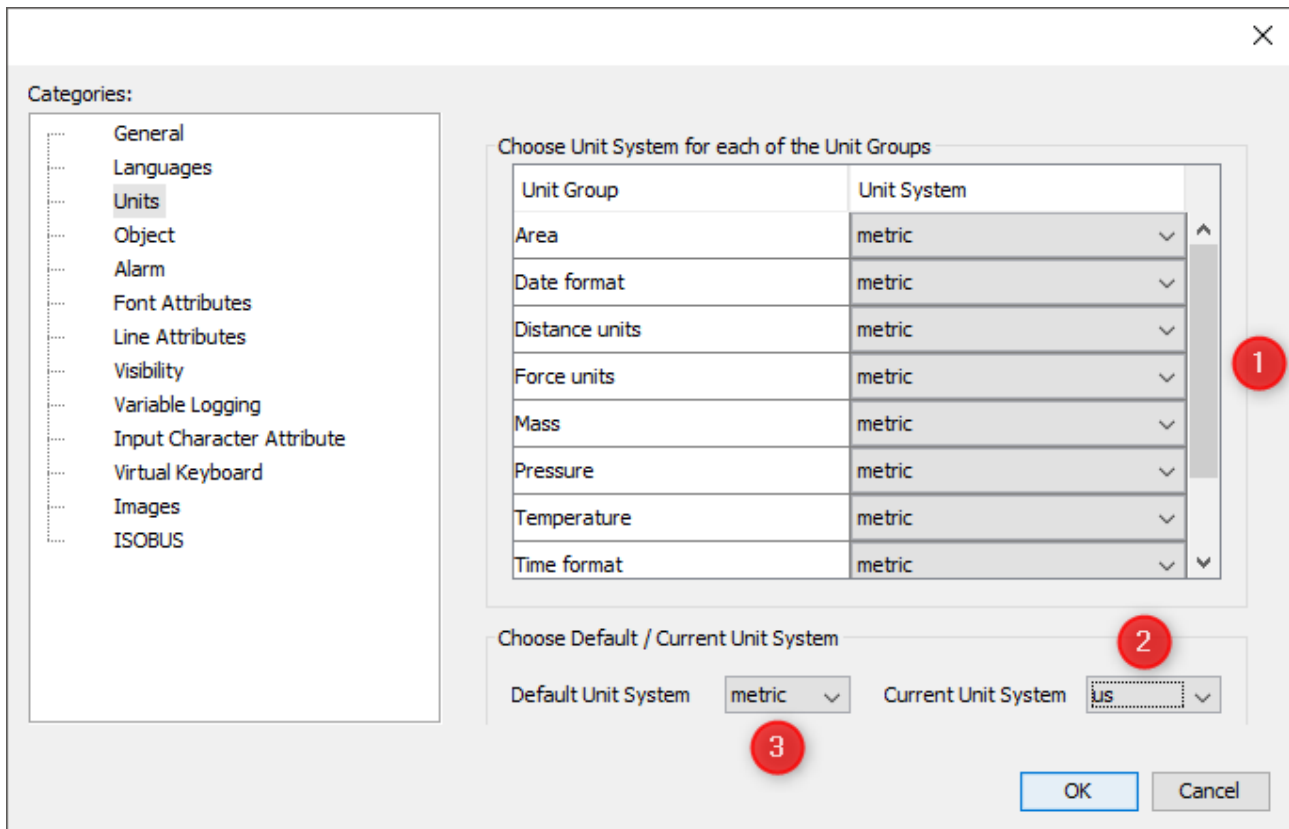
What you see here are possible groups of units which you can use. There are no automatic calculations done in the background. They all have to be defined within the tool and the properties of the objects. The current unit settings for the different unit categories (1) and the general unit setting (2) are only for the development phase. Setting the default unit system (3) does currently not have an effect.

To enable this feature for an object you have to change the dependency property of an object.
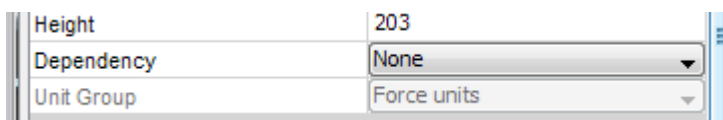
Change the dependency from "None" to "Unit" and you will see that some properties will get "..." buttons.
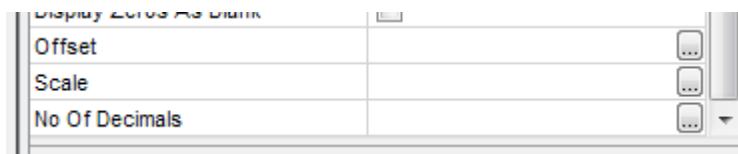
This is an example for a numeric object. These properties differ depending on the object selected (please see object reference for details). Usually you can enter only one value in "Offset", "Scale" or "No of Decimals". Now with the Unit dependency you can click the "..." button and get a new dialog.

This is the example for the "Scale". Here can now enter the values which shall be used for the scale depending on the unit which is currently chosen for the unit group.

For example: You choose the unit group "Force Units". Per default this group is selected as metric, so the scale value entered in the metric field is used. To change this you can make use of two mechanisms. First you can set the "Current Unit System" within the project properties (this will only have an effect in the designer, not at run time). The second is that you can change it during run time with the @SettingsForceUnits variable. Please note that you can not choose all unit systems for all variables. Please see the variables[174] for details. For the force unit group you can choose, for example, only: similar to ISO 11783-7, A.23.5.8

0	metric [kN, N (Newton)]
1	imperial [pounds-force]

So what this feature does, in the end, is switch between different offset and scaling value sets for objects displaying numerics directly or graphically, and texts for String Fields.

3.11.4   Variable logging

Important note: If you have selected variables inside the variable logging dialog and delete them outside (e.g. variable list), the variables are not automatically updated. If you have deleted variables for example with the variable list, please go one time inside the variable logging dialog and save the project afterwards. Now the lists are up to date. Do not worry if any variable which is not available at runtime is still in the list, this is anyway handled at runtime of the device.

Please note: To start the logging on the device, the variable @LogLevel has to be set to a value >0.

Variable Logging provides the possibility to capture valuable process and sensor data, giving the insight into past process upsets and events. To configure the Variable Logging use the main menu tab File | Project Properties and select the Variable Logging category.
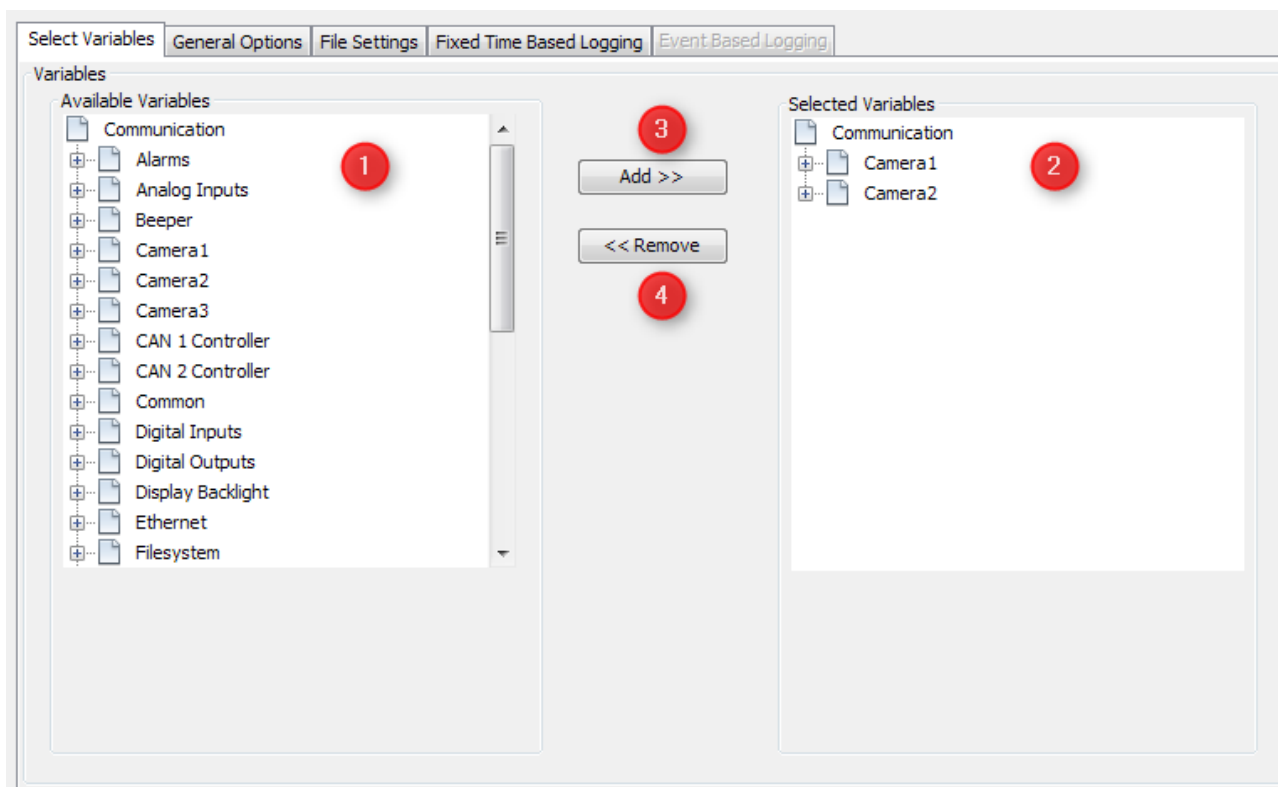
The tab provides you an overview of the current available variables (1). Now you can add (3) and remove (4) the variables which you like to log with the according labelled buttons. All selected variables you will find in the right tree view (2). Variables are written to a file. How the variables are written is defined by the method of logging. You basically have two choices "fixed time based logging" and "event based logging".

General Options
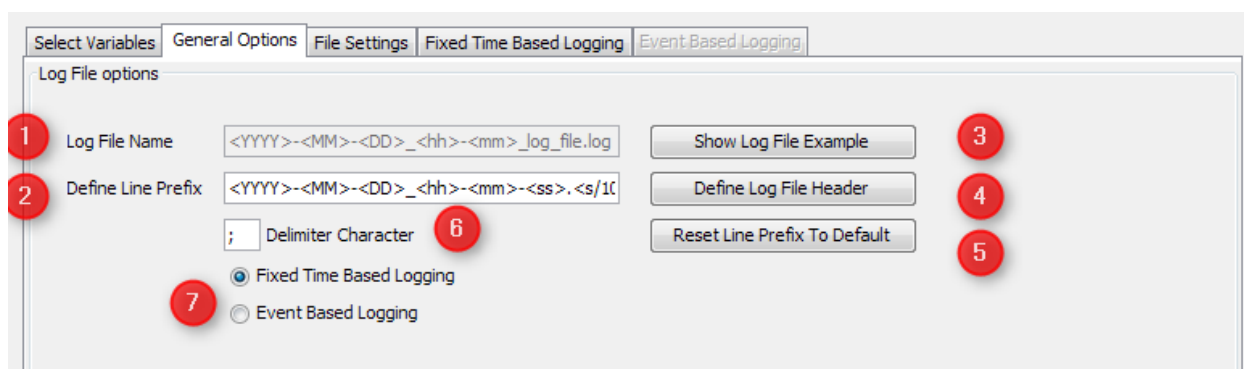
Changes to older versions: Log path and Copy/Move path can be set during runtime with the variables @LogPath and @LogCopyPath. The defaults are mentioned below.

Variable	Default
@LogPath	/opt/pclient/logfiles
@LogCopyPath	/disk/usbsda1

Configure the Log File Options
The option "Log File Name" (1) is disabled. The file name is chosen by the system and can not be changed.

Show Log File Example (3)
This shows an example of how a log file could look like. **Note: This is not a run time preview so the shown example will not change based on the changes you made to the settings.**
Define Log Header File (4)
When clicking the button Define Log Header File, a new dialog opens to insert the text that will be later printed in the log file at every start of the variable logger. The appearing dialog only consists of a text input field and an OK and Cancel button. The following example text is set by default:
Log File of Project Version

The following variables will be logged:
<LIST_OF_LOGGED_VARIABLES>

After finishing the input, use the OK button to store the log file header or the Cancel button to discard all changes. To see all tag options go to <u>Allowed tags for file header</u>⟂100 .
Define Line Prefix (2)

Right besides the label Define Line Prefix (2), enter a string with which every line of the log file will begin. On the right side of the line edit field a button Reset Line Prefix is lo-cated. Clicking this button will set the following default text to the line edit field:
<YYYY>-<MM>-<DD>_<hh>-<mm>-<ss>.<s/10>
Define Delimiter Character (6)

On the right side of the label Delimiter Character, enter one character that will be inserted between each variable value (time based) or between variable name, event and value (event based). It is possible to enter only one character.
"Fixed Time Based Logging" or "Event Based Logging" (7)
You can choose the option which is right for you. The time based option writes all selected variables in one line. This makes it easy to transfer the data into a spreadsheet. The variable entries are sorted in the order you defined inside the time based logging tab.

```
 2   2014-03-13_15-01-33.0 10;22;13;32;0;0;0;0;0;
 3   2014-03-13_15-01-33.5 10;22;13;32;0;0;0;0;0;
 4   2014-03-13_15-01-34.0 10;22;13;32;0;0;0;0;0;
 5   2014-03-13_15-01-34.5 10;22;13;32;0;0;0;0;0;
 6   2014-03-13_15-01-35.1 10;22;13;32;0;0;0;0;0;
 7   2014-03-13_15-01-35.6 10;22;13;32;0;0;0;0;0;
 8   2014-03-13_15-01-36.1 10;22;13;32;0;0;0;0;0;
 9   2014-03-13_15-01-36.6 10;22;13;32;0;0;0;0;0;
10   2014-03-13_15-01-37.1 10;22;13;32;0;0;0;0;0;
11   2014-03-13_15-01-37.6 10;22;13;32;0;0;0;0;0;
12   2014-03-13_15-01-38.1 10;22;13;32;0;0;0;0;0;
13   2014-03-13_15-01-38.6 10;22;13;32;0;0;0;0;0;
14   2014-03-13_15-01-39.1 10;22;13;32;0;0;0;0;0;
```

The event based option writes one line per variable per event. By default it includes the variable name and the event when the variable is written. If you like to have shorter entries please refer to the chapter <u>Event based logging</u>⟂100 .

```
2  2014-03-13_15-39-22.9 New Variable_3;LOG_TIMER1_EXPIRED_EVENT;0
3  2014-03-13_15-39-22.9 New Variable_7;LOG_TIMER1_EXPIRED_EVENT;0
4  2014-03-13_15-39-22.9 New Variable_2;LOG_TIMER2_EXPIRED_EVENT;0
5  2014-03-13_15-39-22.9 New Variable_4;LOG_TIMER2_EXPIRED_EVENT;0
6  2014-03-13_15-39-22.9 New Variable_6;LOG_TIMER2_EXPIRED_EVENT;0
7  2014-03-13_15-39-22.9 New Variable_8;LOG_TIMER2_EXPIRED_EVENT;0
8  2014-03-13_15-39-22.9 New Variable_1;LOG_TIMER3_EXPIRED_EVENT;0
```

File settings

It is possible to select between two options that will effect the creation of new log files:

- Continuous Logging
- New File at Restart

The user can select one of the options by clicking one of the radio buttons with the respective labeling. When selecting continues logging only two files are created. You can see this at the log file name it changed to "data_log_file.log". When the filesize is reached one backup is created and a new "data_log_file.log" is created. When using the "New File at restart" a circular buffer is used for the variable logging. It is based on maximum file size and number of files.

To define the maximum file size of one logging file, enter the size in kilobytes right besides the label Maximum File Size. A Size of 0 will stop logging when no more disk space is available. Default value is set to 1000 kilobyte. During runtime it is possible to adjust the maximum file size with the variable @LogMaximumLogFileSize. **Please note that this variable is also checked at runtime. You can only set the file size between the range of 100 kilobyte up to 50000 kilobyte. Please also note that this variable is remanent. That means that the value on the device always overwrites the value coming from the project. To reinitialize all variables please re-install the runtime on the device.**

"Continuous Logging" and "New File at restart". Both options have main impact on the file based ringbuffer of the logging mechanism. For both options you can set the maximum filesize. If you do that the maximum filesize on the device is calculated. The image above

shows the second option. With "Continuous Logging" you have a maximum of two files of the file size in the memory, which you can see above. The number of files field is not changed accordingly, it is only changed by the "new file at restart" option.

"New file at restart" opens new options (as seen in the screenshot). Now the option "Number of files" is enabled and you can set the number of files in the ringbuffer. Please note that this is a file based ringbuffer not a variable based one. As you can see here the maximum filesize is calculated with Number of files + 1. The reason is because we open a new file and delete afterwards.

For variable logging you can use half of the available mass storage of your device. The usage is indicated by two fields. One is the current disk usage and the two color coded fields with the 512MB and 1GB label. These are in first place indicators. At runtime the real available space is checked. If the space is not sufficent older data is overwritten. This is shown by the error variable @LogCopyError (codes 20 and 25).

Fixed Time Based Logging
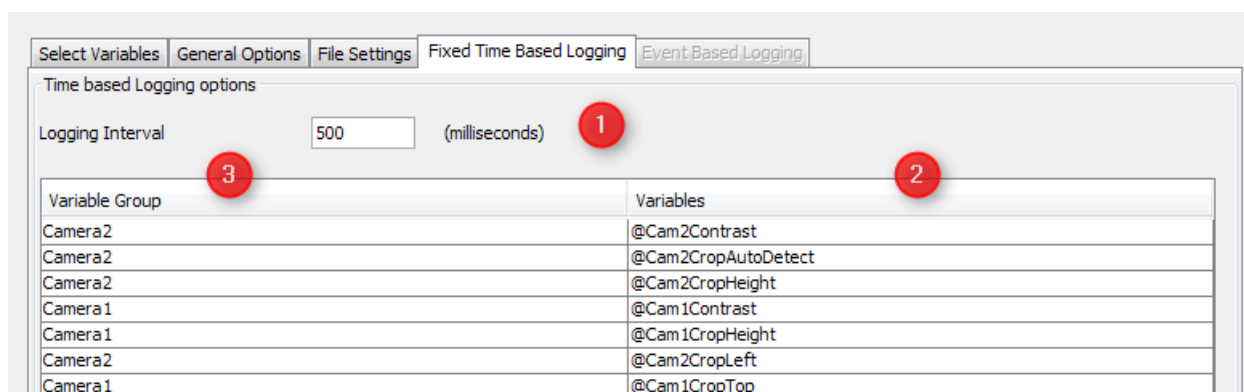
Here is only one option for the time based logging. It is a timer value (1) that can be defined in the line edit field besides the Logging Interval label. The lowest number, the user can enter shall be 500. You can sort the variables by drag and drop. The shown order will be the order in the log file. The header columns can not be used for sorting.

Event Based Logging
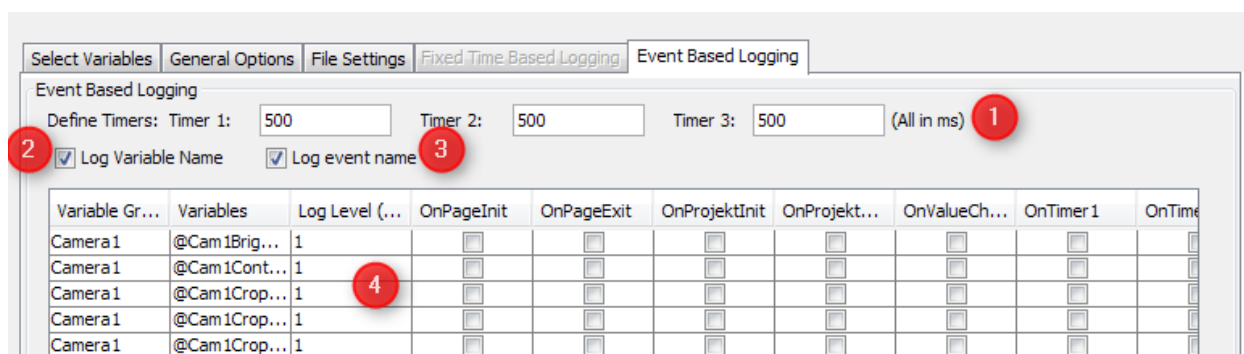
Timer settings (1)

When doing event based logging, the user can select three timers. He can enter the timers in milliseconds besides each label (Timer X:). A variable can be assigned to one of these timers (then it will be logged each time the timer expires).

To shorten the log lines you can use option (3) and (4). Per default everything will be logged.

Sorting the table

You can sort the table by clicking on column headers.

Please note: Event based logged variables are logged when an event occurs! They can not be logged in an user specific order even if you changed the sorting of the table.
Log Level must be switched at runtime between the 0 (no logging), and 1 to 9 (logging the variable up to the defined level only) with the internal variable @LogLevel. The default value of the @LogLevel is set to 0 on the terminal, so it shall be changed to activate the logging. To change the log level variable you can use a numeric field or set the variable via Java Script. The other columns define the triggers for logging the selected variable. By clicking the editable cells they are marked as selected (like a check box). Multi select is possible.

⊞ Starting and Stopping the logging at runtime

The logging on the device is controlled by the variable [@LogLevel](#)[174] . By default, this variable has the value 0 (logging off). To start the logging in either mode, set the variable to a value >0.

Copy/Move the logged files

The copy/move is started with the variable @LogCopyNow.

Variable	Value	Description
@LogCopyNow	1	Copy current log File from @LogPath to @LogCopyPath
@LogCopyNow	2	Move current log File from @LogPath to @LogCopyPath
@LogCopyAllNow	1	Copy all log Files from @LogPath to @LogCopyPath
@LogCopyAllNow	2	Move all log Files from @LogPath to @LogCopyPath

Please note: When you start a copy / move process and the variable logging is already started, the variable logging is stopped. When copy / move are finished the variable logging is resumed. The best way to start a copy / move is to use a Java Script and bind it to an "onRelease" action of a button or softkey. The copy / move path could be also changed to runtime setting the internal variable @LogCopyPath. Please note that @LogCopyPath information (USB Stick / SD Card) is depending on the display you are currently operating. In some cases it could be that the path information will not met the actual pathes on the device. In such a case it is recommended to change the @LogCopyPath information at runtime to the right path. Please contact the support if you need further information.

Please note: If @LogCopyPath do not exist, the copy / move feature is not creating a folder.

Please note: The runtime is running on a full featured Linux system. In some cases the mount point of the external devices (SD, USB, etc.) are different from the default. In those cases it is needed to change the @LogCopyPath. The @LogCopyPath is not updated

automatically when plug in an USB stick. If you plug a USB after the device is powered on you have to set the @LogCopyPath accordingly.

Variable	Default
@Log Path	/opt/pclient/logfiles
@Log CopyP ath	/disk/usbsda1

Use the @LogCopyProgress to show the overall progress when copy / move the log files to an external memory. Connect this variable to a linear bargraph and you have a nice progress bar.

To copy and move files, we are using a sync write to avoid data loss, when the user un-plug an external storage. This is more secure than a usual write, but is slower, too. The reason why using sync write is simple: If the progress bar shows 100% you can be sure that the write progress is nearly finished. Please wait for 2 seconds before unplugging the external memory.

For a better usage we introduced two new variables: @LogCopyFinished and @LogCopyError. The first variable goes to 1 when the internal copy / move process is finally finished. You will see on the device that the progress bar shows 100% and the finish variable will update somewhat later (< 1 second).

The second variable shows an error code. If everything is ok the Error Code is 0. The error code table show the detailed codes.

Everything OK:

Variable	Expected value
@LogCo pyProgre ss	100
@LogCo pyFinish ed	1
@LogCo pyError	0

Something went wrong:

Variable	Expected Value
@LogC opyProg ress	<= 100
@LogC opyFinis hed	1
@LogC opyError	> 0

Please note: After the copy process is finished the external device is not unmounted from the file system, but you can unplug without an explicit unmount or eject. Please wait for 2 seconds after the copy/move finished.

To make sure the USB stick is mounted correctly, you can use a Java Script like this for your copy/move event:

```
var usb_mounted = getVariableValue("@USBMemStatus"); //check
if a USB stick is mounted

if (usb_mounted == 1)

{
    var usb_path = getVariableValue("@USBMountPath"); //get
path of mounted usb stick
    setVariableValue("@LogCopyPath",usb_path); // Set the
path of the USB stick
    setVariableValue("@LogCopyAllNow",1); // Copy LogData to
the selected path (move is 2 instead of 1)
}
```

Allowed Tags for file header and line prefix

Description	Tag
Year	<YYYY>
Year short	<YY>
Month	<MM>
Day	<DD>
Hour	<hh>
Minute	<mm>
Seconds	<ss>
Milliseconds	<s/10>
List of variables	<LIST_OF_LOGGED_VARIABLES>
Any string variable content	<variable_name>

Changes from older version

If you have an older version it is necessary that you load the project with the new version and save it once. Only new data is added all other settings stay as they are. There is one major change when you previously used "New file at restart". With this option the maximum number of files is set to 10. That means if you have more than 10 files on your system and you update the GUI, the oldest file will be deleted and a new one is created.

Error codes

Code	Explanation
16	File Error. Can happen in two ways. First a file was not found or something went wrong during writing on external storage

Code	Explanation
20	The already used disk space plus the new file will exceed the reserved disk space (half partition). The oldest File will be deleted. (only for "new file at restart")
25	The available disk space is not sufficient for the next new logFile. The oldest File will be deleted. (only for "new file at restart")
41	No Directory. This happens if one of the directories set for copy / move are wrong or do not exist. Or if target and destination are the same. Note: Directories are not created if they are not existing!
42	Directory empty. Nothing to copy, because source directory is empty.
43	This code shall not occur in normal operation. If you see this code it could be that the operation is started twice. In that case the operation is terminated.

3.12   Project Search

The program has a very powerful project-wide search feature.

General

The search can be accessed:

Manually:
1. by pressing CTRL-F
2. by clicking into the menu Window - Project-Search...
3. by clicking into the menu Edit - Project-Search
4. by using Variable View/Project-Tree - Variable Context-Menu - Find Usage of Variable
5. by using Project-Tree - JS-Node - Context-Menu - Find Usage of JavaScript

Automatically:
1. During load of Project - if the loaded project has warnings
2. If a variable should be deleted and the variable is used in JS- files or in other DDO-objects/CAN mappings. The search will show its usages.
3. If a JS file should be deleted and the script is used in other JS files or in DDO-objects/CAN mappings. The search will show its usages.

The default position of the project search component is a tab next to the project tree tab.
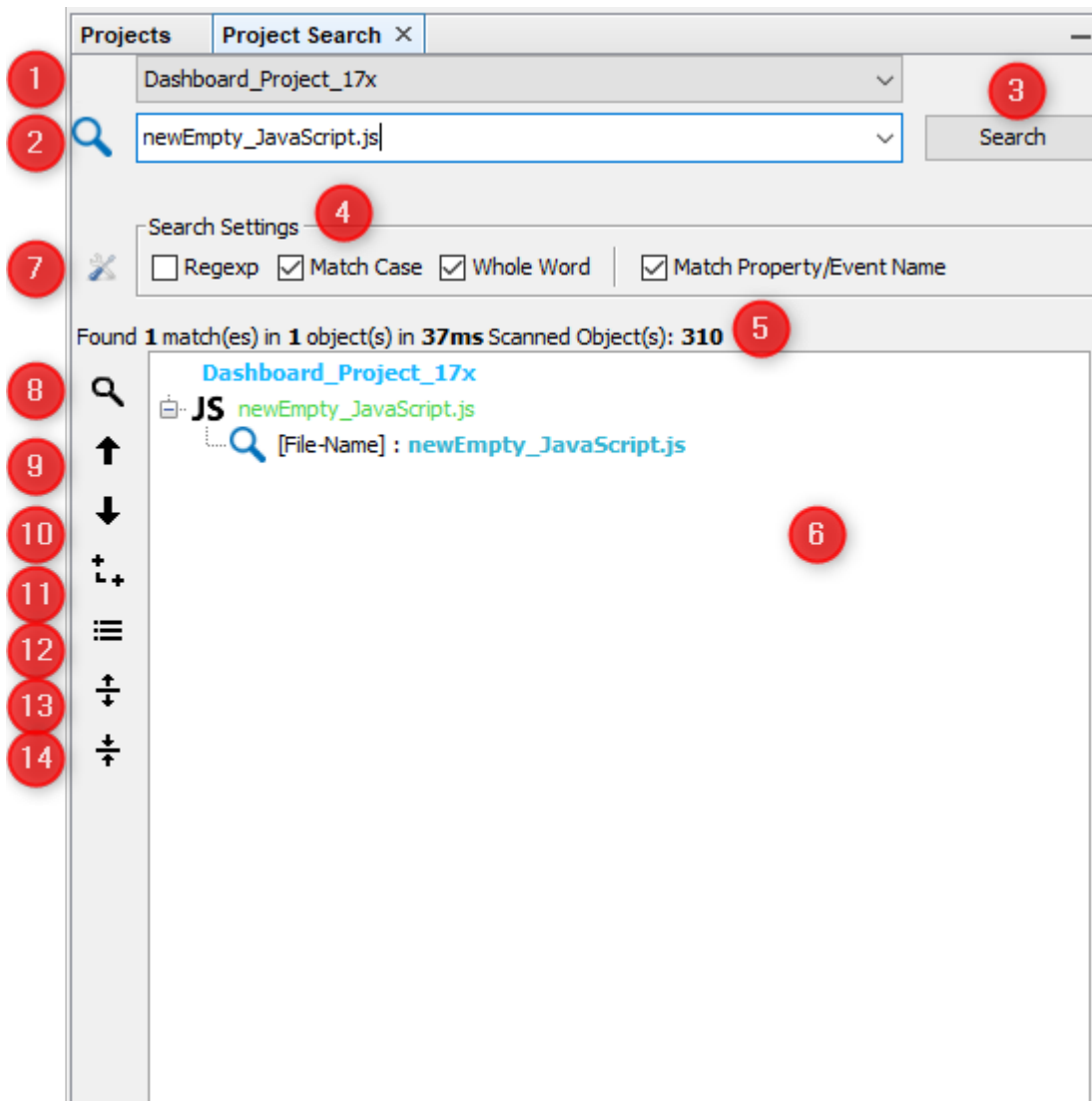
1. Combo-Box: list of all opened projects
2. Text box/Combo box: search command
3. Button: search
4. Check boxes: search settings
5. Label: search details
6. Tree: search results
7. Latching button: show all search settings
8. Latching button: show detailed search results. (property/event matches)

9. Button: go to previous search result node (with rollover)
10. Button: go to next search result node (with Rollover)

11. Latching button: show search results as a tree (similar to project tree)
12. Latching button: show search results as a list

13. Latching button: expand search tree
14. Latching button: collapse search tree

Explanation of the elements

1. Combo-Box: list of all opened projects
   - List is **empty** if no project is available
   - List is **always enabled**
   - The **selected project** is used for the search
2. Text box/Combo box: search command
   - Combo-Box is **always enabled**
   - Combo-Box contains the **last 10** search commands
   - The last 10 search commands will be **saved and restored** after start of Projektor.
   - If the search command is invalid the border will be painted in red and an info message will be shown below the combo-box with detailed information.

   - If a searched was performed and the appropriate project is closed and no other opened project exists, a message will be shown that the project was closed and the search-tree(6) will be cleared.
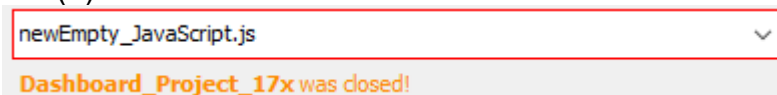
   - If more than 5000 search hits are found the an error is shown. This is currently necessary because otherwise expanding the tree takes too much time.

3. Button: search
   - Button is only enabled if at least one opened project exists and the search command is not empty
   - The Button starts the search and the results are shown in the search tree(6)
   - The search can also be started if the Enter key is pressed while the search command text field has focus.
4. Check boxes: search settings
   - Regexp: Use Regular-Expressions for search command
     Default-State=Deselected
   - Match Case: Lower/Upper case must match
     Default-State=Deselected
   - Whole Word: The search expression must be a separated word.
     Default-State=Deselected
   - Match Property/Event Name: The named Property/Event must match exactly => example is shown later
     Default-State=Selected
5. Label: search details
   - Found objects and found matches, search time and count of all searched objects => Only if at least one match was found

   - No Matching objects found! search time and count of all search objects => if no match was found

6. Tree: search results
   - Shows the Project-Node of the current selected project and the search results

- If no match was found only the Project-Node is visible.
- If no opened project exists the following message is shown: No current search!
7. Latching button: show all search settings
    - Shows/hides the detailed search settings panel
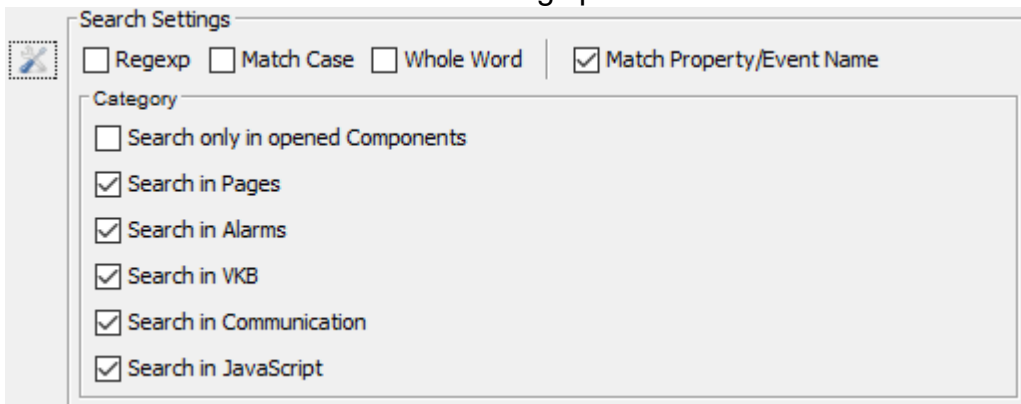
- Default states:
  All check boxes are selected except of the first
  If a check box will be deselected, the according project part won't be considered for the search.
  Pages does also include MiniView[576] -Pages.
  The Project itself with its properties/events will be always considered for the search.
- If the first check box is selected (Search only in opened components) the other check boxes will be deselected and disabled
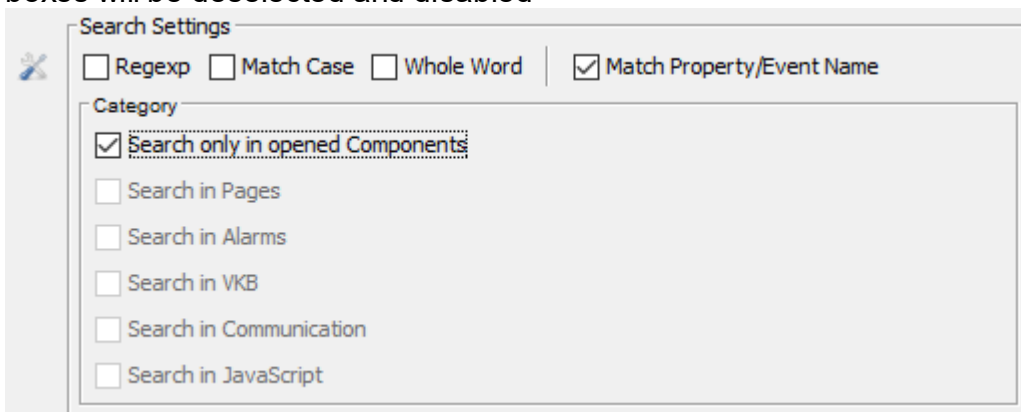
- If the first check box is deselected the other check boxes will be selected and enabled again.
- If **no** check-box is selected a **warning** is shown
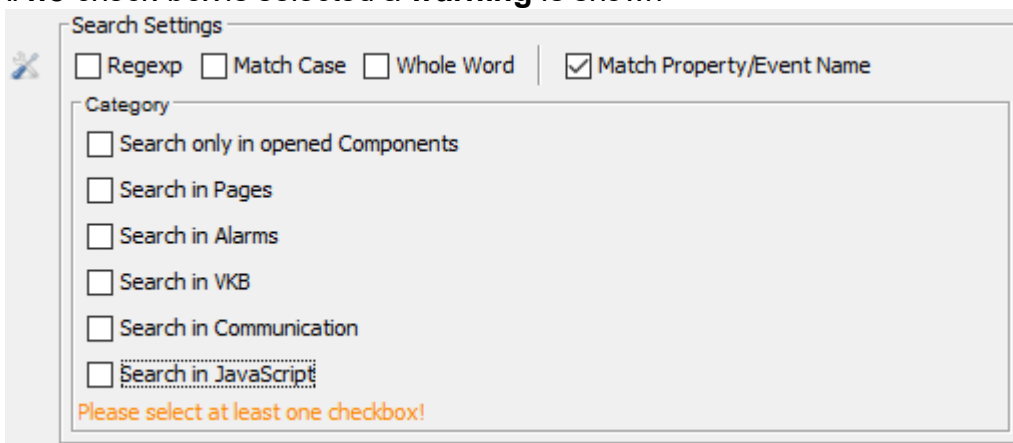
8. Latching button: show detailed search results. (property/event matches)
   - Shows/Hides the detailed search results (Default is Enabled/Selected)
   - Enabled

- Disabled

9. Button: go to previous search result node (with rollover)
   - Selects the previous search result object and opens the appropriate page/alarm/VKB and selects the object in the scene.
   - For communication objects nothing special happens
   - For JS files the file will be opened.
   - If currently the first object is selected an the button is pressed, it jumps to the last object in the list (roll over)
10. Button: go to next search result node (with Rollover)
    - same as for previous button, only the other direction
11. Latching button: show search results as a tree (similar to project tree)
    - Switches between List/Tree View (Default is List-View)

- Tree view:
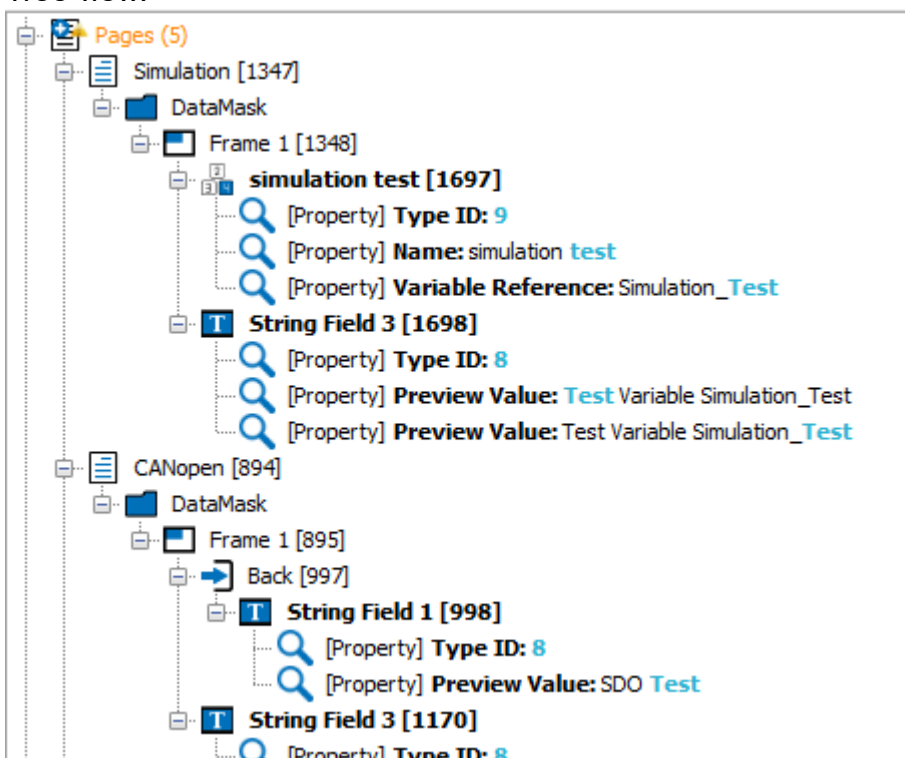
12.Latching button: show search results as a list
   - Switches between List/Tree View (Default is List-View)
   - List view:
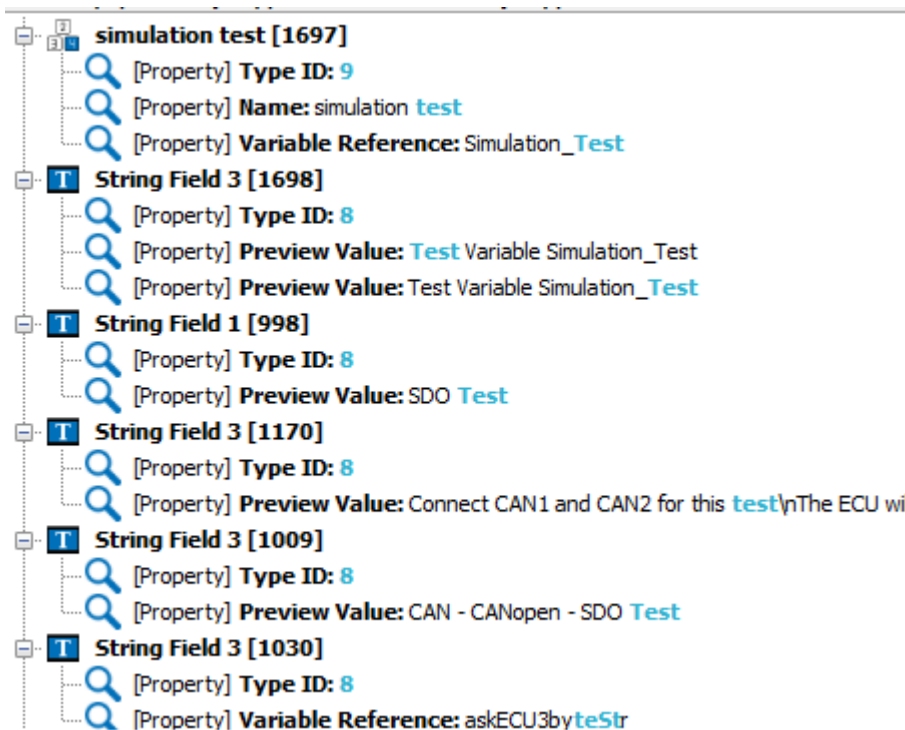
13.Latching button: expand search tree
   - Expands search tree (Default state: selected)
     Should only be used if tree view is enabled (makes no sense otherwise)
14.Latching button: collapse search tree
   - Collapse Search-Tree (Default State: Deselected)

Should only be used if Tree-View is enabled (makes no sense otherwise)
The tree will be collapsed, till the Pages
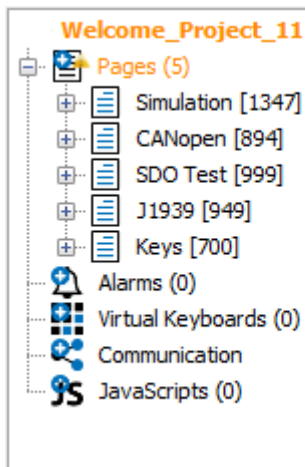
The search command

The Search-Command-Text-Box can be used in many different ways.
For the current examples the Welcome-Project with default search settings are used.

Possible syntaxes:
- Simple String (see Example 1)
- Property Search (see Example 2)
- Number (see Example 3)
- Combination of String and Property (see Example 4)

General syntax rules:
Property - Operator - Value
#[Property-Name]=Value

Additionally several search expressions can be connected with && or ||
#[Property-Name]=Value && #[Property-Name]<=Value

Additionally brackets **()** can be used:
(#[Property-Name]=Value && #[Property-Name]!=Value) || #[Property-Name]>Value

For property searches the following operators are available:

For Strings and Numbers:
- [Equal/Contain] **=** or **==**
- [Not Equal] **!=** or **<>**

Only for Numbers:
- [Greater] **>** or **>=**
- [Lower] **<** or **<=**

Example 1 => Simple String:

Search Command:
test

For the 1st example the simple word **test** is typed into the search command box.
The String **test** is now search in every property/event of all Objects of all Pages/Alarms and VKBs in the Welcome-Project. Beside it will be searched in all Communication Objects (Variables, CAN-Messages and Owners) and in all JS-Files.
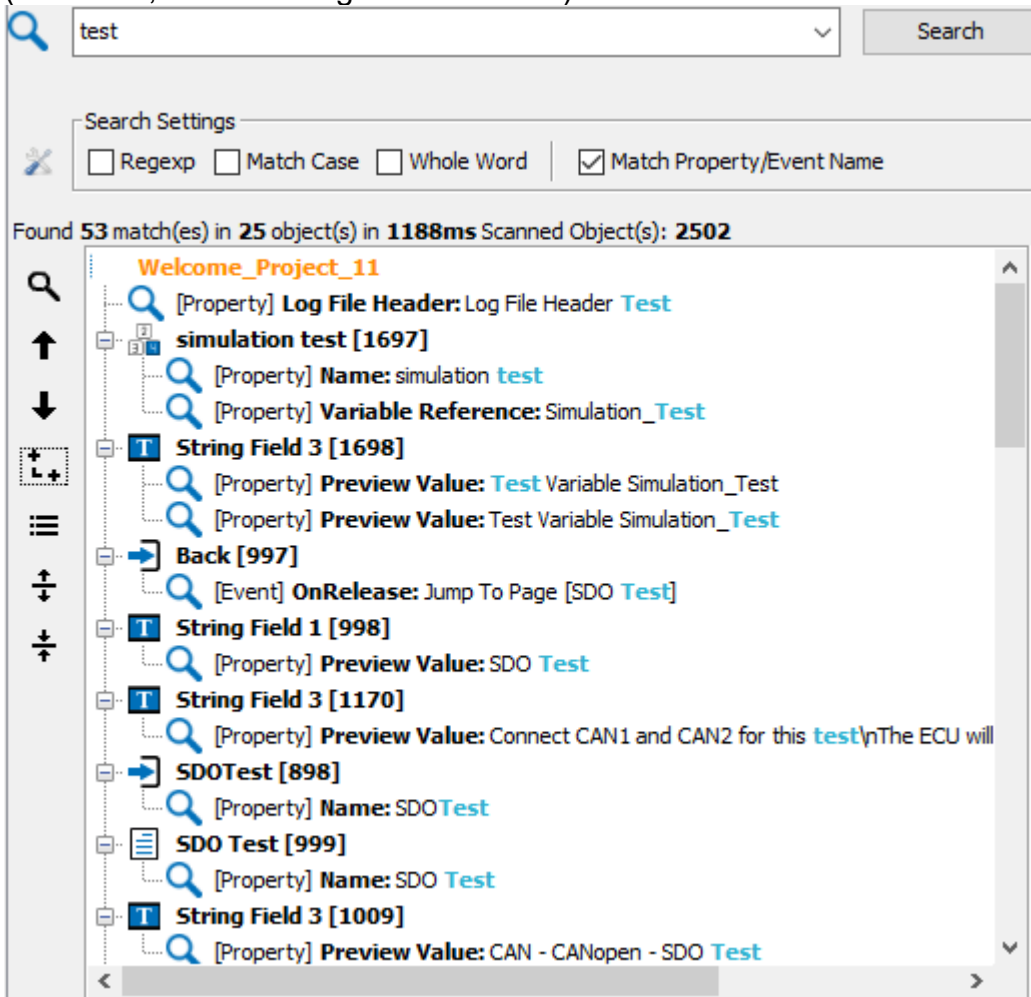
Example 2 => Property Search:

Search Command:
#[Name]=test

The 2nd example shows a search where the simple word **test** is searched only in Name properties. All other properties will be ignored.

The String **test** is now searched in every Name property of all Objects in all Pages/Alarms and VKBs in the Welcome-Project. It will also be searched in all Name properties of the Communication Objects (Variables, CAN-Messages and Owners) and in all JS file names.

If the property NAME is searched, the JS file names will also be considered. For all other properties the JS files will be ignored, because they don't have properties.
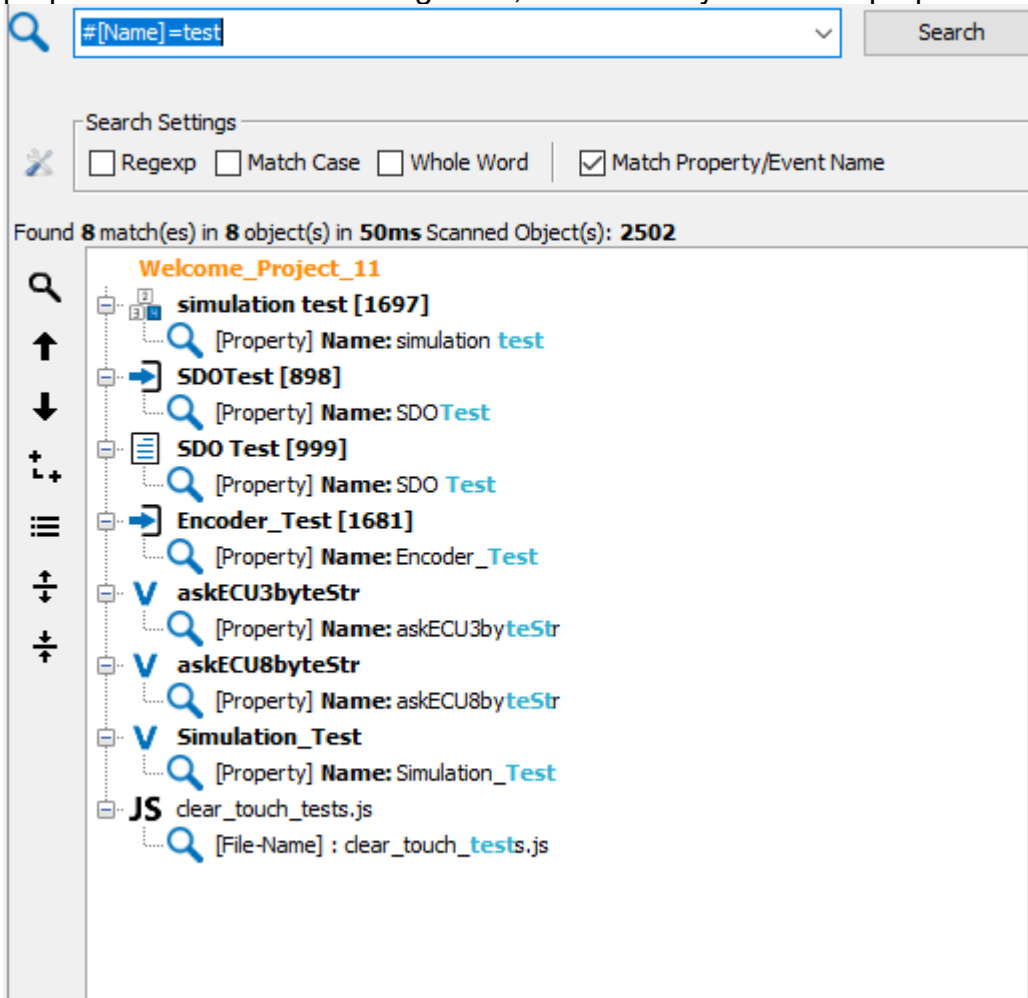
Example 3 => Simple Integer search:

Search Command:
12

For the 3rd example the number **12** is typed for the search.

The number **12** is now searched in every property/event of all Objects of all Pages/Alarms and VKBs in the Welcome-Project. It will also be searched in all Communication Objects (Variables, CAN-Messages and Owners) and in all JS files.

If the data type of the property is a number (Integer, Float, Double) the value must match exactly.
E.g.
Property: Width = 123 => would not match, but
Property: Height = 12 => would match

If the data Type of the property is a string the value must contain the string value **12**
E.g.
Property: Name = 1200 bits

=> would match
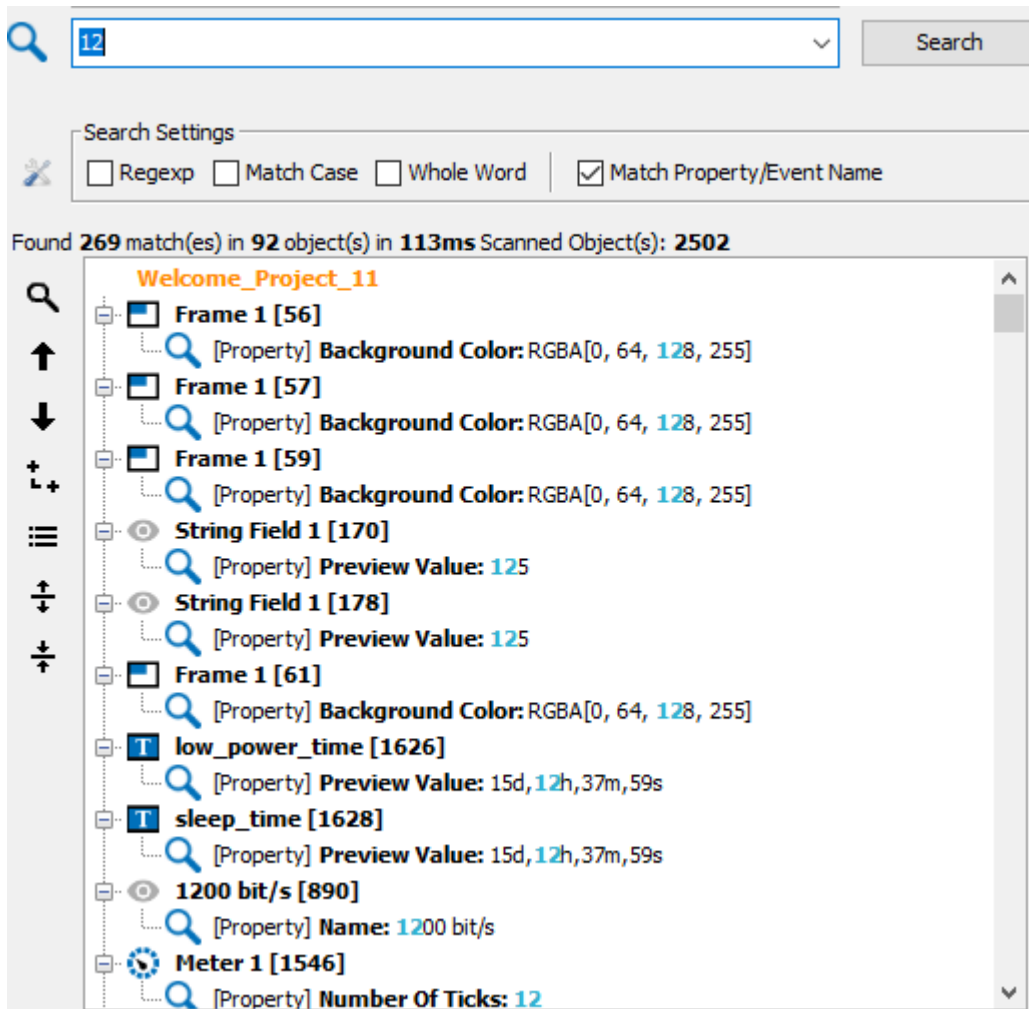
Example 4 => Connecting search commands

Search Command:
(#[Type ID]=8 || #[Type ID]=9) && test

In the 4th example it is searched for the word **test** again, but now the of the found object(s) must be 8 or 9 (String Field or Numeric Field)

The String **test** is now searched in all properties of all Objects in all Pages/Alarms and VKBs in the Welcome-Project. It will also be searched in all properties of the Communication Objects (Variables, CAN-Messages and Owners).
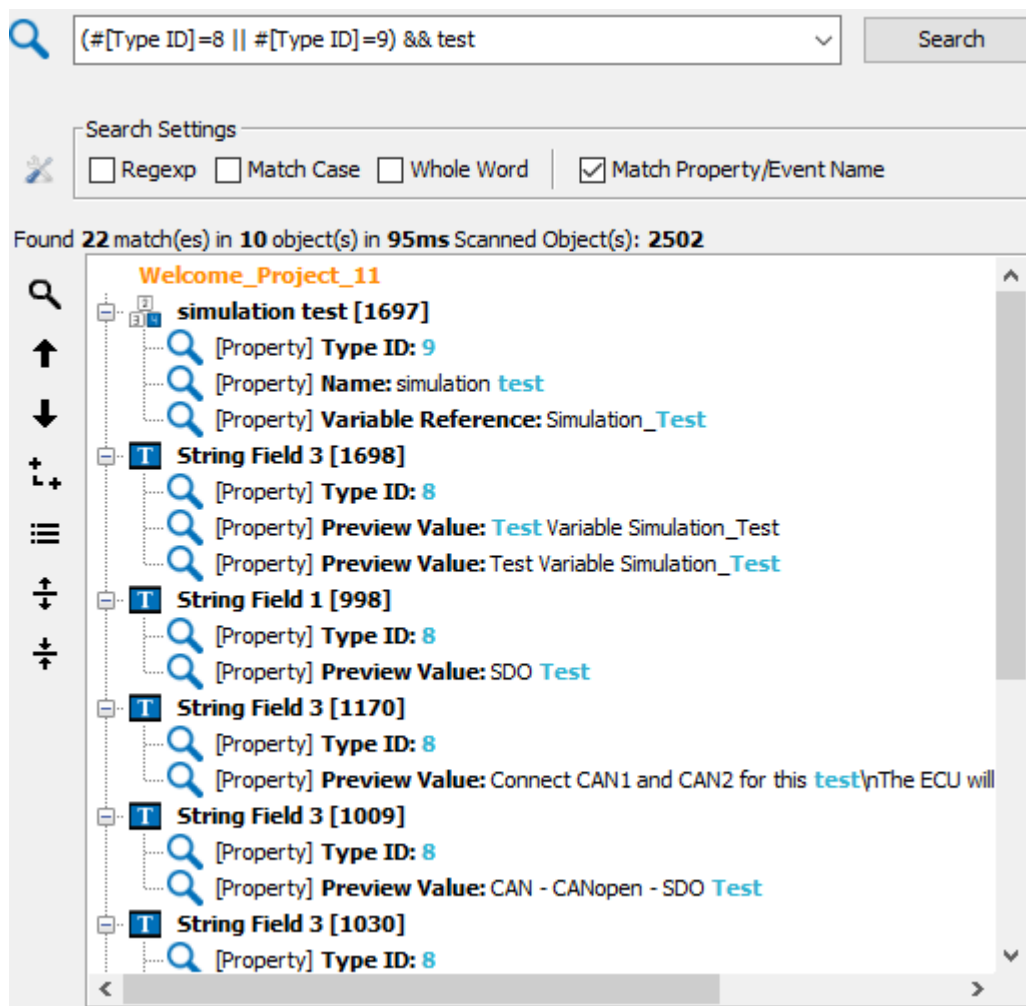
Note:

All properties and events can be used for the search.

Exceptions are only:
- Border Profile and
- List-Items

For the Project the following properties are searchable:
- General
- Object
- Alarm
- Variable Logging
- ISO-Bus
- Day-/Night

Currently it is not possible to search for properties of an event, e.g. the Property Periods (100 ms) of the event OnProjectRepeat.

Example 5 => Searching for all input objects on Page X

Open the page(s) you want to search in

Settings:
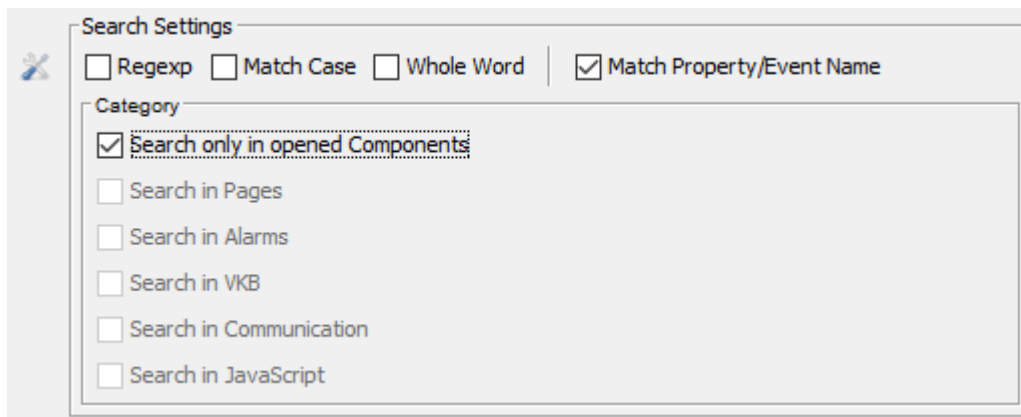
Search Command: **#[Set As Input]=true**

Example 6 => Exchange the Font property for all String Fields with DejaVu (bulk property editing)

Here it is searched for all String Fields that don't have language dependency that use the Font DejaVu

Search command: **#[Font]=DejaVu Sans && #[Type ID]=8 && #[Dependency]=None**

Now the Font property of all found objects can be selected, by right-clicking the Font property of one of the found objects:
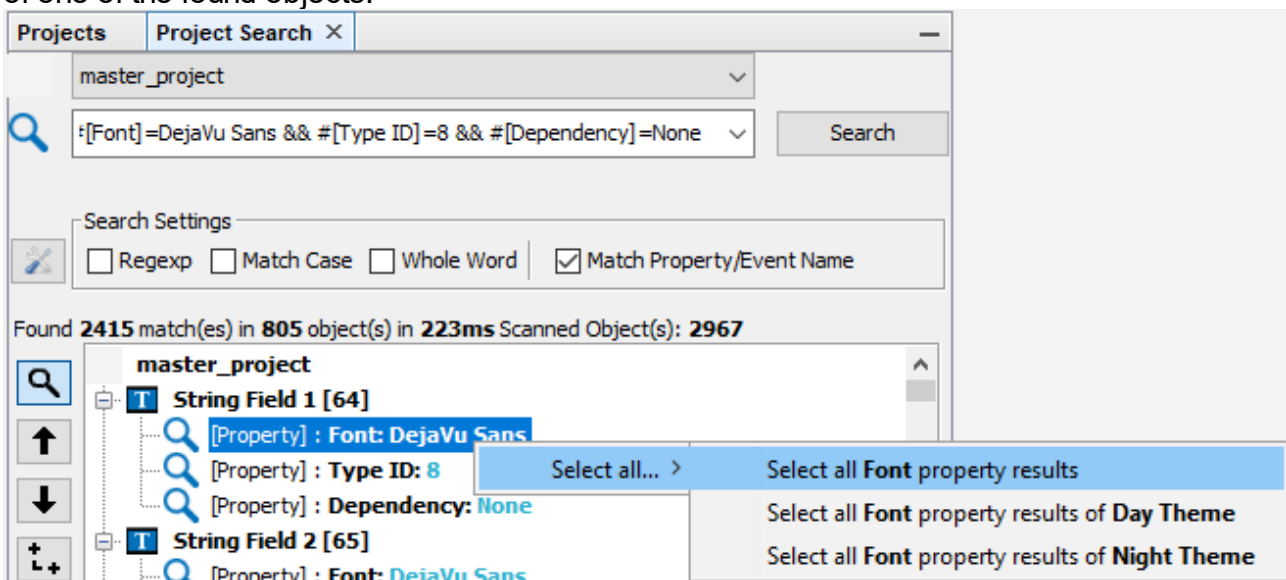
Note that either ALL objects that use this font can be selected, or only all results in day theme or night theme, respectively.
With clicking Select all Font property results, all the objects will be selected and in the property window the Font property can be changed for all objects at once:
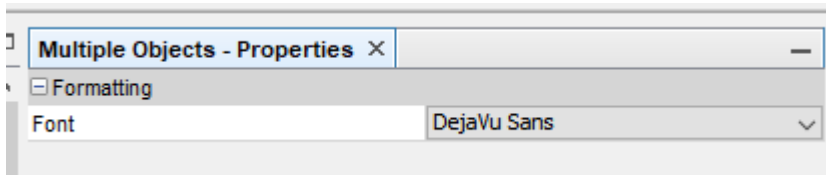
Tips & Tricks

Special commands for #():

- With the search command **#[Event]!=No Action** it can be searched for all configured events in a project.
- It is also possible to search for a specific event, e.g. OnPress: **#[OnPress]!=No Action**
- It is possible to search for special actions, e.g. for Jump to Page: **#[Jump To Page]!=No Action**
- With the search command **#[References]>1** it can be searched for all objects that have reference copies of itself in the project.
- The search command **#[Warning]=true** shows all objects that have a warning

Check box Match Property/Event Name

- If the check box is activated, the Property / Event name must match exactly. If e.g. #[name] is typed instead of #[Name], the search will not find the property.
- If the check box is de-activated, the lower and upper cases will be ignored and the above example would match with the property Name. Internally, the String method contains() is used, which means, that a search with **#[ame]** would also work.

Keyboard Shortcuts

If the search result tree has focus, the following keyboard shortcuts can be used:

- **F3** - go to the next search result node
- **Alt + F3** - go to the next detailed search result node
- **F5** - go to the previous search result node
- **Alt + F5** - go to the previous detailed search result node

- In all cases, for graphical objects, when double-clicked, the appropriate page will be opened with the object selected.
- If **F5** (or **Alt + F5**) is used for JavaScript search results, the JavaScript file will be opened and the matching part will be selected.

Result order

The result order of the search results can be changed for list view by opening the context menu of the project node in the search tree.
The following search orders are available:
- Projekt-Tree (Default)
- Name (Ascending, Descending)
- ID (Ascending, Descending) -> only makes sense for objects that have an ID

Update a property for several objects

If e.g. the font property can be updated for all String Fields on a certain page with these steps
1. Open the page where the fonts shall be changed and close all others.
2. Go to the detailed search settings and select the check box **Search only in opened Components**
3. Type the following text in the search command line (to find all String Fields with font DejaVu that don't have a dependency): **#[Font]=DejaVu Sans && #[Type ID]=8 && #[Dependency] =None**
4. Open the context menu of a font search detail node and choose **Select all Font property results**
5. Go to the property sheet and select a different font.
The changed font will be updated in the search tree.

Other important things to know:

FindUsage

For FindUsage of Variables and Find Usage of JS File the following gui elements will be updated automatically in the search component:
- The appropriate project of the variable / JS file will be set
- The variable name will be set in the search commend text box
- Match Case and Whole Word will be selected
- Selection to search in Pages/Alarms/VKBs, Communication and JavaScripts is activated

Updating a property / event of an object which is visible in search tree

If a property value is updated which is present in the current search tree, the following things happen:
- the old value stays visible and the new value will be shown with green font
- an attention icon will be added in the upper right corner of the search icon (magnifier) of the updated property
- an updated icon (asterisk) will be added to the parent object of the updated property
- If the value is reverted to the original value, all named changes will also be reverted

Deleting an object present in the search tree

If an object is getting deleted which is present in the search tree, the following things happen:

In list view:
- Object stays visible in the list view
- The icon is updated to a red cross (which is the symbol for a deleted object)
- The property/event sheet is not visible anymore

In tree view:
- The deleted object will be removed from the tree.

3.13   Project Versioning

Sometimes users would like to have an old state from some months ago of the project to do some edits for a special customer who shouldn't get the latest changes.
Sometimes it happened that a project had an error and could not be opened anymore in the program.

For these use cases and others we are introducing a simple, yet powerful project versioning system.

We used JGit, a Java library implementing the widely known Git version control system.

Please note that we did NOT implement branches or merging of different project revisions.

How can I start using this feature?

New projects

In the new project dialog, an option appears in the second screen.

Panel	OPUS A3 Standard ⌄
	☐ Use ISOBUS Application
	☑ Enable project versioning
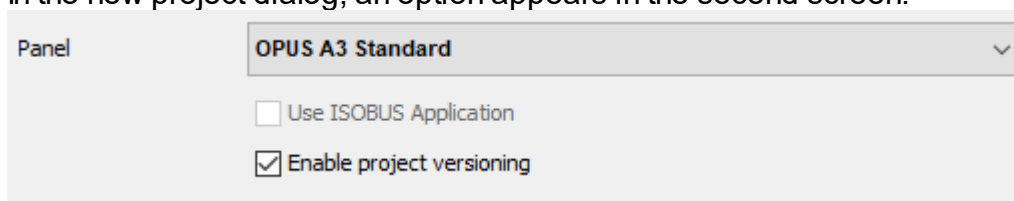
If this check box is checked (default setting), project versioning is enabled for this project.

Existing projects

For existing projects, the versioning will not appear upon project opening.
When you save the project for the first time, this dialog will appear.

Initialize repository?   ✕

The project has been saved.

The project versioning feature is activated, but the current project was not intialized for the version control system. Do you want to initialize it?

Please select:
- "Yes" to continue the versioning process.
- "No" to cancel the versioning process.
- "No & Disable" to cancel the versioning process and disable the feature for this project.

[Yes]   [No]   [No & Disable]

Choose yes to start using the versioning system for the project.
Choose No to not start using the versioning system for the project at the moment, i.e. you will be asked the same question at the next save.
Choose Cancel not start using the versioning system for the project and disable versioning for this project, i.e. you will not be asked this question again (but you can enable the versioning again in the project properties).

How can I turn versioning on / off later on?

Versioning can be turned on / off in the project properties of each project.
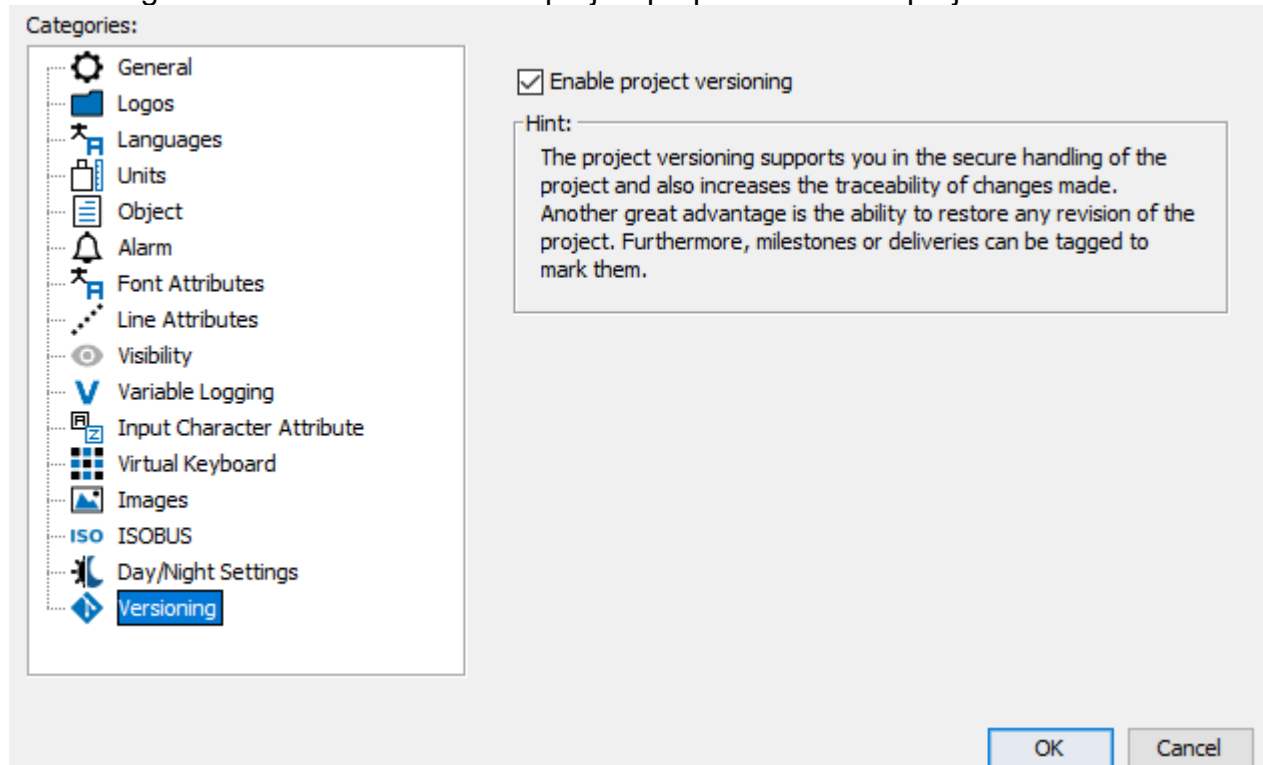
If versioning is turned off, the existing revision data is kept, so if versioning is enabled again, all existing revisions are still available.
It is also possible to still open the revision browser to see all existing revisions. Tags can also still be managed.

What happens in my project folder?

If you enabled versioning for a new or existing project, a folder and a file will be added to the project folder.

In the .git folder all necessary information and data for the project versioning is located.
Please don't edit this folder or anything of its content in any way and don't delete it.
The file .gitignore states some files that are excluded from the versioning. These are files that are not necessary for project loading.
At the moment, these files are listed in there.

- /terminal_files.zip
- /imported_fonts.zip
- /log.0
- /log.1
- /log.2

As with the .git folder, please don't edit or remove this file.

How can I access the different stages of my project?

When opening a versioned project, you will see a dialog like this:
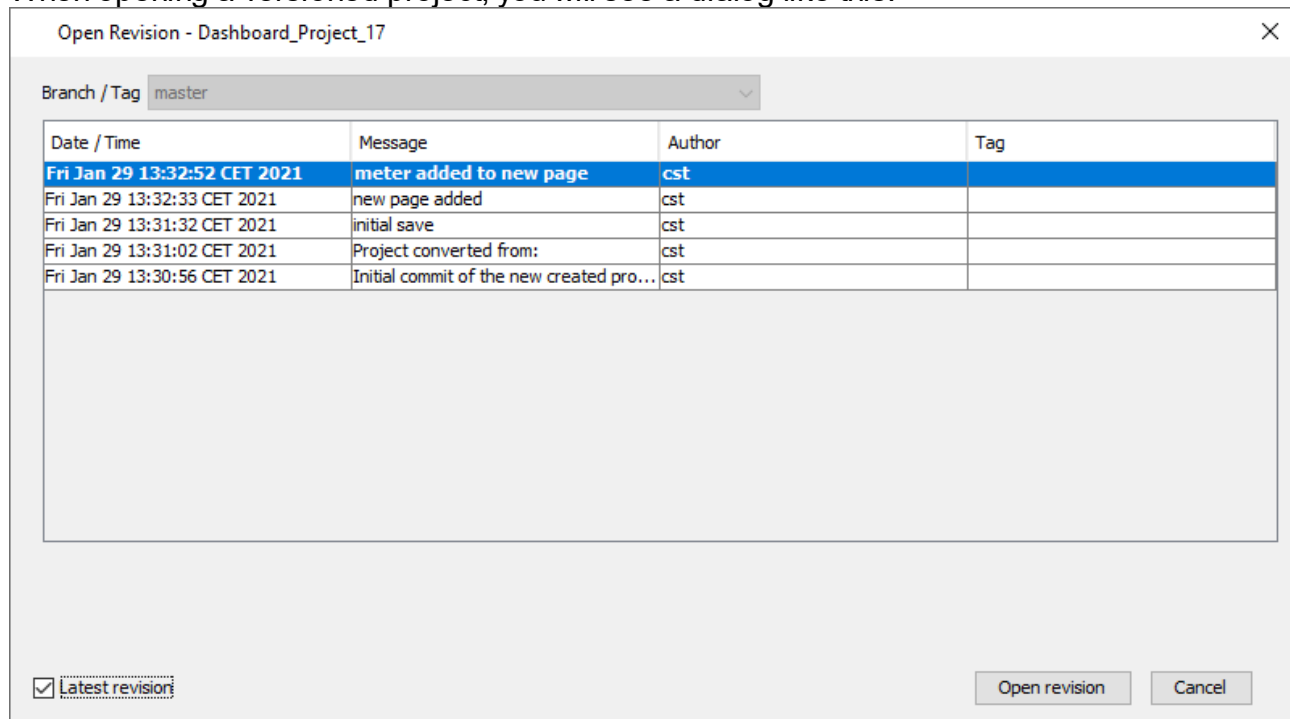
The dialog shows the different revisions of this project. A revision is a certain state of the project, defined usually by a project save or a conversion.
If you just click "Open revision", it will open the latest revision of the project (probably the most important use case).
If you want to access one of the older revisions, you need to uncheck the "Latest revision" check box. Then you can select any of the revisions and load these by clicking "Open revision".
Cancel stops opening the project.

Will I lose newer stages of my project if I open an old revision?

If you open an older revision and then try to save the project, this dialog will appear:
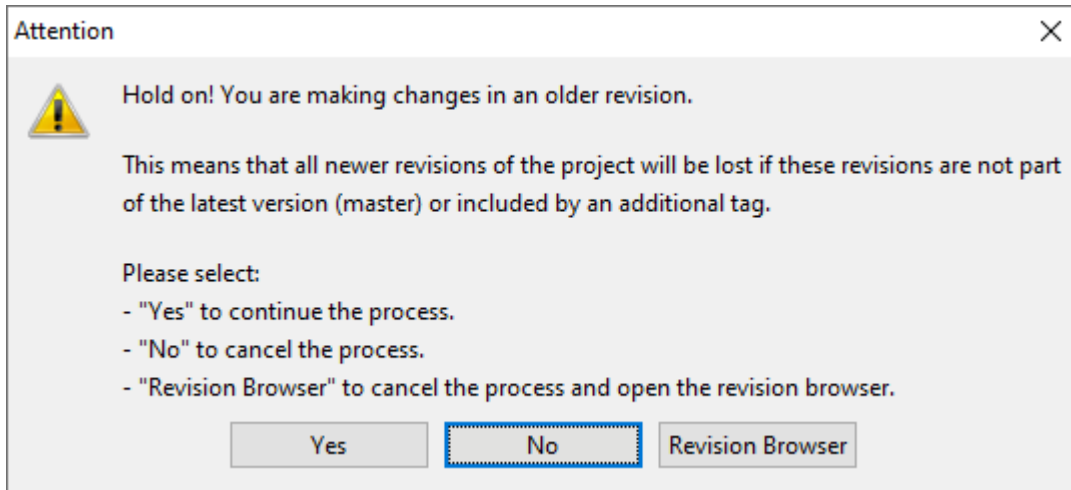
So yes, if you click yes now, all newer revisions of the project will be deleted (unless they have been [tagged](#)[127]).

If you click No, nothing happens. If you click Revision Browser, the revision browser will be opened and you can manage your revisions.

How do I use the revision browser?

You can access the revision browser by clicking on the toolbar item or through the menu item Tools -> Revision Browser

In the revision browser the different revisions of the currently selected project can be viewed. Additionally, tags can be created, moved and deleted.

What are tags?

If you open an older revision of a project, make changes to it and try to save it, you will be warned that all newer revisions will be lost if the save is completed.
However, it might be interesting to conserve certain project states, e.g. as delivery states as they went to the customers.
For this purpose tags can be used.
Tags can be added to any project revision in the revision browser.
Additionally, a tag can be set in the commit dialog that appears during a project save.

Note: If you move a tag, it is possible that revisions are deleted, if they are not part of that tag, another tag or the latest version ("master").

What is this dialog that appears upon saving the project?

When a versioned project is saved, a revision commit dialog appears. Here the user can add a comment to a commit and choose or create a new tag.
A click on Commit Revision commits the currently saved revision to the versioning system.
Cancel cancels the commit process.
Note: The project has already been saved at that point. The save process is not canceled by clicking Cancel here.

Can I use versioning but not have this pesky dialog at each save?

It is possible to enable auto-commit in the [options]^{72}.
Note that enabling this means that no commit messages can be added to project revisions which can reduce clarity.
It is still possible to add tags to project revisions in the revision browser.

When do I risk losing a revision?

The versioning system only deletes revision that do not have a reference to the "master" (i.e. the latest state of a project) or to a tag.
Let's look at an example:

A first save of a project

Revison 1 (Master)

Some edits to the project, delivery to a customer

Revison 1   Revison 2 (Master)
                   Tag
                   Delivery_2021_02_0
                   3

More edits to the project

Revison 1   Revison 2   Revison 3 (Master)
                   Tag
                   Delivery_2021_02
                   _03
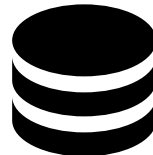
At this stage, revision 1 and 2 are referenced by the tag "Delivery_2021_02_03".
Revisions 1, 2 and 3 are referenced by the latest revision ("master").

Now we load Revision 1 of the project, make changes to it and save the project (agreeing to the warning that is shown when committing the save).
What happens?

Tag "Delivery_2021_02_03"

Revison 1   Revison 2

Tag
Delivery_2021_02_0
3

"Master" branch

Revison 1   Revison 4 (Master)
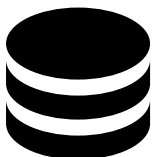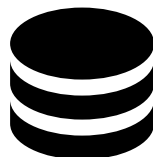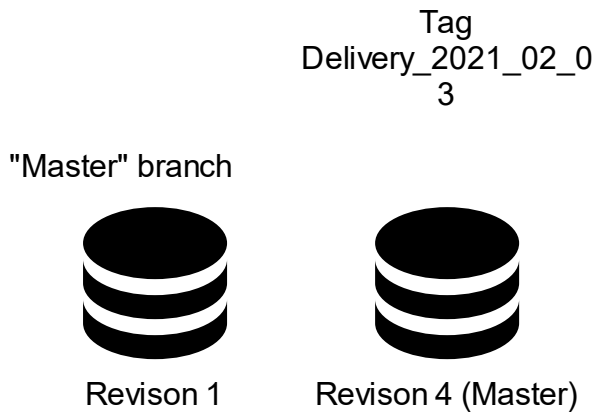
Note that by doing the above work flow, revision 3 was deleted, because it did not have any reference, not by the tag "Delivery_2021_02_03" nor by the "master".

My save process was interrupted, my computer crashed or something, my project is broken, what do I do now?

When opening the project, uncheck the "Latest revision" check box and load the second revision from the list (assuming that the first one is the one that is broken).

3.14   Sample Projects

In the first step of the dialog to create a new project, it is possible to use a Sample project instead of the standard empty project.
Sample projects have pre-configured content. They will be created for the chosen device automatically. The layout may not always be perfect. Most sample projects will look better in landscape orientation.
The number of sample projects included in the program may vary.

User Sample projects

Users can also create their own sample projects as a starting point for new project.
To create a sample project, select a project in the project tree and click the menu File -> Create Sample
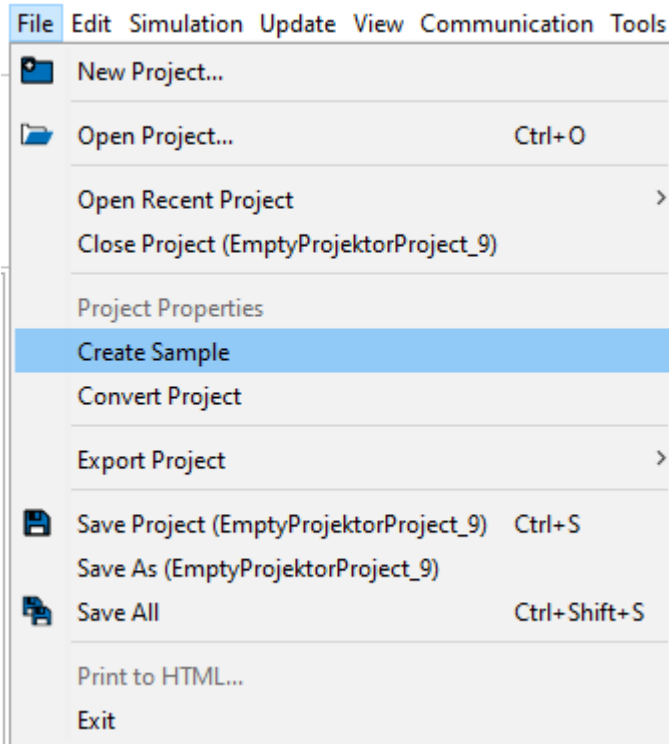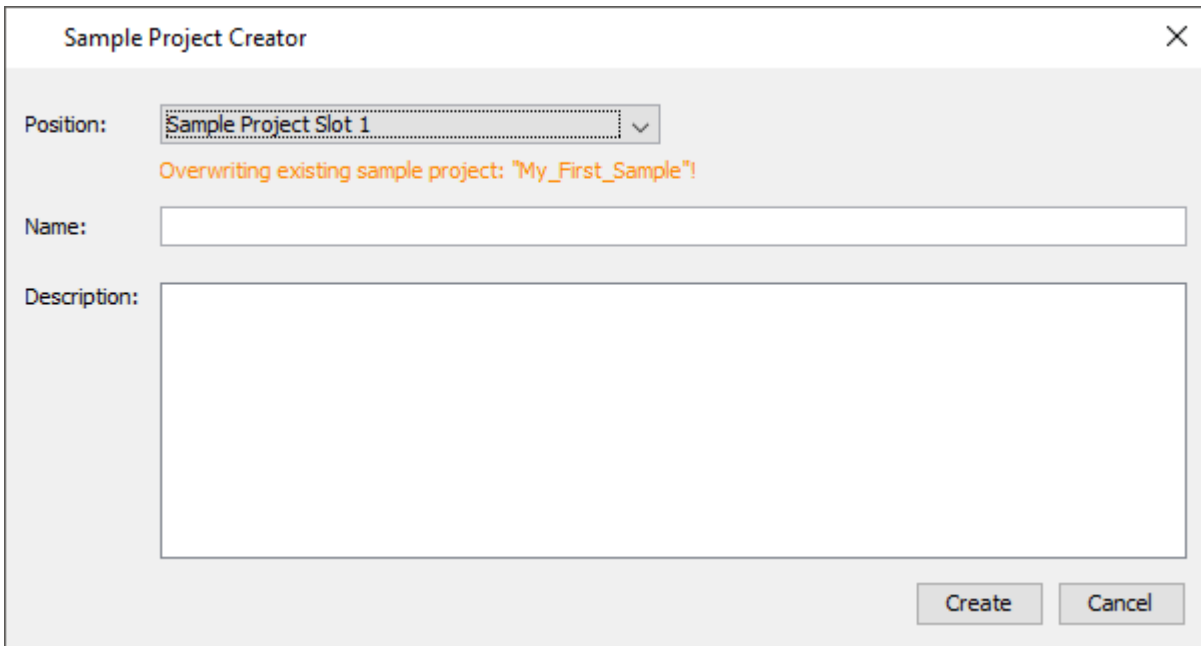
In the dialog, choose one of the sample project slots, give a name and, optionally, a description, and click **Create**.
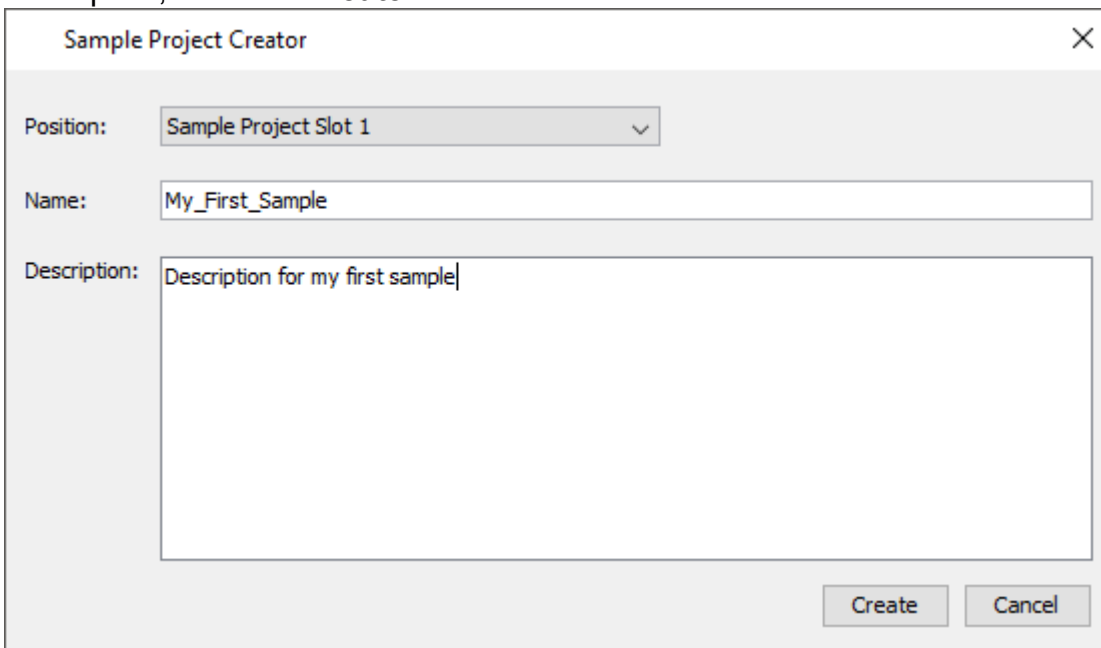
The creation of the sample project will be confirmed in a dialog.
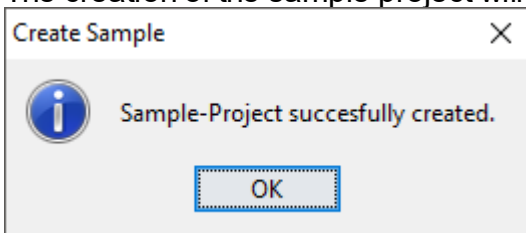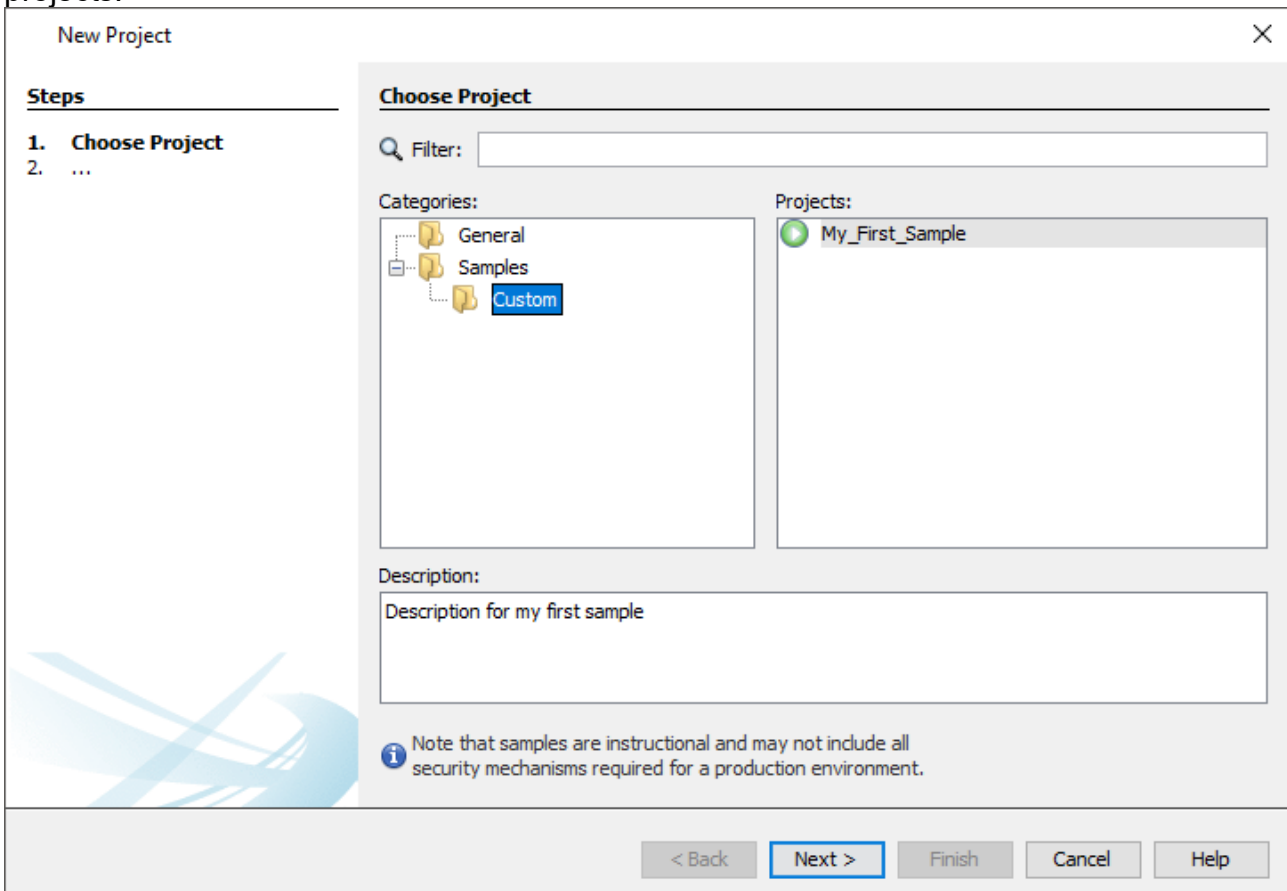
If a custom sample project slot is already used, the dialog will warn that the old sample project will be overwritten.

Now, a Custom folder is displayed in the new project dialog that contains the custom sample projects.

Please note: Depending on the exact version of the program, it is possible that a backup of the project is made when a sample project is created. This backup is created because of technical reasons. It can be deleted.

Tradeshow Dashboard project
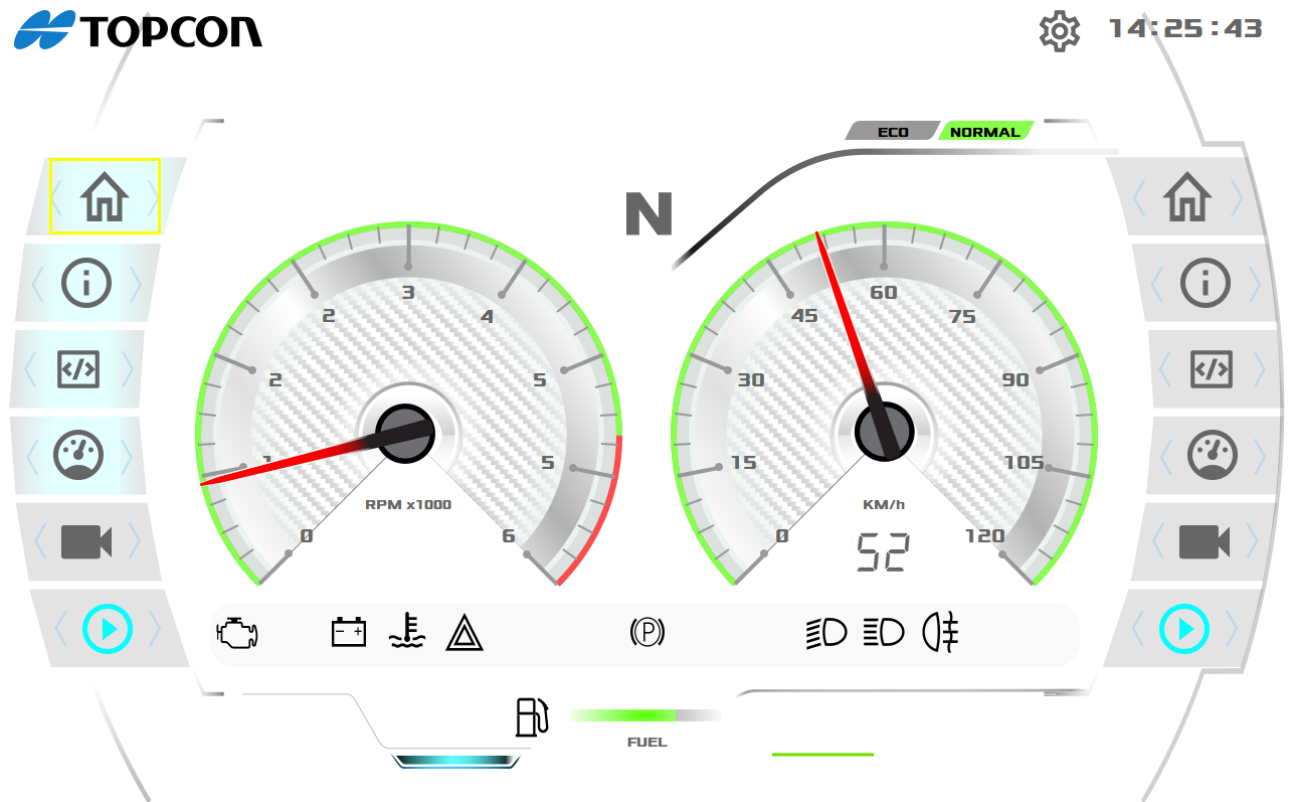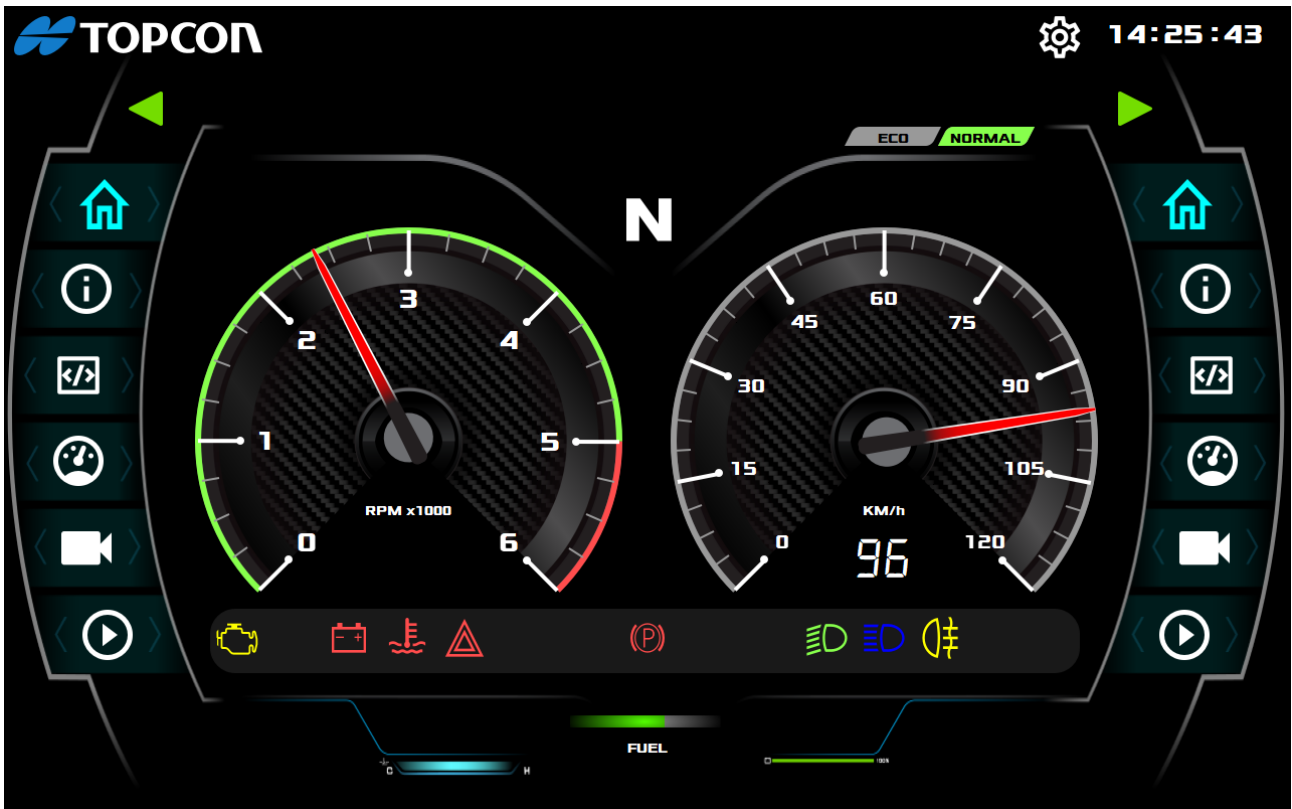
This project is a ready-to-use dashboard for a trade show, complete with moving needles and blinking icons. There are 6 pages, with a hardware, a software programming information screen, a dynamic gauge to see the flexibility of our objects, a camera page and a multimedia player that can play back a video. The project works for all devices, with the main focus on the B-series.

3.15   Start Page

The new Start Page will show the most important information in one tab.
In the Welcome tab you will see the recently opened projects in the top left.

In the top of the right column you will see the RSS feed (Latest News) which will notify you of Projektor updates and other important information. Don't worry, we will not spam you.
The feed refreshes every hour. If there are any problems reaching our server, a notification will be shown that there was an error loading the feed. Additionally, a reload button appears. At this point the feed will not refresh automatically. Only when the feed can be loaded again after reload has been pressed, the automatic refresh will start again.

Below the RSS feed, a Tip of the day feed is shown. Here you can find a random tip regarding the OPUS Projektor and the PClient and our devices. Some might be trivial to you, as an expert, but some will surely give you a tiny bit of new and helpful information.
The feed refreshes every hour.If there are any problems reaching our server, a notification will be shown that there was an error loading the feed. Additionally, a reload button appears. At this point the feed will not refresh automatically. Only when the feed can be loaded again after reload has been pressed, the automatic refresh will start again.

3.16 Symbol Library

We offer three sets of symbols in the OPUS Projektor:

2425 ISO 7000 symbols
4391 Material Design symbols
1433 Remix symbols

Material Design Symbols

These symbols can be used for free.
The files were modified from its original state to be used with the blend color feature.
The original files are available here.
The files are distributed with the Apache license 2.0 and the SIL Open Font license.

Remix symbols

These symbols can be used for free.
The files were modified from its original state to be used with the blend color feature.
The original files are available here.
The files are distributed with the Apache license 2.0.

ISO 7000 Symbols

We are offering a collection of over 2400 symbols from the ISO 7000 catalog to be used within projects.
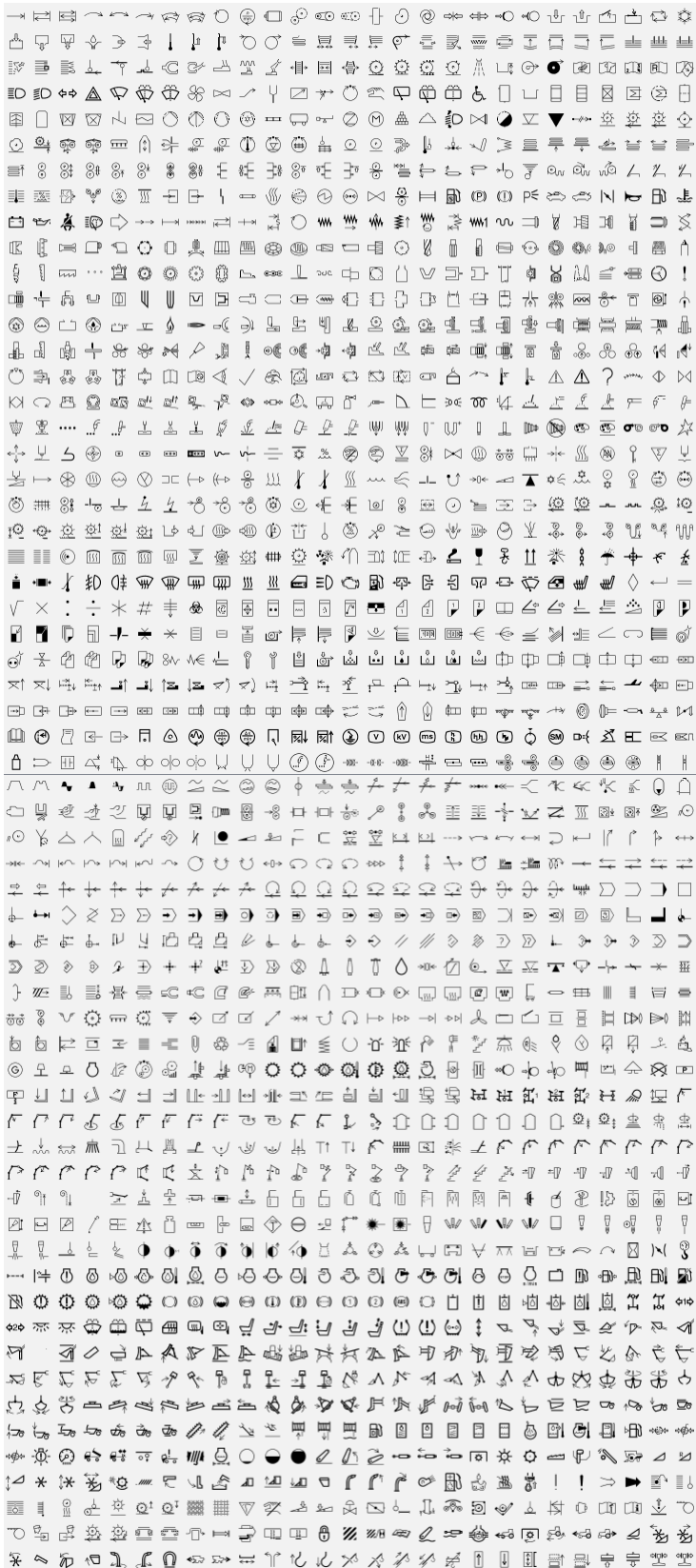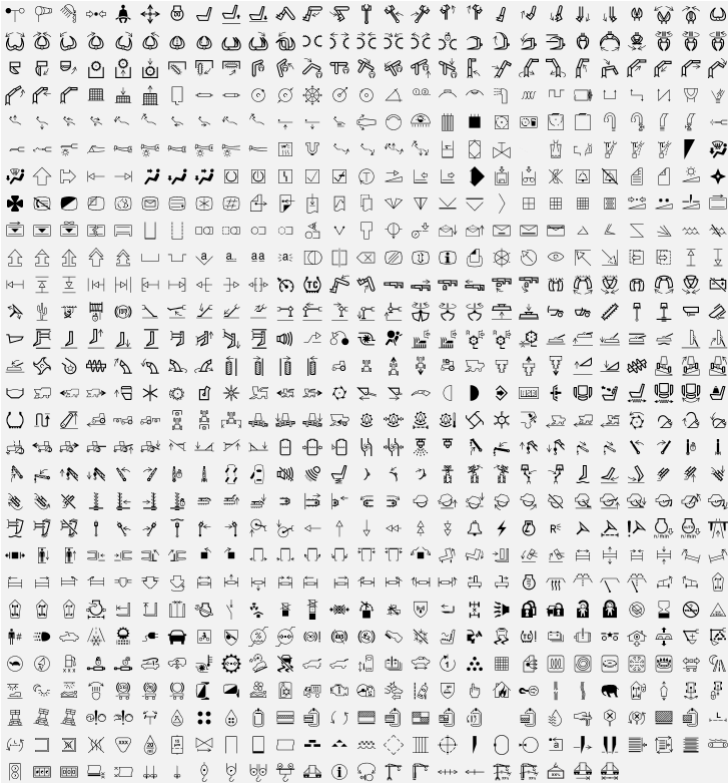
You are not allowed to use these symbols outside of projects for our devices!

The full symbols package has to be purchased. Please contact your vendor for more information.

General

You can access the symbols by opening the Symbol Library toolbar on the right side (in the default view).
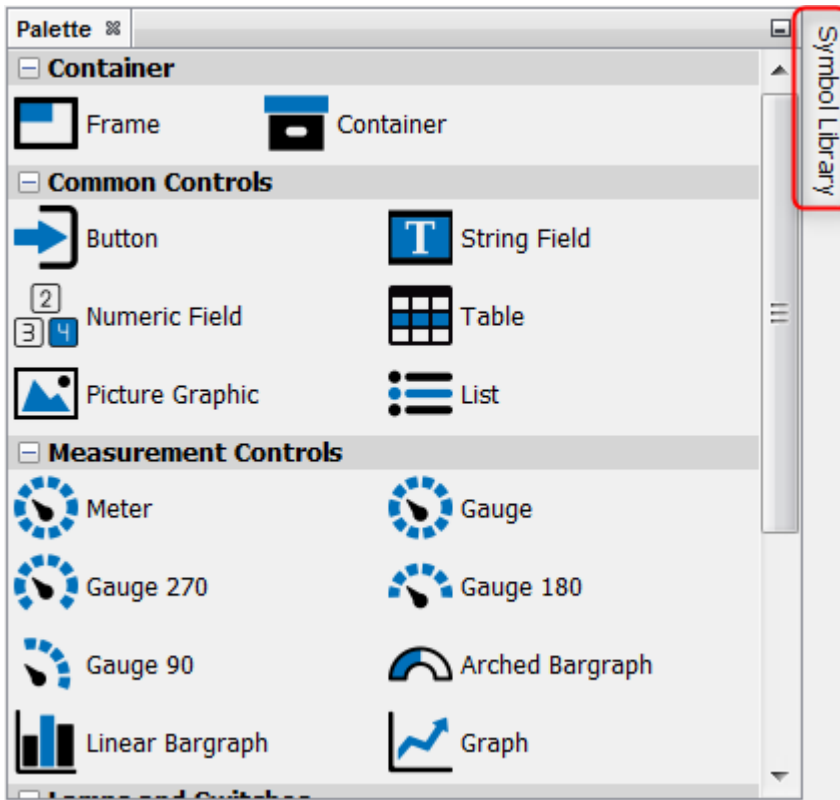
You can also (re-)open the window by clicking the menu Window -> Symbol
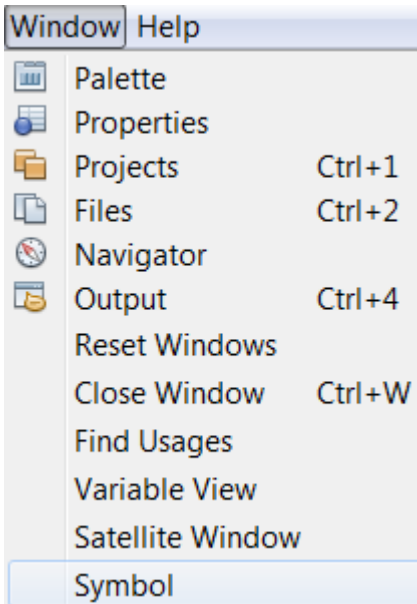
When the library is opened for the first time after the program was started, it takes a while to load the preview images.
You will see a loading pop-up
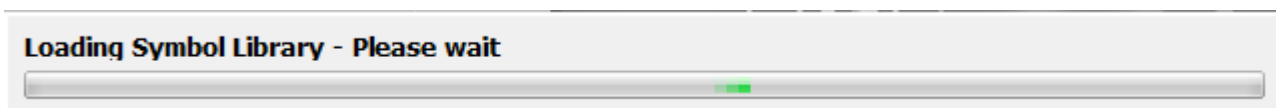
and you can see the loading progress in the lower right corner of the program.

Symbols: 384 of 2426   16%

Once the previews are loaded, they are instantly available as long as the program stays open. Changing the zoom level requires a reloading of the preview symbols.

UI elements

In the bottom of the symbol library window there are some UI elements
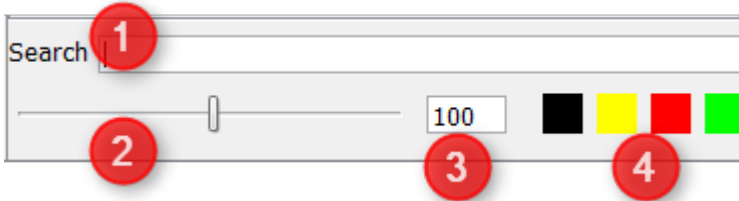
1 - Search bar. Here you can enter search words to filter the symbols. The symbols will be filtered live and all images containing the words will be shown in a new tab called Search. The search will go through all available symbol sets.
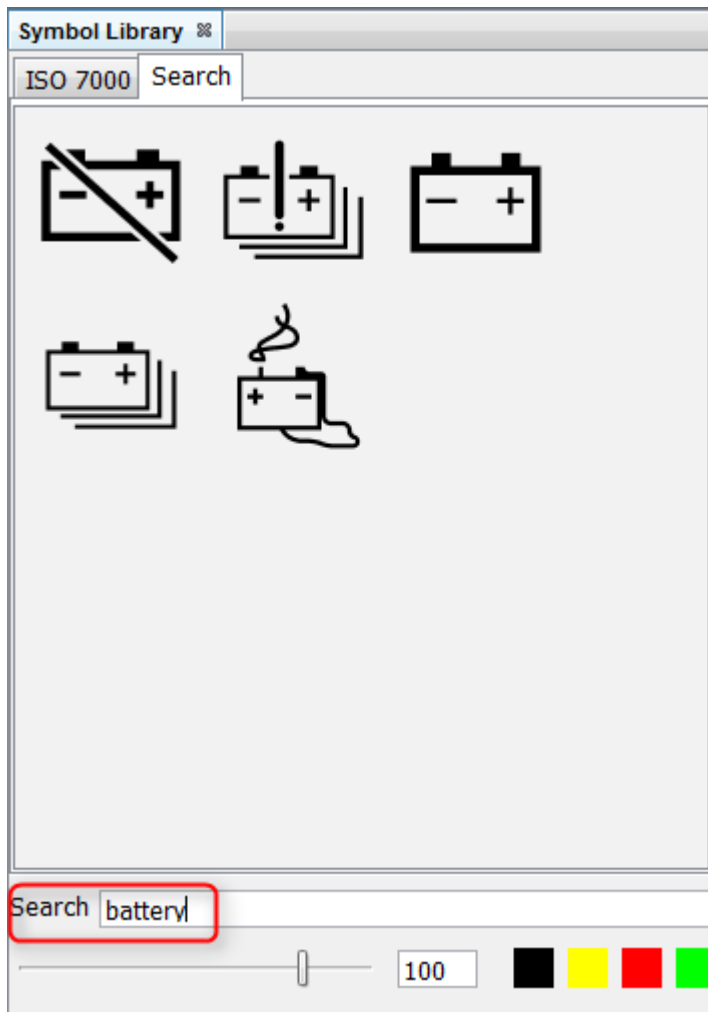
2 - Zoom slider. Here you can change the zoom level of the preview images in the symbol library window. Sliding to the left shows more smaller preview images, sliding to the right shows less larger preview images.

3 - Image size. Here you can set the size of the image in pixels, with which the image will be created. The image will be square, so the size will be X times X. This is only true if the image is not used within another object, e.g. as a background image. Then the size of the parent object determines the size of the image.

4 - Blend colors. Here you can set 4 different blend colors for the creation of the images in your project. The first color is the color of the object in most cases. See here[141] for the use of the other colors.

Using the symbols

You put the symbols into your project with Drag & Drop[43]. This works both on the scene as well as on the project tree.
Depending on where you drop the image, you will get a context menu with different options how to use the image, e.g.

Frame:

Add Picture Graphic Object
Add Picture Graphic(s) Into New List ▸
Background Image
Border Image
Video Image

Meters & Gauges:

Background Image
Border Image
Needle Image
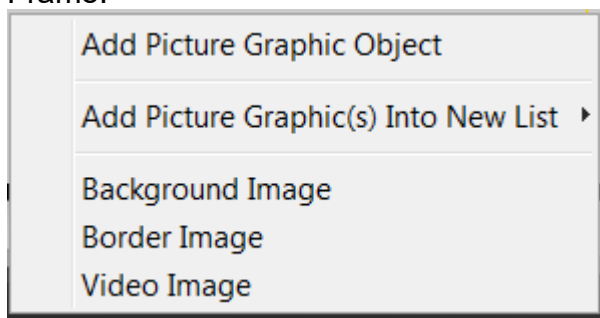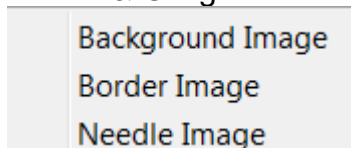
Buttons:

Add Picture Graphic Object
Add Picture Graphic(s) Into New List ▸
Background Image
Border Image
Pressed Image
Released Image
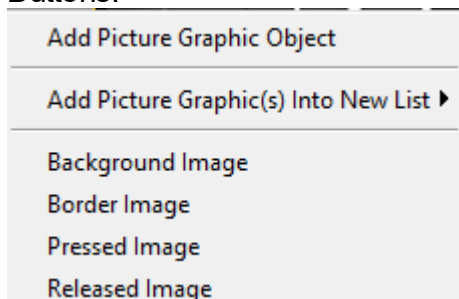
Add Picture Graphic Object - This will create a new Picture Graphic Object with the symbol image as the Image To Display
Background Image - This will use the image as the background image of the object that it was dropped on

Border Image - This will use the image as the border image of the object that it was dropped on

Video Image - This will use the image as the placeholder image for a video frame

Needle Image - This will use the image as the needle image of the object that it was dropped on

Pressed Image - This will use the image as the pressed state image of the button that it was dropped on

Released Image - This will use the image as the released state image of the button that it was dropped on

Add Picture Graphic(s) Into New List

When dropping a symbol image on a frame or container, the option Add Picture Graphic(s) Into New List is offered.

Opening the sub context menu shows 3 different options:
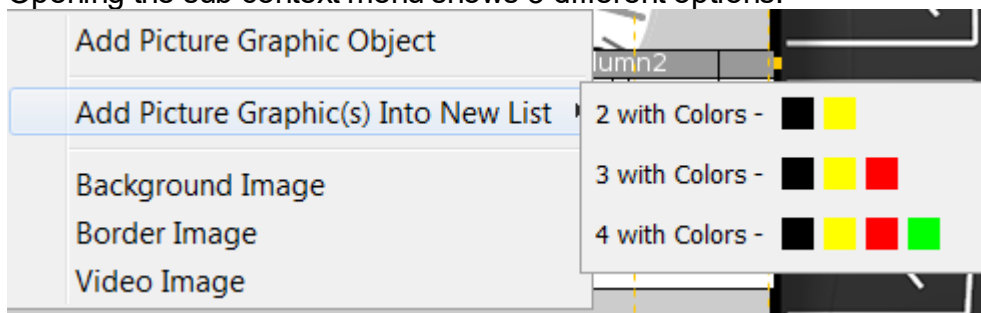
These options will create a new list object[477] with 2 to 4 instances of the symbol with up to 4 different colors. The colors can be set in the UI interface of the symbol library window by clicking on one of the 4 color fields and choosing a color.
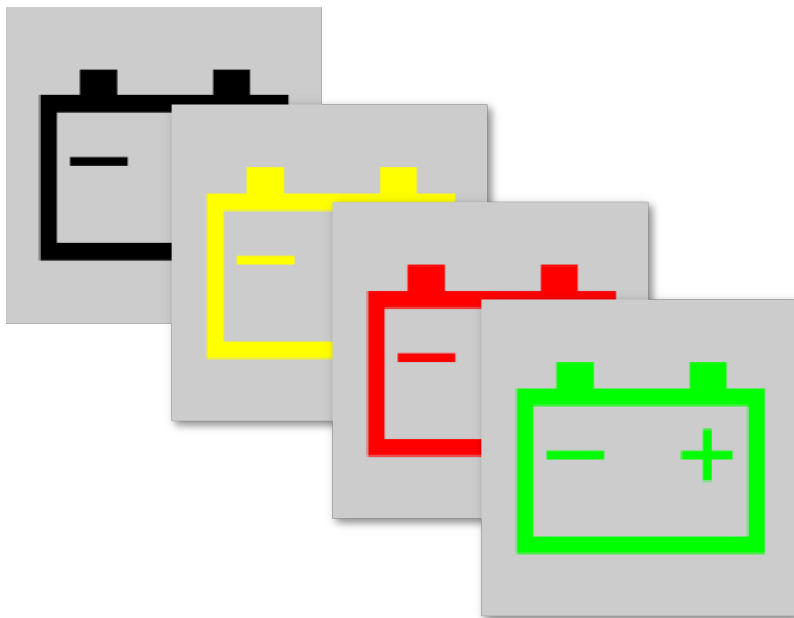
Please note that the blend colors are project-independent and will be saved and kept even after the program was closed.

You can NOT change the color(s) of symbol images at runtime!

Favourites

Symbol images can be set to favourites to be found easier.
This can be done by right-clicking on a symbol image and choosing Add To Favourites in the context menu:
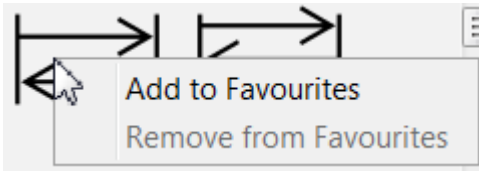
This will create a new tab Favourites where this and all other images set to favourites are displayed.
Images can also be removed from this tab by right-clicking and choosing Remove from Favourites.

Please note that the favourites are project-independent and will be saved and kept even after the program was closed.

3.17 Undo/Redo

Undo/Redo can be triggered in several ways:

- Keyboard shortcuts
  - CTRL-Z (undo)
  - CTRL-Y (redo)

The program needs to have focus for this to work. It doesn't matter which component in the program has focus. The step is performed in the currently active project.

- Toolbar

Press the arrow to the left to undo, press the arrow to the right to redo. Undo is only enabled when at least one action has been performed. Redo is only enabled when an undo step is selected.

- Menu Edit -> Undo/Redo
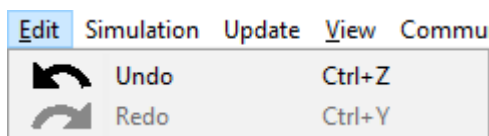
- Undo/Redo list in Undo/Redo window
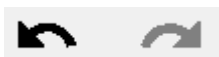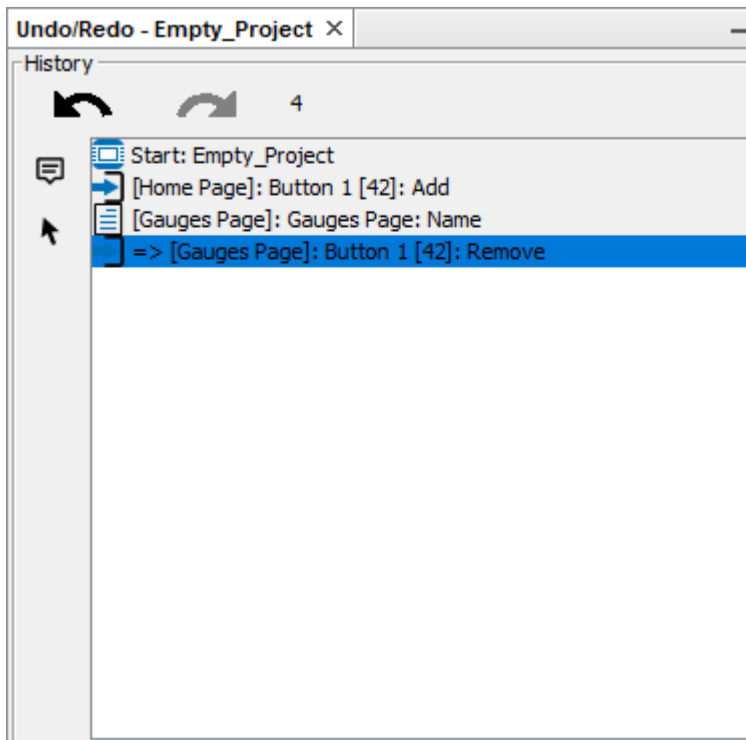
 The undo/redo buttons are only enabled if actions were done (undo) and undo steps were selected (redo).

 The number next to the undo/redo buttons show how many steps were recorded since starting the program / opening the project.

 This toggle button enables/disables tooltips of undo entries in the undo history list

 This toggle button enables/disables mouse navigation in the undo history list

The undo history list always starts with "Start: <project name>".
All modifications in the project create an entry in this list. If you happen to find an action that does NOT create an entry even though it should, let us know.
The list is currently limited to 1000 entries. If the 1001st action is made, the first entry in the list will be deleted and so on.
The following actions create an entry in the undo history:

- Graphical Object changes (all objects in the project tree) in the category:
  - Pages (MiniView-Pages)
  - Alarms
  - VKBs
  - All Property/Event and Move/Resize changes
- Communication Object changes (all objects in the project tree) in the category
  - Communication (Add/Remove/Update Communication objects)
  - All Property/Event changes
- JS changes (all JS files in the project tree) in the category
  - JavaScripts (Add/Remove/Rename/Update JS files)

- Project-Properties/Project-Event changes
- All dialog changes which modify the project
  - Simulation-Settings
  - Variable-Manager
  - Terminal and Owner(ECUs) Configuration...
  - Port Configuration
  - Configure Mappings Dialog
  - CAN-Freestyle: Transmit-Sequences
  - CAN-Freestyle: CAN-ID Masks
  - J1939: DM1/DM2 Settings
  - Import Variables
  - Import DBC-File
  - DataType definitions
  - Process Configuration
- Current/Default Language/Unit changes
- Hide/Show object changes
- Day/Night Mode changes

Each list entry contains 3 elements:

1. Icon
2. Presentation Name
3. Tooltip

Icon
- For graphical objects the icon of the object is used which was updated.
- For communication objects the icon of the appropriate object is used
- For JS changes the JS icon is used.
- For Project-Property changes, the project icon is used.
- Simulation-Settings the PClient icon is used
- For Day/Night Mode changes the Day/Night Icon is used
- For CAN-Mapping changes the Mapping Icon is used.
- For Protocol changes the Network icon is used.
- All other changes uses the Project icon

Presentation Name
- Graphical Objects
  - Property-Changes: [<Page-Name>]: <Object-Name>: <Property-Name>
  - Event-Changes: [<Page-Name>]: <Object-Name>: <Event-Name>
  - Move-Changes: [<Page-Name>]: <Object-Name>: Move
  - Resize-Changes: [<Page-Name>]: <Object-Name>: Resize
  - Add/Remove: : [<Page-Name>]: <Object-Name>: Add/Remove
- Communication Objects
  - Property-Changes: [Project]: <Object-Name>: <Property-Name>
  - Event-Changes: [Project]: <Object-Name>: <Event-Name>
- JS-Objects
  - Add: [JavaScripts]: <JS-FileName>: Add
  - Remove: [JavaScripts]: <JS-FileName>: Remove

o Rename: [JavaScripts]: <New-JS-FileName>: Rename
- Dialog changes
  o Update: [Project]: <Dialog-Name>: updated

Tooltip
- Graphical Objects
  o Property/Event changes: Old-Value and New-Value is shown
  o Move changes: Old-Location and New-Location is shown
  o Resize changes: Old-Location/Size and New-Location/Size is shown
- Communication Objects
  o Property/Event changes: Old-Value and New-Value is shown
- JS-Objects
  o Rename: Old-Name and New-Name is shown
- Dialog changes
  o Common: No specific informations
  o DBC-Import: Import-Count of Variables/CAN-Mappings/ECUs is shown
  o Variable-Import: Variable-Count is shown + Variables-Names (only 10)

Note: The Time-Stamp (TS) is always shown in the tooltip for all Undo-entries.

JavaScript Undo-Edits
Like mentioned before, all JS changes on the File System can be reverted by Undo/Redo.
Each JS (and each version) will be saved in folder <Project-Folder>/tempScripts
The folder tempScripts will be deleted when the project is closed.
The JavaScriptHandler creates MD5 sums of all JS-Files and saves each version in tempScripts folder (the syntax is: <js_name>.js_<md5>).
It doesn't matter if a JS file is added/removed or updated in the project or on the FS, the JavaScriptHandler tracks all changes and creates the appropriate Undo-Edits in the UndoRedoHandler.
The JS files will be scanned each second on the FS for updates.
If a JS files is opened in JavaScript-Editor, the editor will be closed if the js file gets deleted by the UndoRedoHandler. (This can take a few seconds ~2s)
If the UndoRedoHandler restores an older JS file version, and the JavaScript-Editor of the restored file is opened, it takes at least 2 seconds until the editor is refreshed.

Note: If a JavaScript-Editor has focus and Ctrl+Z or Ctrl+Z is pressed, the UndoRedoHandler of the JavaScript-Editor will be called (not the Projektor-UndoRedoHandler)

Graphical-Object Undo-Edits
- Pages/Alarms and VKBs
Creating/Deleting a Page/MiniViewPage/Alarm/VKB creates an Undo-Edit.
If the UndoRedoHandler deletes a Page/Alarm or VKB the appropriate TopComponent-Window will be closed as well.
If the UndoRedoHandler restores a Page/Alarm or VKB the TopComponent-Window won't be opened again.
- Other objects (Frames, String Fields, Buttons, Softkeys,...)
If the UndoRedoHandler adds/removes or updates one of these objects, the appropriate page will be opened and the object will be selected on the scene.
=> If more than one object is affected all objects will be selected
- Moving/Resizing Objects on the scene

If one or more objects are moved or resized on the scene, only one undo edit will be created => The undo edit will be created when mouse release event occured.
=> The edit is directly created when the first update occurs, but all following updates/undo edits will be merged into the first undo edit.

Communication-Object Undo-Edits
If Undo/Redo for Communication-Objects is performed, the appropriate change will be undone/redone but the engaged objects won't be selected in project tree or variable view.

Automatic Merge
The UndoRedoHandler has an automatic merge mechanism of undo entries. If several undo entries are triggered within 500ms, these entries will be merged to one entry.
This mechanism is used e.g. if the the property Dependency of a String Field is updated from None to Language. All Language dependent properties(Preview Value, Font, ...) are changed now (from Single Value to a Value-Map) and these changes will be merged. If that wasn't done, the user would be able undo each single Property-Change-Edit, which would lead to an inconsistency.
The same mechanism is used if e.g. the data type of a Variable is changed. Consequently, all data dype dependent properties are updated (Min-,Max- and Default-Value) and will be merged to one undo entry.
Another example for the automatic merge is creating a VKB. During the creation a lot of objects will be created. With the automatic merge the whole creation process will be merged to one undo entry.

3.18   Variable View

The variable view is meant for an overview of all variables in a project, quickly adding a variable to the variable list and assigning a variable via drag and drop to a user interface object.
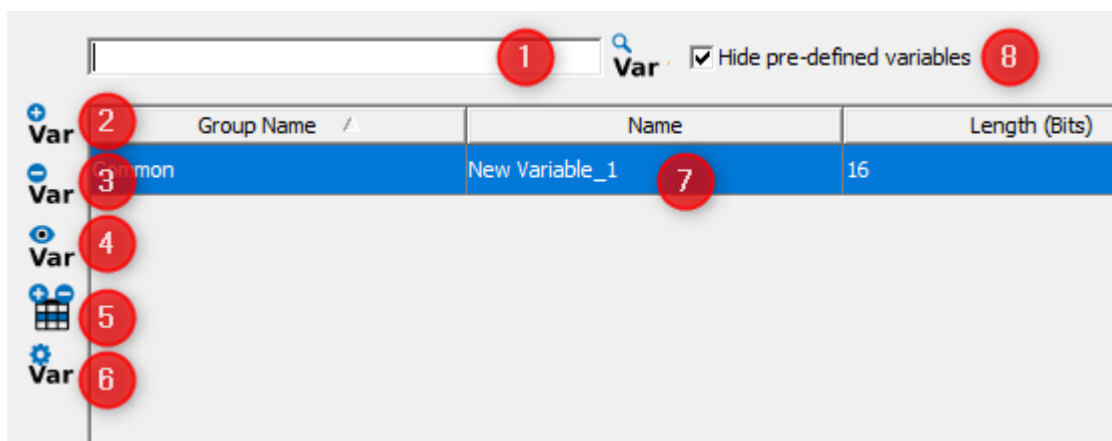
1	Full live search field. You can search for "Group Name", "Name", "Index" and "SubIndex". Results will be filtered live as you type
2	Add a variable quickly. If you press this button a variable is added with a default name with increasing name index and the default owner is "PClient"
3	Delete a variable. First you have to choose a variable or more variables and than delete them. Pre-defined (system) variables[174] can not be deleted.
4	Find usage[148] . Select a variable and press this button. The find usage window opens and shows the connected GUI elements and CAN message mappings.

1	Full live search field. You can search for "Group Name", "Name", "Index" and "SubIndex". Results will be filtered live as you type
5	Additional columns. Here you can choose to show more than the default columns.
6	Variable View Config.
7	Main list. If you click on the header of the columns you set the order of the entries. Please note that you can only sort by one column.
8	Hide / Show pre-defined variables.

If you have selected a variable within the list and click right you will get the following context menu.
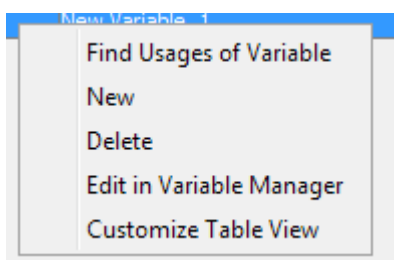
New	The same like you press the add variable icon
Delete	The same like the delete icon
Edit in Variable Manager	Here is the quick jump to the variable manager where you can do more detailed settings for the variable
Find Usages of Variable	Select a variable and look for the connected GUI Elemets and CAN Mappings.
Customize Table View	The same as you where clicking the additional columns icon.

Variable View Config
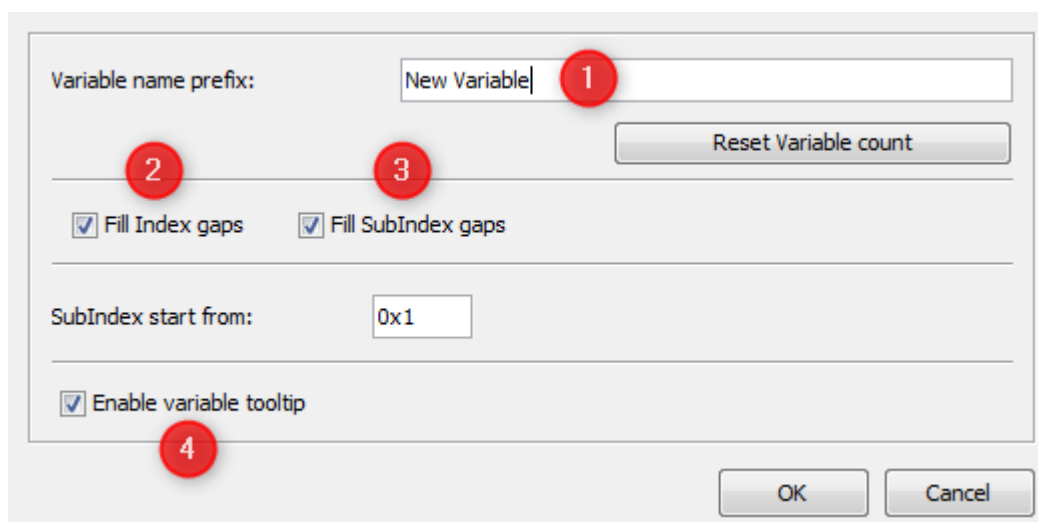
The variable view configuration is mainly for setting up the quick adding of variables. You can define a prefix (1) which is filled up with a number. This is causing names like "New Variable1", "New Variable2" and so on. The number count can be reset by pressing the button "Reset Variable Count". This reset has nothing to do with the variable id or other

indexes. Within the next section you can define how to deal with the index. Fill Index gaps (2) enables the creation of variables with free indices between already existing indices. Fill SubIndex gaps (3) enables the creation of variables with free sub indices between already existing sub indices. for a given index. SubIndex start from: sets the starting sub index when a variable with a new index is created. Please note that the index and subindex is needed by the program and the runtime on the display, but with these default settings you do not really need to take care about this. (4) With this check box it is possible to enable/disable the tooltip of the variables in the Variable View.

Drag and drop variables

All variables can be selected and dragged and dropped onto the designer area and appropriate objects in the project tree. Drag and drop of multiple variables is not supported. Read more about Drag and Drop in the chapter Drag and Drop[43].

3.18.1   Find Variable Usages

Common

The Find Variable Usages will allow the user to see all the usages of a variable at a single place. A variable can be used in the protocols, mappings and DDOs.
When starting the program you should already find a find usage window beside the project tree window (1).
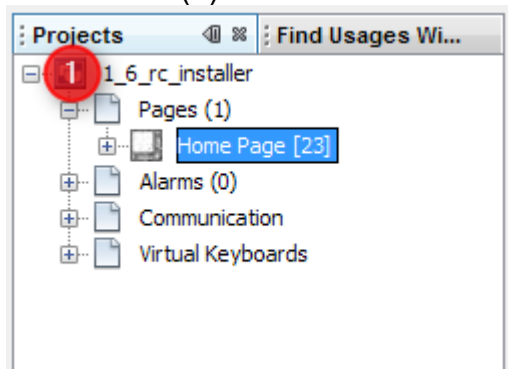
If you can not see this window you can enable this window through the window menu. With this menu you can hide the find usage window too.
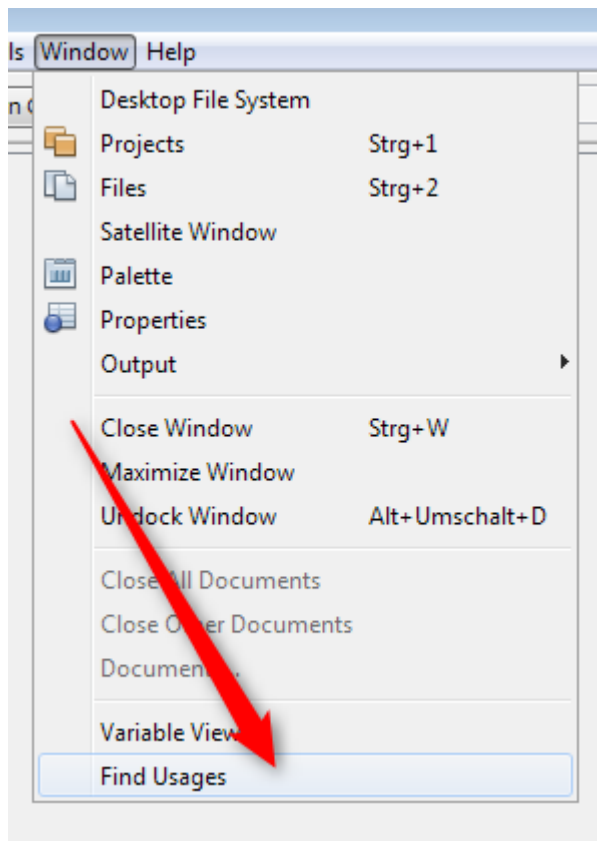
How to use the find usage?

Find usage is started through the variable view on the bottom of the program. First select a variable (1) and than click the find usage icon (2) or, alternatively, right-click on the selected variable and choose "Find Usages of Variable" in the context menu.
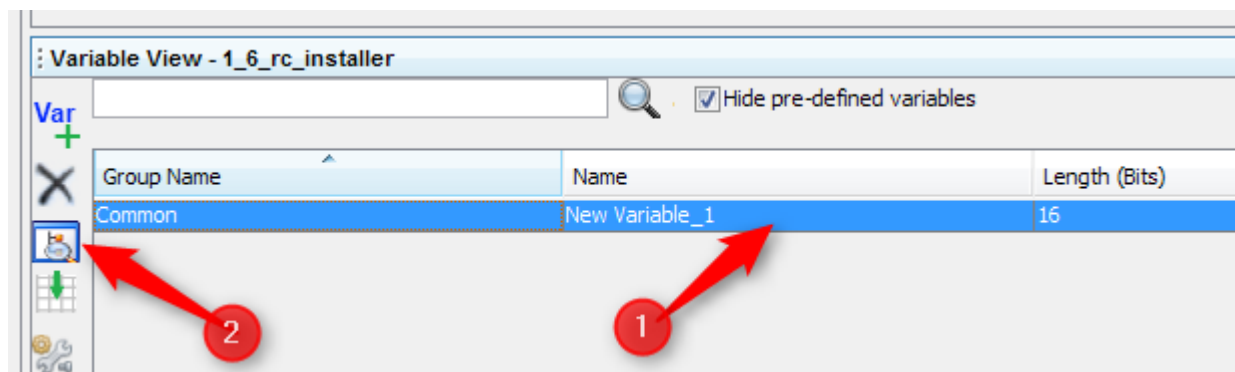
Now the view on the left top side changes. Beginning with the root node of the project (1) you can see the complete tree down to the leafs (3,4).
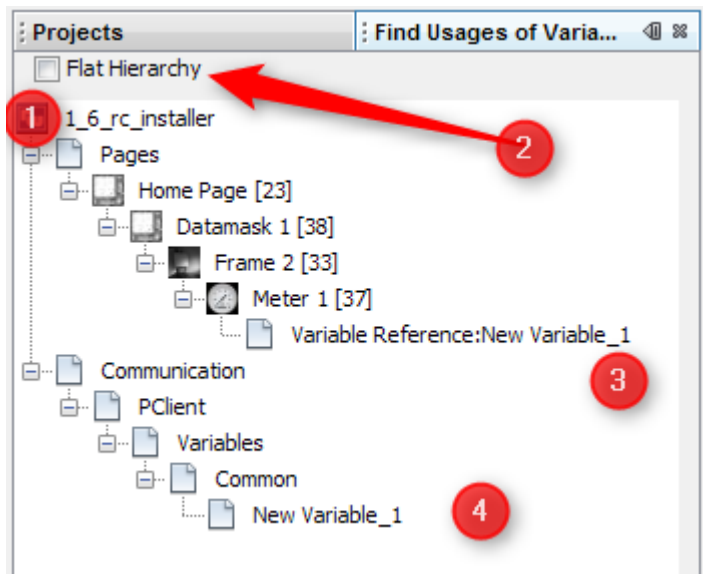
If the hierachy of the project is too deep you have the option to flatten the view (2). This results in:
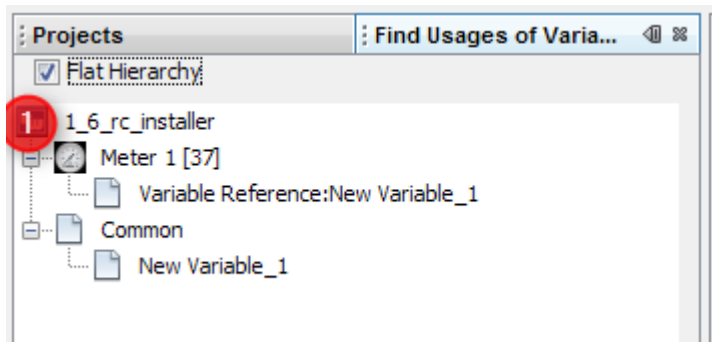
How to navigate with find usages?
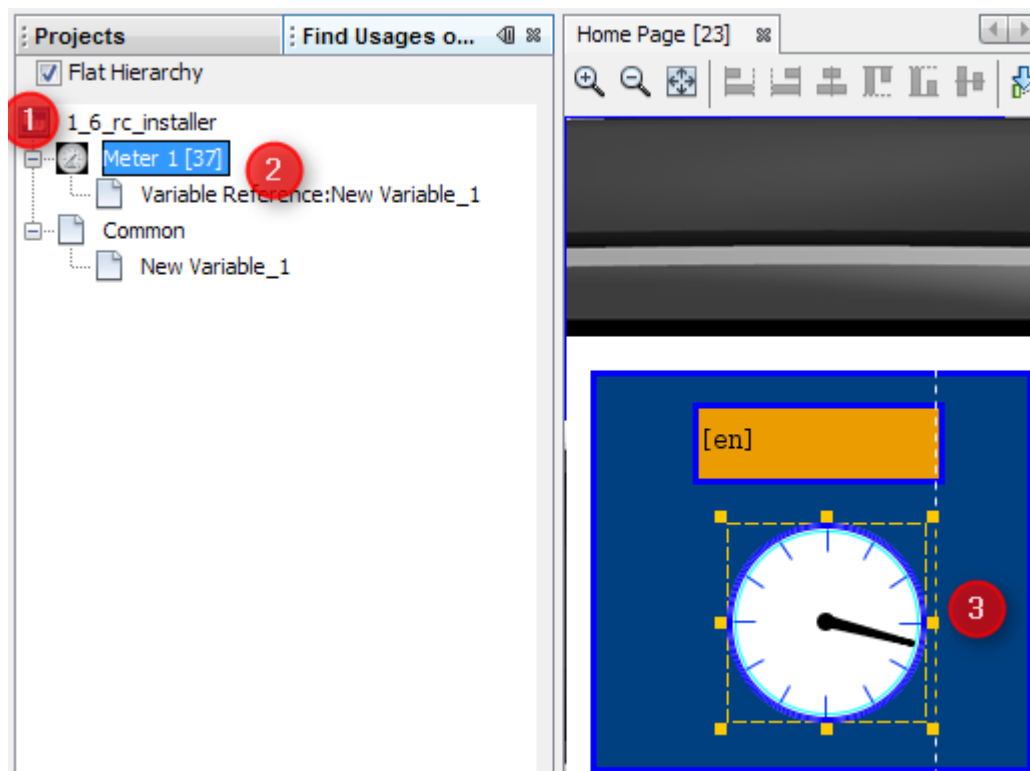
To jump directly to the found page double click directly on the object (2) in the tree (1). This results in opening the page and selecting the correct object (3).
Known issue: In this version, the double click for the objects that are connected with the variables is disabled. Please use the project tree to browse to the object manually.

3.19   Z-Order

Objects have a certain order in each page.
The order is represented by the order in the project tree:
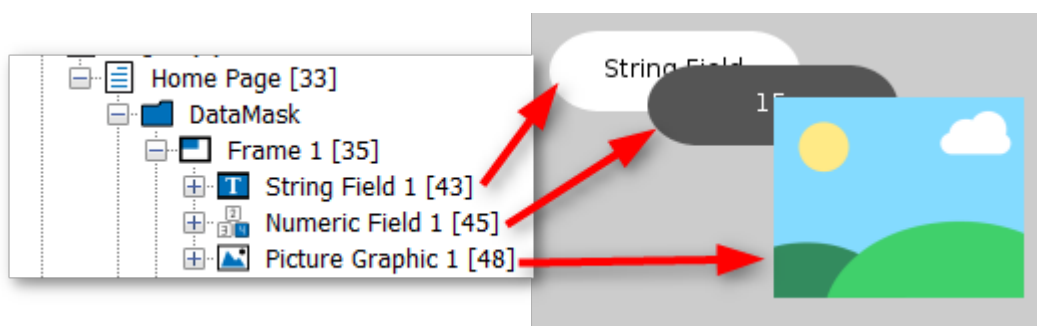
The bottom of a level in the project tree represents the foreground, the top represents the background.
Newly created objects are always put at the bottom of the project tree / in the foreground of the scene.

The Z-Order can be changed in several ways:

1.

By right-clicking the object in the project tree or in the scene and choosing the category Z-Order.

Then the object can be set
- one step to the front (Level Up)
- one step to the back (Level Down)
- into the background (Bring To Back)
- into the foreground (Bring To Front)

2.

By using the Level Down [image] and Level Up [image] buttons in the page editor.

3.
By dragging and dropping an object in the project tree.
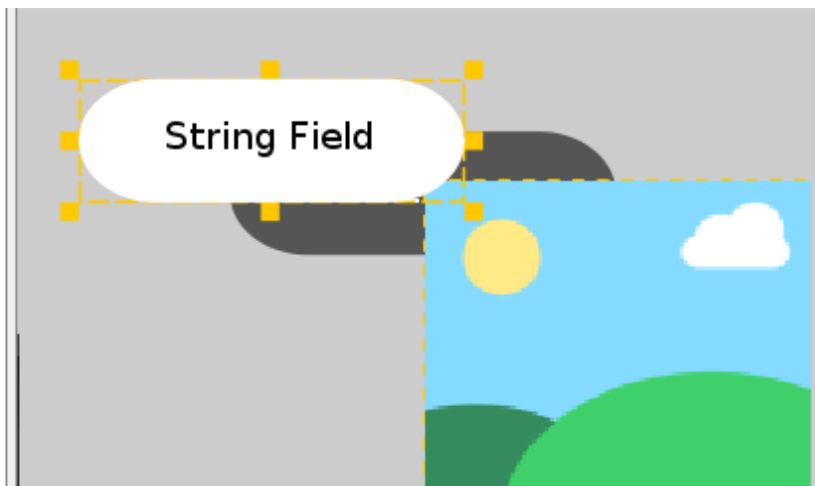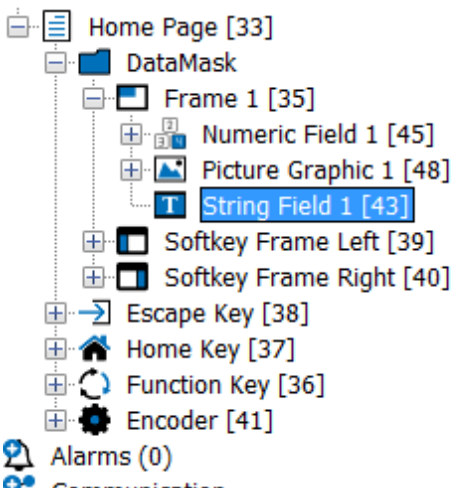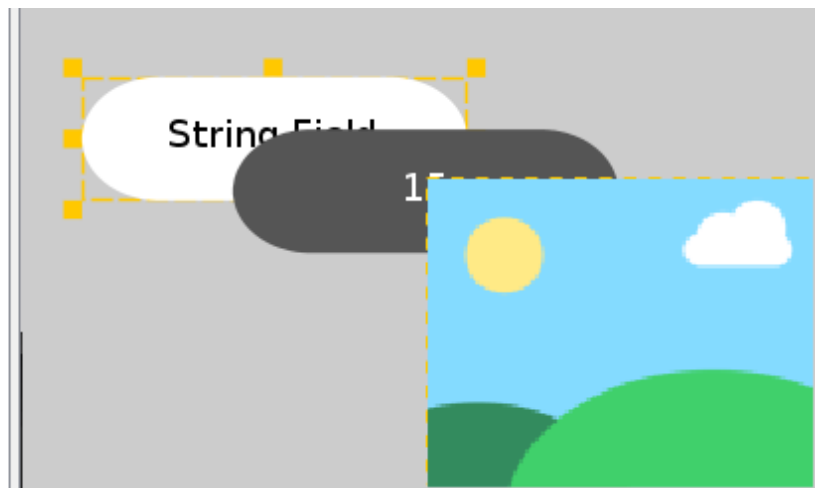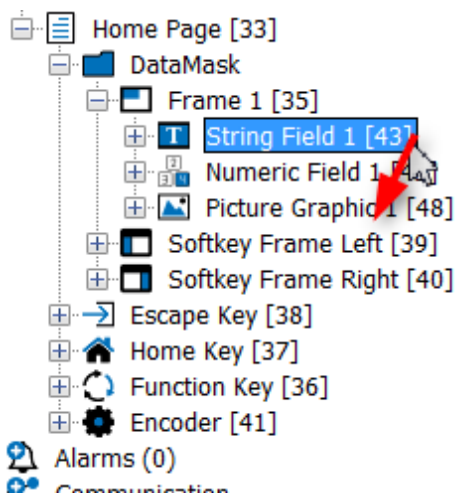
The Z-Order in a page also influences the order that input objects are selected with the encoder.

There is a property called "Roll Over" that can be accessed with JavaScript. If set to true for input objects like Numeric Fields, turning the encoder beyond the absolute value limit will roll the value over to the absolute minimum value. For frames, it will switch from the "last" object on the page back to the first one.

Variables

4   Variables

The main approach followed in the program is to create pages and the navigation between them which is a more static definition.

The dynamics come in with variables changing their values. Other tools use the term "signal" instead of "variable" to describe the same thing. A CANopen "object" also describes what we call a "variable" here. The same is true for J1939 "Suspect Parameter Numbers (SPNs)".

This page provides an overview of the general concept behind the variables.

There are follow-up pages explaining the menus and dialogs to create and manage variables owners[160] and variables[165] .

Owner Concept

Every variable in the GUI designer has to have an owner assigned. The owner is the only one controlling the value of the variable.

If somebody else tries to write the value of the variable this is considered a write request which is forwarded to the owner of the variable.

Only if the owner confirms the new value by providing an update of the variable it is considered changed.

If the variable is expected to be frequently updated by the owner e.g. via CAN bus message then they can also be considered invalid if the update does not arrive in time.

Types of Owners

Generally there are two different types of owners:
- Remote ECUs connected via CAN bus
- Other programs/processes running on the display.

Examples of the second type are "Hardware Daemon" providing all of the hardware related predefined variables and "UD4" used for updates of the GUI definition on the display.

If considering to add your own process to communicate with PClient please contact our support team and expect to include some C++ libraries for the communication between the two in your program.

PClient as Owner

If PClient itself is the owner of a variable generally the same rules apply as for other owners. PClient has no built-in knowledge about which values of a variable are acceptable and which are not - except for the absolute minimum and maximum values. Therefore PClient basically accepts and confirms all write requests to its variables. Since PClient is active as long as the GUI is shown, PClient variables are valid if they ever got a valid value.

Which Owner should be assigned to a variable?

Generally the idea is to assign the variable to the owner knowing the system's limits and measuring its state. This means that e.g. engine coolant temperature should be owned by engine ECU.

On the other hand there may be some setpoints and GUI internal variables which should be owned by PClient.

A decision has to be made if a variable has to be considered a measured value as well as a setpoint.

Assume engine speed is a measured value coming from engine ECU but at the same time the operator can set a new desired engine speed by applying a new value to the engine speed variable.

Then there are (at least) two different sources for new values of the same variable.

If engine ECU is the owner of engine speed in this example, then the following will happen:
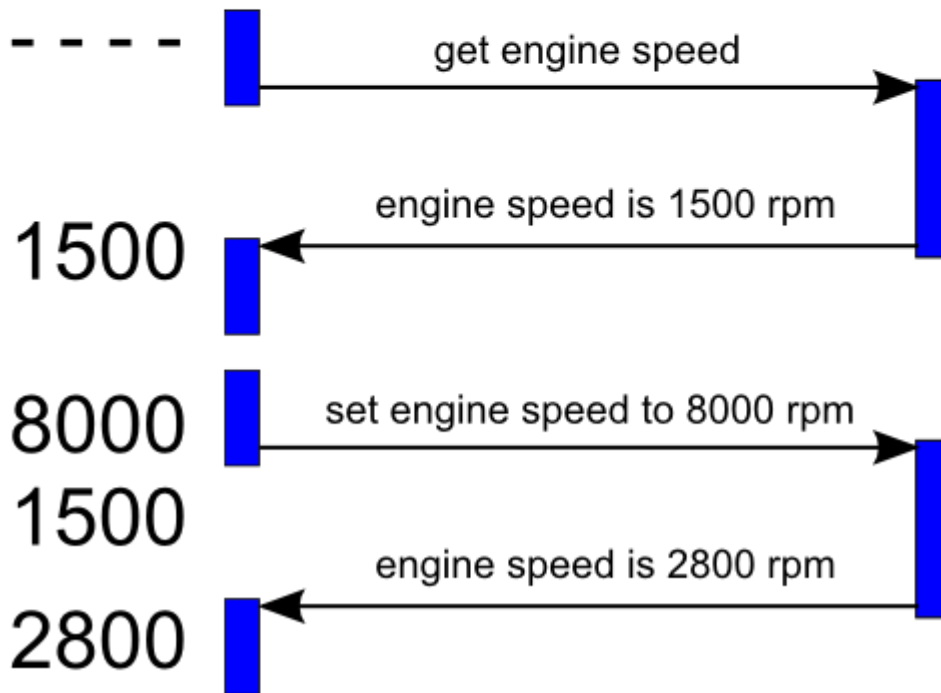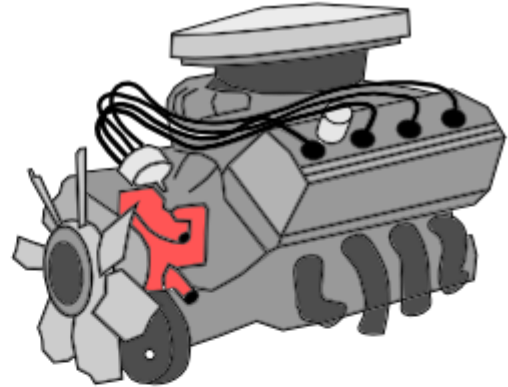
One can see three effects here:
1. Typically the value is invalid until the owner provides the first value - there is an option though to avoid that.
2. After providing the new setpoint the value shown will switch back to the last value provided by the owner.
3. Since the owner is in control it may limit the value provided in the write request to something reasonable in current conditions.

If the display should e.g. forward the engine speed to another bus than the one where engine ECU is connected then it will only forward values confirmed by the owner.

If flickering back to last known value is not acceptable and the GUI can make sure that all settings made are valid then PClient could become the owner which would mean that any value of the variable is valid and variable will stay valid at any point of time.
If display should forward the value owned by PClient to different busses it will forward each and every value modification made by anyone - because it considers all modifications valid.

Remanent Variables

Remanent variables keep their values over power cycles. They get re-initialized whenever a new PClient is installed on the display.
Remanent variables are intended for operator settings and other values which will change but whose last value should be remembered when PClient is started again.

Predefined Variables

On the display there is a variety of predefined variables[174] you can directly use in your GUI. They provide e.g. hardware states such as I/O values for digital and analog ports or CAN bus status as well as display specific properties e.g. the part number and serial number. Predefined variables are also used to e.g. configure camera interfaces, display backlight or beeper activity and obtain and set current time as provided by real time clock (RTC). Other predefined variables provide information specific to your GUI, e.g. dealing with virtual keyboards and alarms and visibility. There are also variables showing the recent pages visited and the time a page was active.

ISO-VT Variables

There is a range of indices reserved for ISO-VT. These variables are not existing by default but you can create them as they are defined in the variable table.
This approach allows to save the resources if you don't use ISO-VT on your display while still providing the information in a standardized manner not interfering with your user defined variables.

Data Types

Every variable has to have a type. By default character strings and numeric types are supported.
Numeric types are either signed or unsigned and have a length in bits.
In order to support the minimum requirements of CANopen there are a few predefined data types which can not be deleted or modified.
Additionally both two bit types are provided which are used for some predefined variables representing booleans with an additional "undefined / unknown / don't care" state required e.g. for J1939 protocol.
Note: String variables are intended to carry readable strings. Therefore their value is considered to end where a "\0" character occurs.
Note: If string values should be shown on display they are interpreted as UTF-8 format which is the same as ASCII for the first 127 characters only.

Organizing principles

The program provides two separate means of variable organization:

- Variable groups are supposed to support the designer in keeping variables where he will find them again. These groups are not relevant for the device.
- Index and subindex are used to identify a variable even on display side. This concept was mainly introduced for CANopen support but is used for all variables.

Variable Groups

It is suggested to create variable groups for all variables belonging together in one way or the other. One option would be to create one variable group per ECU. Another would be to find variables belonging to a certain subject and create a group for them. Anyways each variable has to belong to exactly one variable group and if you don't create your own groups all your variables will end up in "common" group.
Note: In main screen variables can be handled and searched using variable view[146] but when choosing a variable in a dialog it can really be helpful to have them ordered in meaningful groups.

Index and Subindex

CANopen organizes all variables as "objects" in its "object dictionary". From this object dictionary definition we "borrowed" the index and subindex approach. And since the display does not follow any of the predefined CANopen "device profiles" we had to move them all into the CANopen index range of 0x2000 ... 0x5FFF where proprietary objects have to reside.
Within this range we reserved index 0x2000 ... 0x2FFF for our predefined variables whereas user defined variables are located in the 0x3000 ... 0x5FFF range.
CANopen (and our program) allow two approaches to handle subindices: Either the subindex 0 is the only variable defined for an index or subindex 0 defines the highest subindex existing for an index.
It is recommended to use the latter approach and create subindices starting from 1. If you really want to use CANopen, the program will calculate the correct value of subindex 0 in this case.
Similar to variable groups you can use index as a grouping mechanism for many sub-indices. It would probably be a good idea to store all variables with the same index in the same variable group.

Organization of predefined variables

For the predefined variables every variable group contains exactly one index with all its sub-indices.

Variable import

It is possible to import variables from another project or via a .csv file with parameters for variables. This way large amounts of variables can be created quickly. The .csv file has to look like this:

```
Variable;Index;Sub-
Index;Owner;Group;Property_Description;Data-
Type;Length;Access Type;Min Val;Max Value;Default
Value;Initialize Value As Invalid;Remanant;Force writing
test1;0x3011;0x01;;;;;INTEGER;8;;;;;TRUE;FALSE;FALSE
test2;0x3011;0x02;;;;;INTEGER;16;;;;;TRUE;FALSE;FALSE
test3;0x3011;0x03;;;;;INTEGER;8;;;;;TRUE;FALSE;FALSE
```

```
test4;0x3011;0x04;;;;INTEGER;16;;;;;TRUE;FALSE;FALSE
```

To import variables, click in the menu Communication -> Import Export Variables -> Import Variables
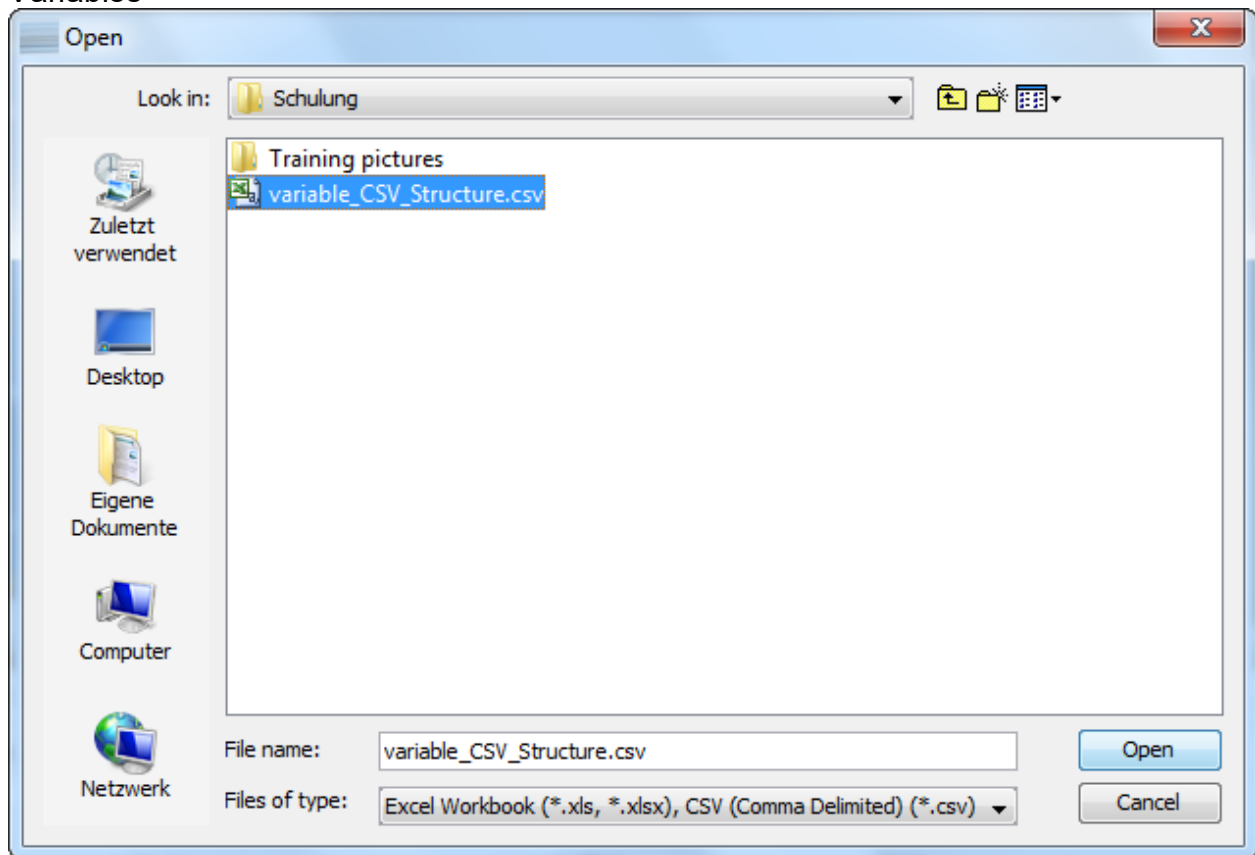
In the file dialog, choose the according file type (project xml or CSV) and the file from which you want to import the variables and click Open.
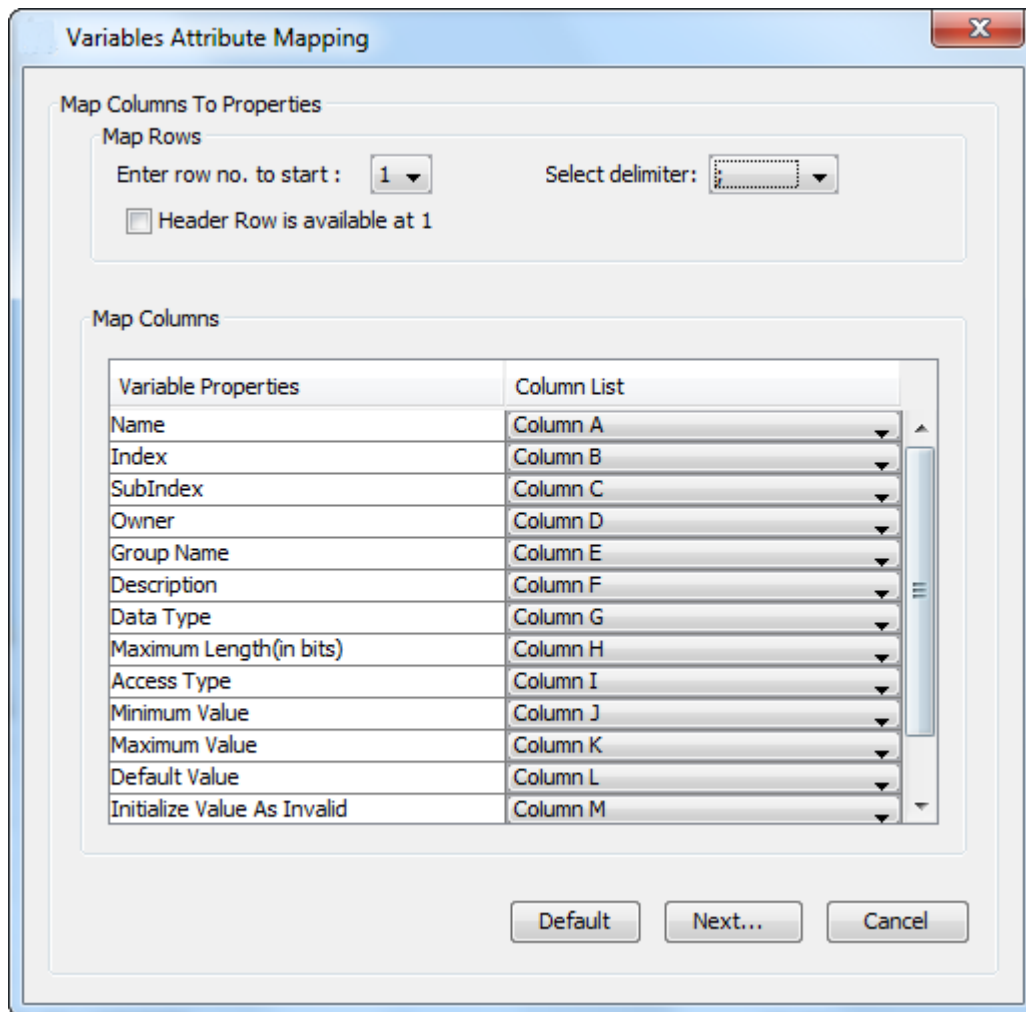
In this dialog you can choose the appropriate delimiter, the row where the program should start reading the variable data, and a table where you can tell the program which column contains which property for the variables. If you have used the format shown above you can use the default values and click Next...

If you haven't given all properties for all variables, the program will show dialoges and suggest default values for these properties. After confirming these a last dialog will be shown where you can choose which variables you actually want to import.
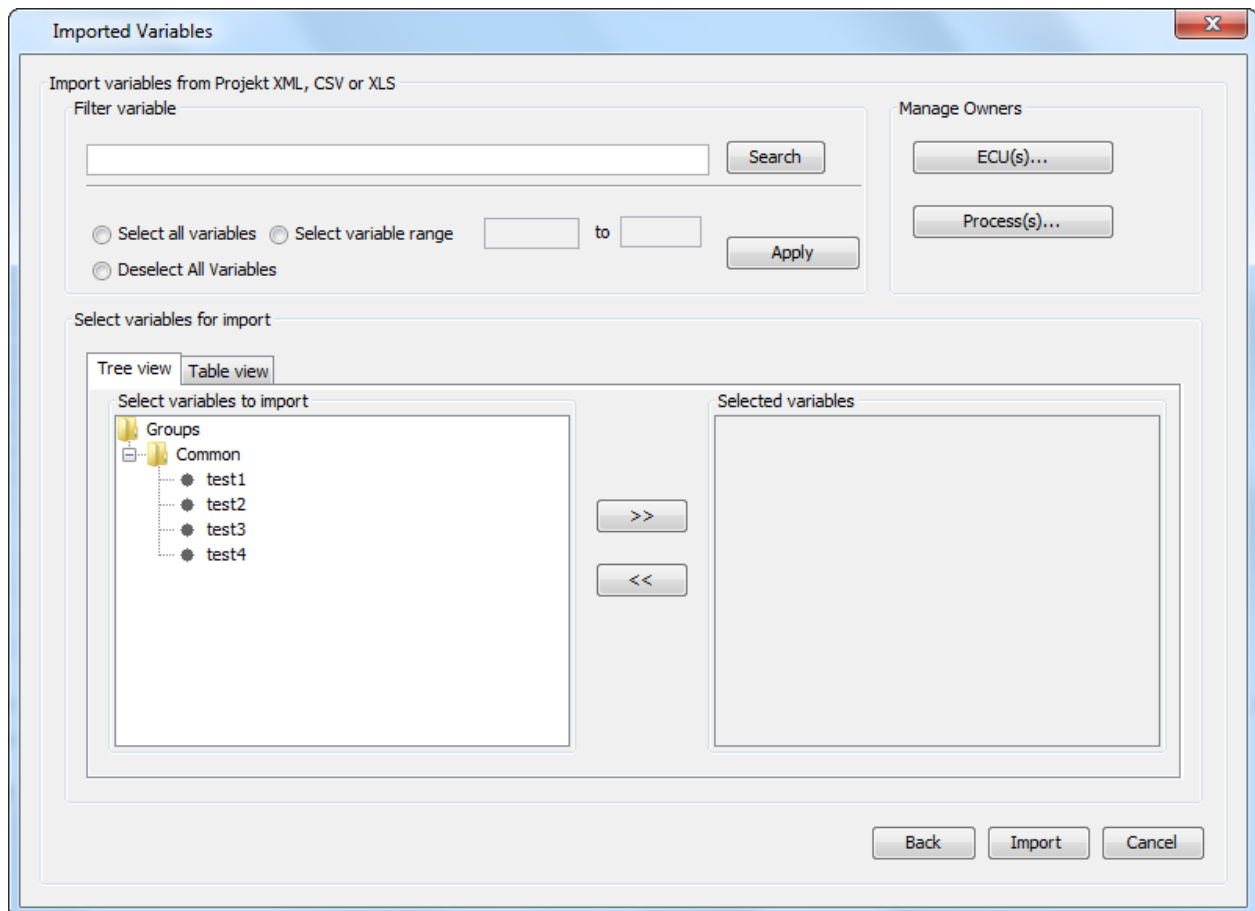
You can select single variables or all variables. Click on the button >> to choose variables for import. Click import to create all the chosen variables in your project.

4.1 Variable Owner Menus

On this help page, the user interfaces to create and manage variable owners are handled.
It is based on the general concept of variables and owners[154] but concentrates on the specific properties one can define for variable owners.
There is one follow-up page explaining the creation and maintenance of variable properties[165].

Creating a Variable Owner on CAN Bus

There are two options to open the variable owner dialog:

Open Terminal and Owner ECU(s) Configuration from Menu
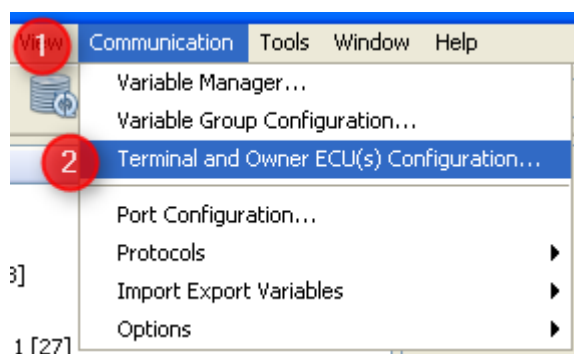
1. Open "Communication" menu.
2. Select "Terminal and Owner ECU(s) Configuration ...".

Open Terminal and Owner ECU(s) Configuration from Variable Manager
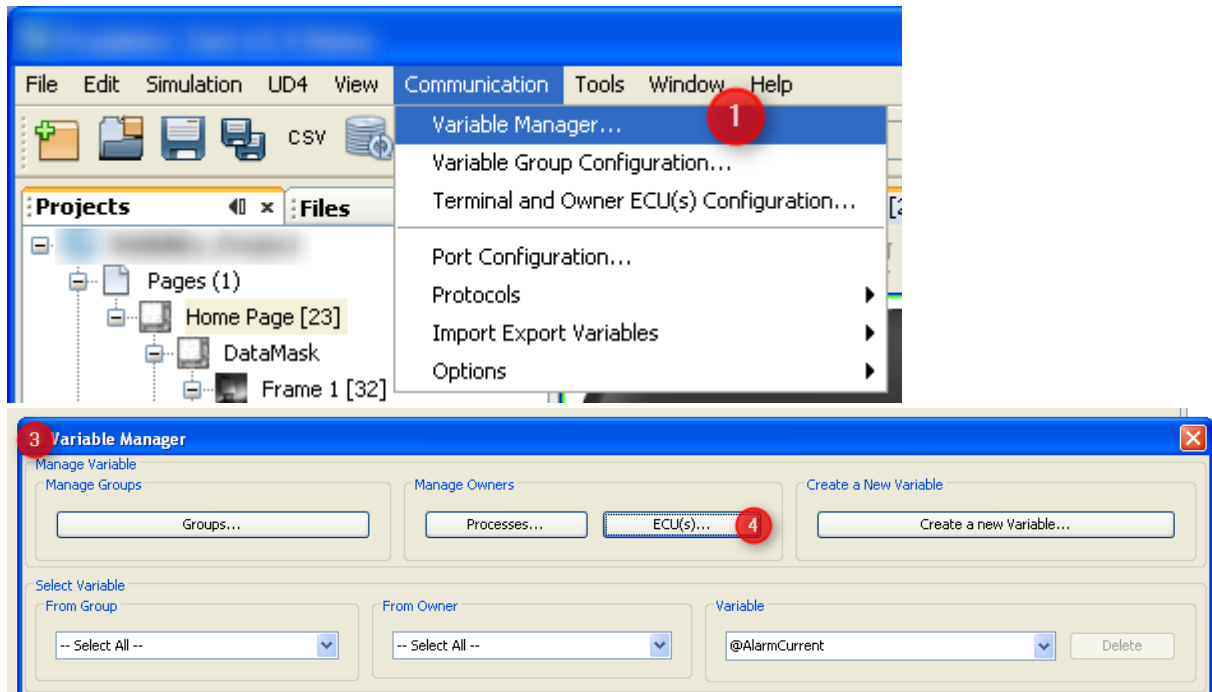
1. Open "Communication" menu.
2. Select "Variable Manager ..."
3. The Variable Manager dialog opens which could stay open for all variable creation and management.
4. Press button "ECU(s)...".

Terminal and Owner ECU(s) Configuration
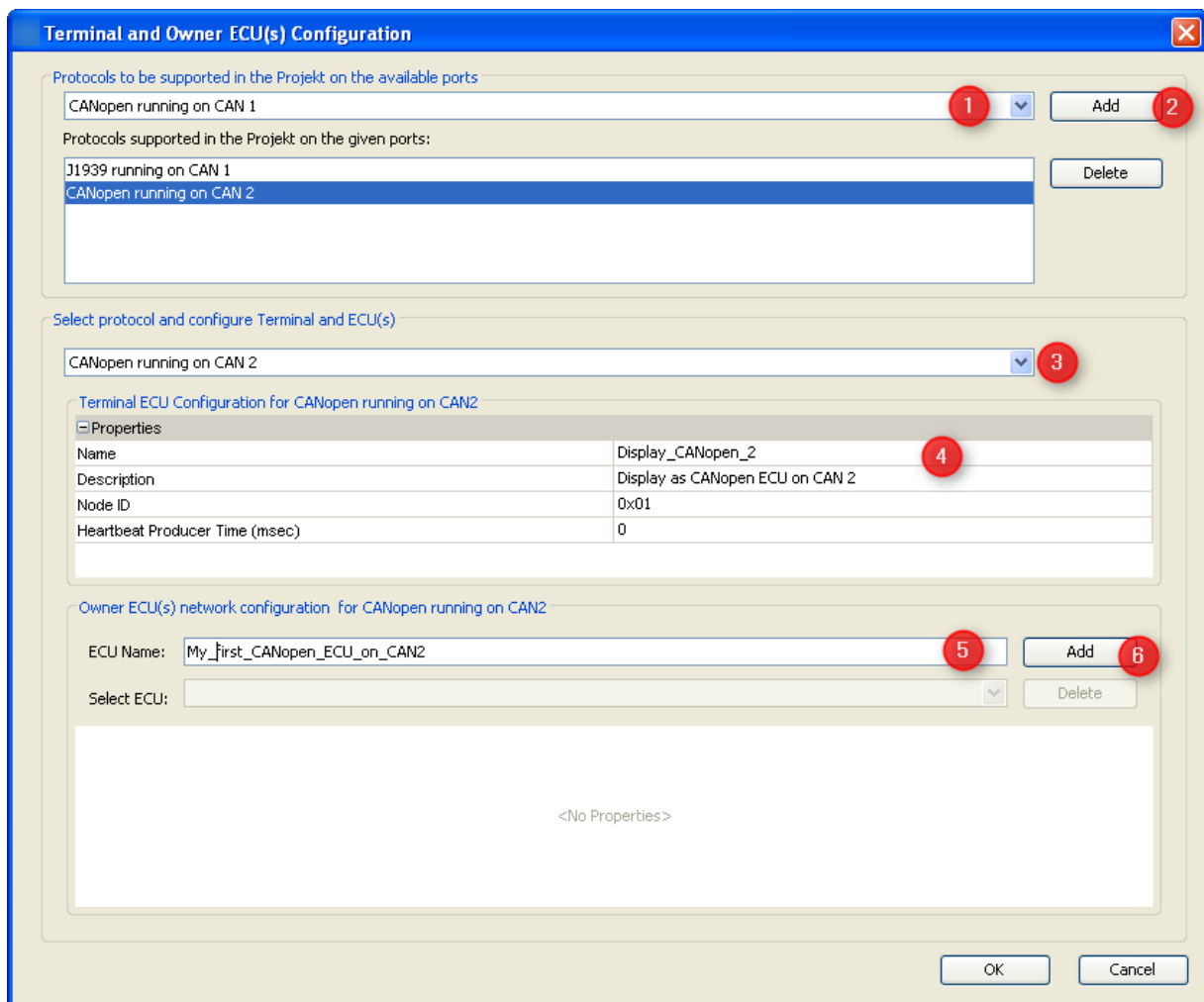
1. Select additional protocol and port it should run on.
2. Press "Add".
3. The new protocol will be selected automatically. In order to modify settings of an existing protocol select it here.
4. Edit display's properties for the protocol - these typically define how the display can be identified on the bus.
5. Type name of an ECU
6. Press "Add" in order to add new ECU to the protocol selected.
7. Edit properties of new ECU

Note: This only describes the general approach of adding CAN ECUs as variable owners. Detailed descriptions about their properties can be found in the pages of the respective protocols: CAN Freestyle[308] , CANopen[328] and J1939[345] .

Processes as Variable Owners

The description above describes remote variable owners providing their variables via CAN bus.
Another source of variables are other programs running on the display. These are configured in "Process Configuration" dialog.
Again there are two ways to open this dialog.

Open Process Configuration from Menu
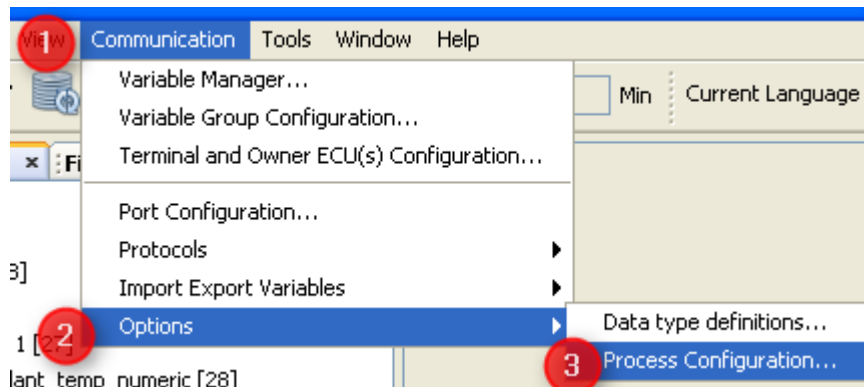
1. Open "Communication" menu.
2. Select "Options".
3. Choose "Process Configuration...".

Open Process Configuration from Variable Manager
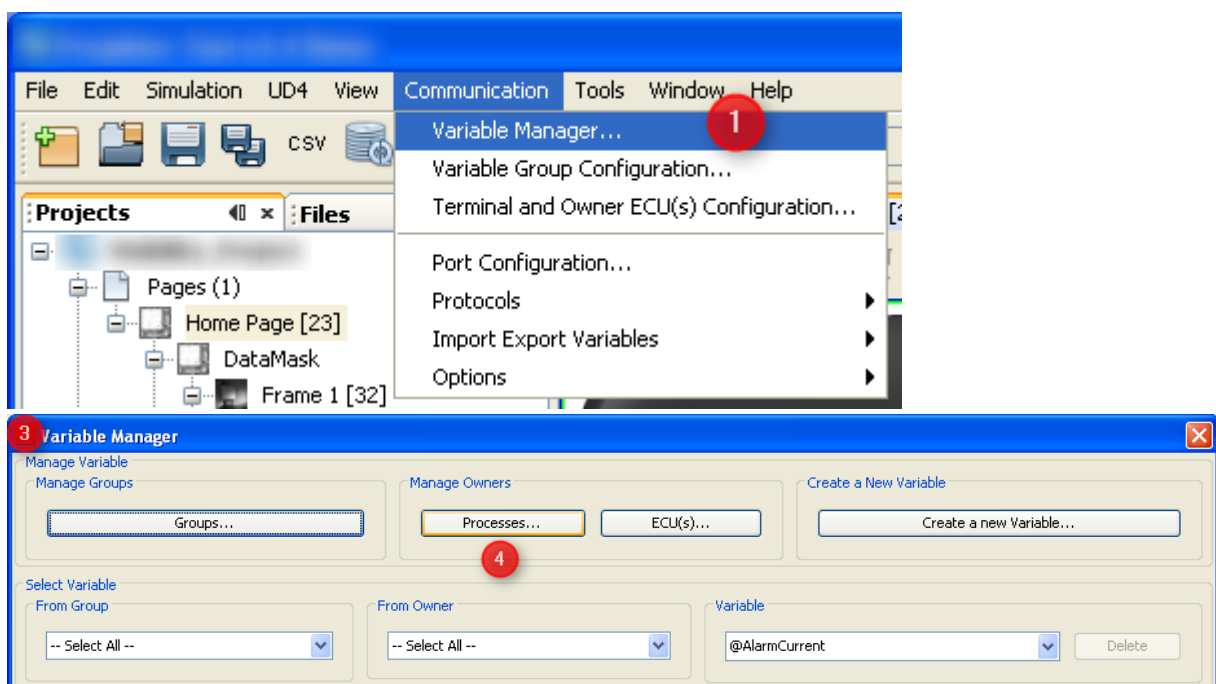
1. Open"Communication" menu.
2. Select "Variable Manager...".
3. Variable Manager dialog opens which provides all the links to create and maintain variables.
4. Press button "Processes...".

Processes Configuration
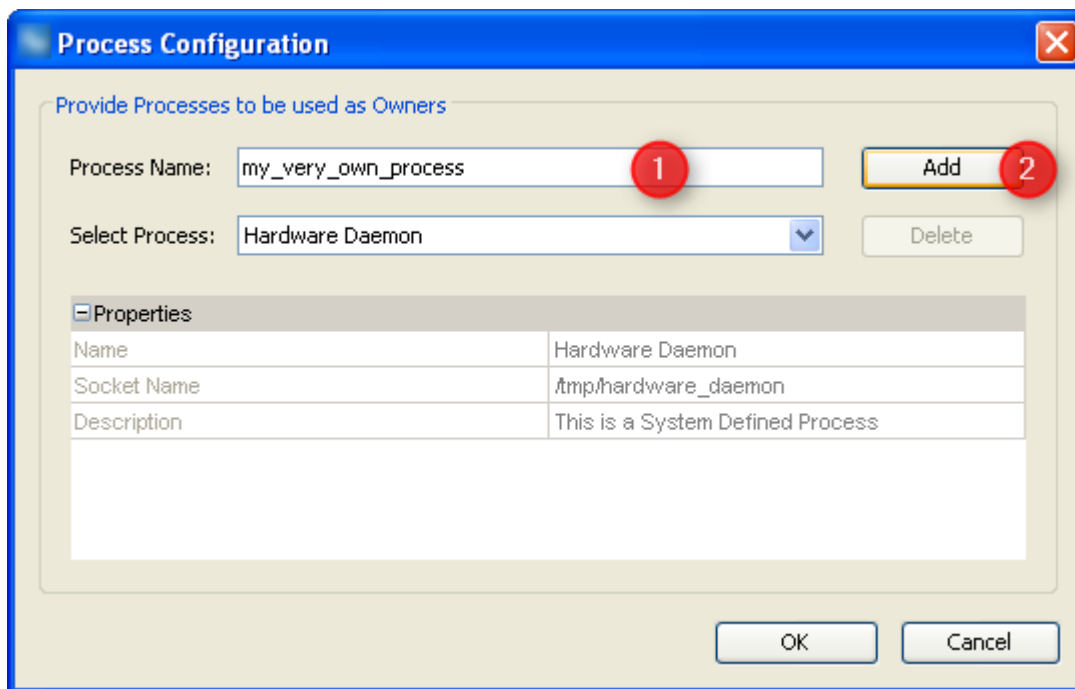
1. To add another process type the name of the program here.
2. Press "Add" button to add it to the GUI definition.

Once you have added the process or selected an existing one you can edit their properties.

Note: System defined processes can only be viewed but not edited or deleted.
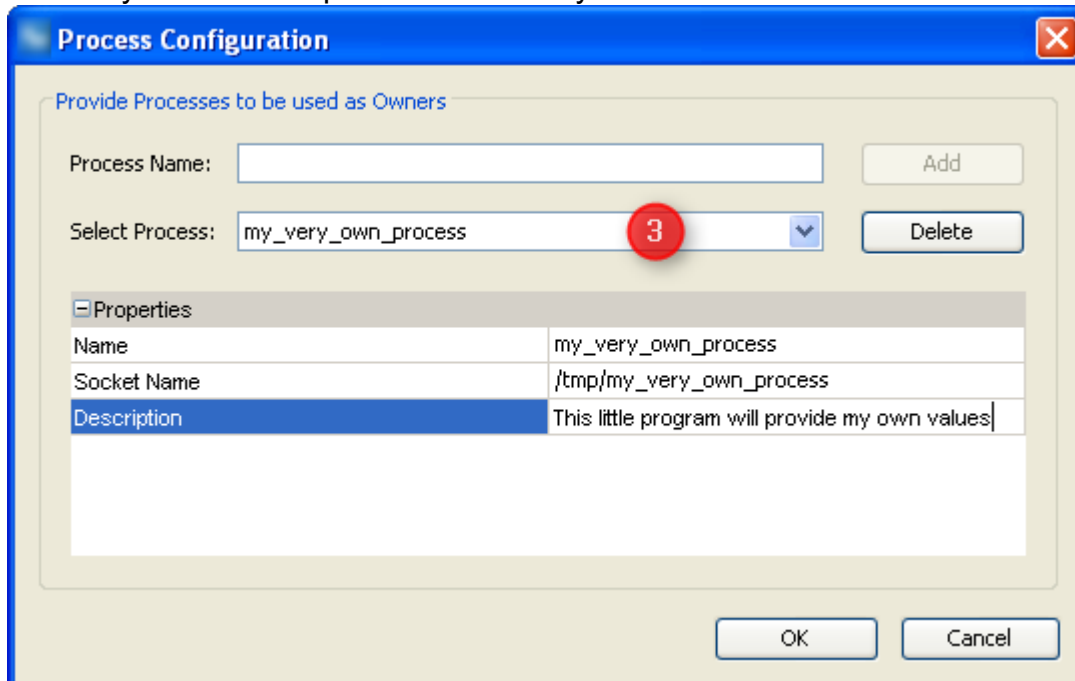
3. If you didn't just add the process you have to select it here.

Name

The name of the program to communicate to. This should be the name of the executable as you would type it on command line in order to run the program.

Socket Name

This describes the interface PClient will use to communicate with the process. By default this is the same as the program's name but located in folder "tmp".
If you should not know what a socket is: Consider it a kind of file which is used to exchange variable values.

Description

As with all the description properties this mainly exists for your convenience (especially when re-opening your GUI definition after a longer period of time).
So take the time to add a sensible description of what this process should do and why it was created. Then it could be helpful for others looking into your GUI definition, e.g. our support team.

4.2   Variable Menus

This is a page describing variable creation and maintenance.
It is based on the concepts described in general variable approach[154] and also uses settings explained in variable owner configuration[160] .

Definition of Data Types

Data types can be created and maintained in the "Data type definitions" dialog.

Open Data Types Definitions Dialog
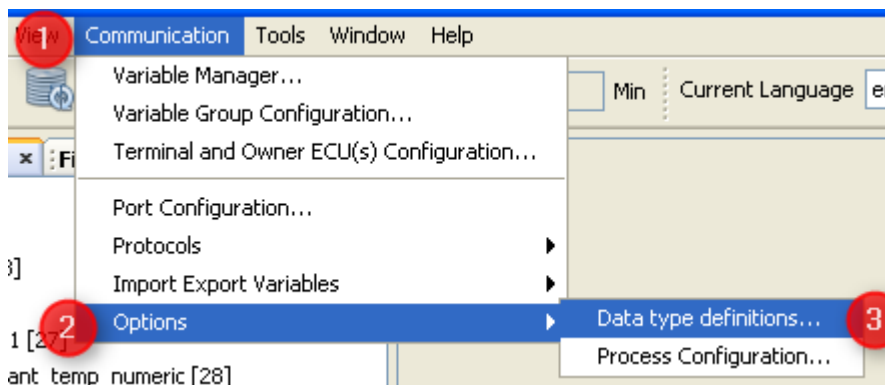
1. Open menu "Communication".
2. Select "Options"
3. Choose

Use Data Type Definitions Dialog
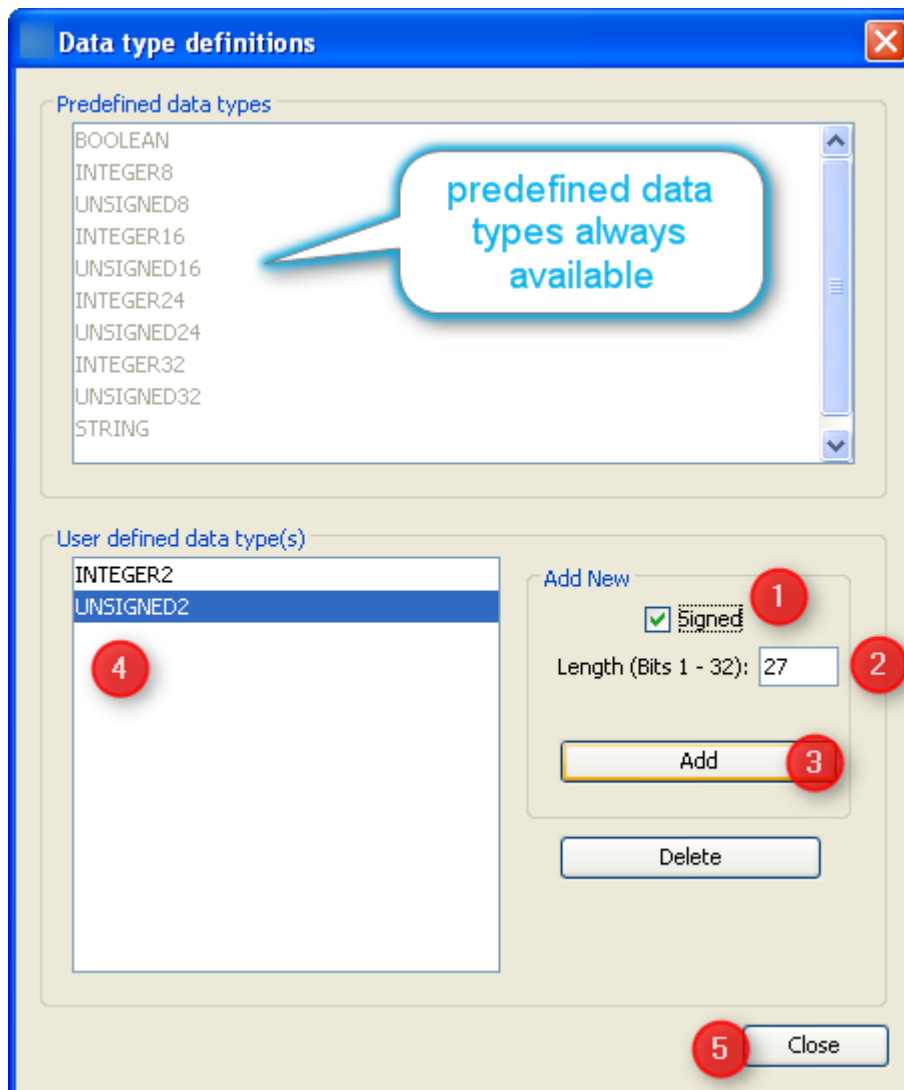
1. Check "Signed" if you need an additional signed data type.
2. Type the length of the new data type in bits.
3. Press "Add" button to add your new data type to the GUI definition.
4. New type will appear in this list (for our example it would be "INTEGER27").
5. Press "Close" button once you're finished adding and deleting variable data types.

Create and Maintain Variable Groups

As stated in general description[154] it is good practice to organize variables in groups to support maintenance once there are more than a few of them.
There are two ways to open the "Variable Groups Configuration" dialog.

Open Variable Groups Configuration from Menu
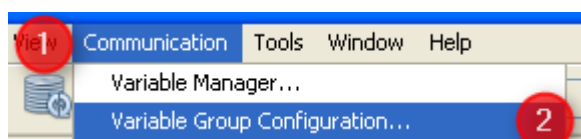
1. Open "Communication" menu.
2. Select "Variable Group Configuration..."

Open Variable Groups Configuration from Variable Manager
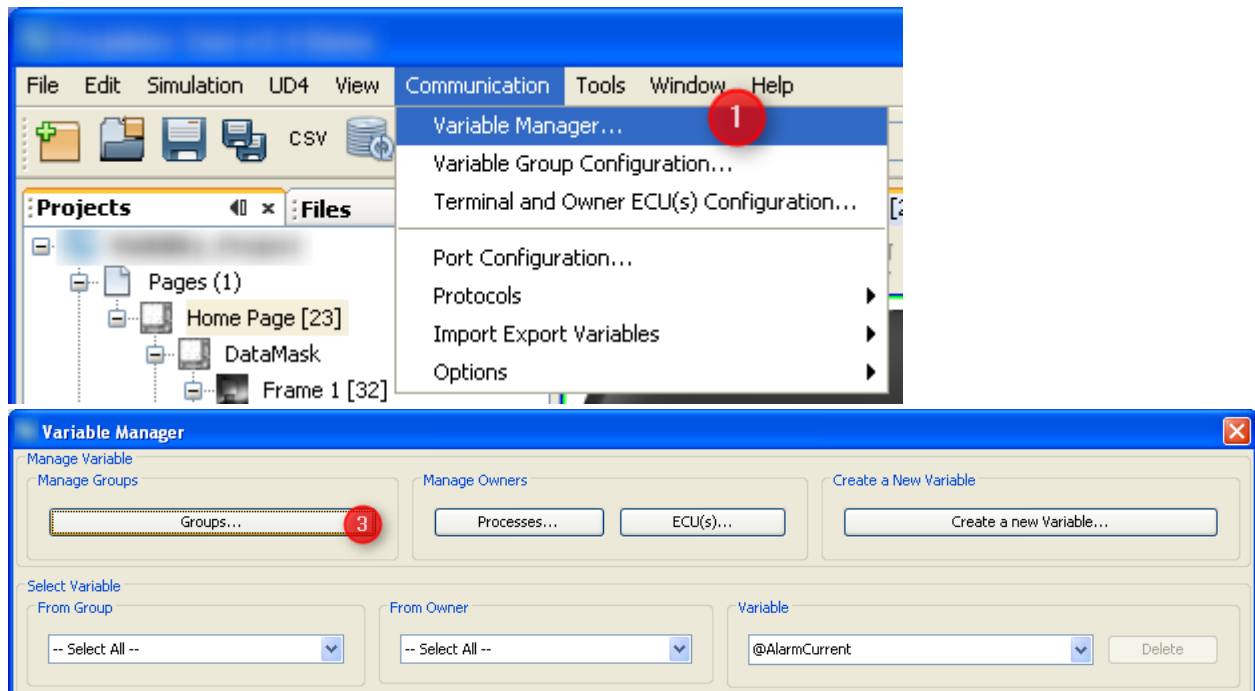
1. Open Menu "Communication".
2. Select "Variable Manager..."
3. Press button "Groups..."

Variable Group Configuration
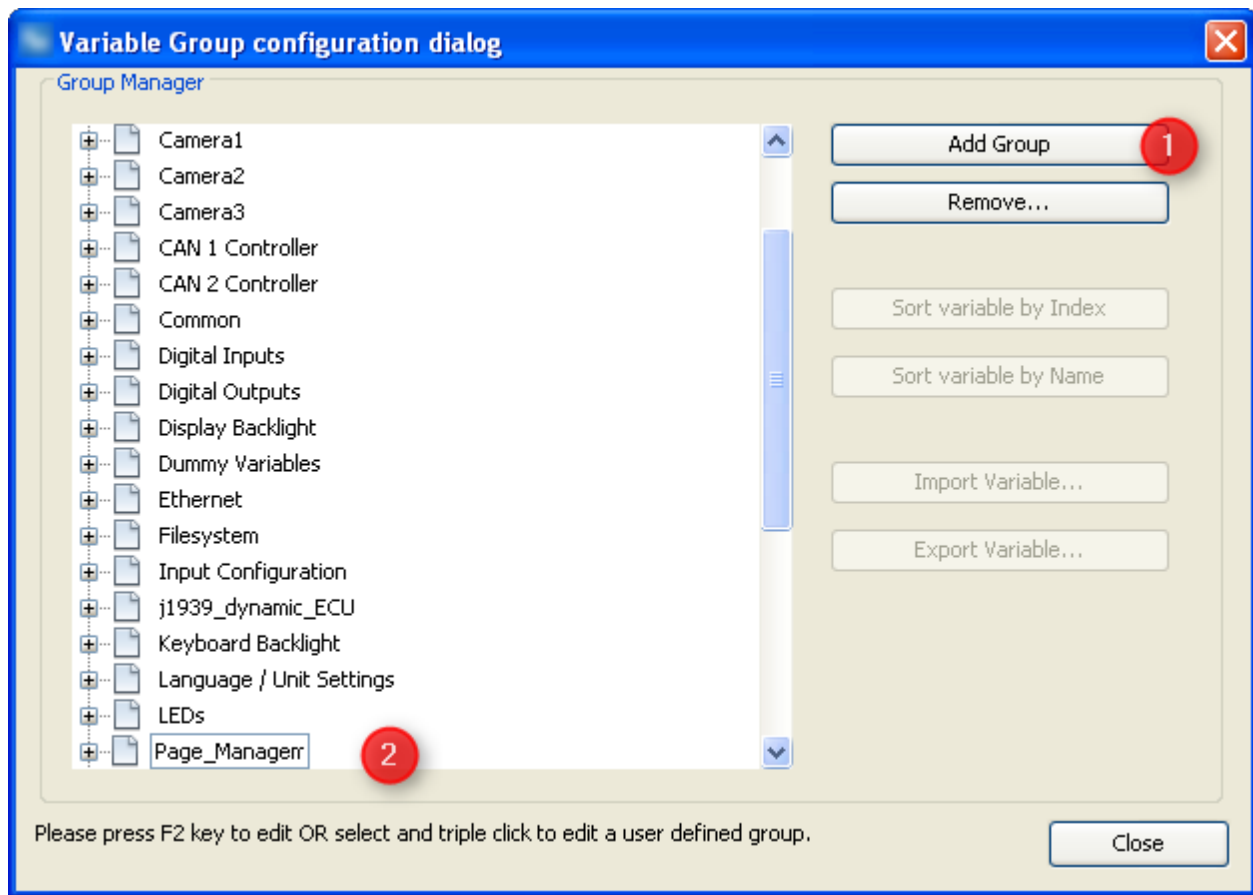
Initially the dialog only shows the predefined variable groups which can not be modified or deleted.
In order to add your group(s)
1. Press "Add Group" button
2. Type the name of your group in the box that appears and press "Enter".
The new group will be moved where it belongs alphabetically in the list of variable groups.
Like that it could move out of the visible part of the list.
You can create more groups by repeating the steps above.
In order to modify the name of one of your groups, follow the instruction on the bottom of the dialog.
Press the "Remove..." button in order to delete one of your groups. If it should already contain variables they will be moved to "Common" group.

Variable Manager

The variable manager dialog is the main place to create and maintain variables.

Open Variable Manager Dialog
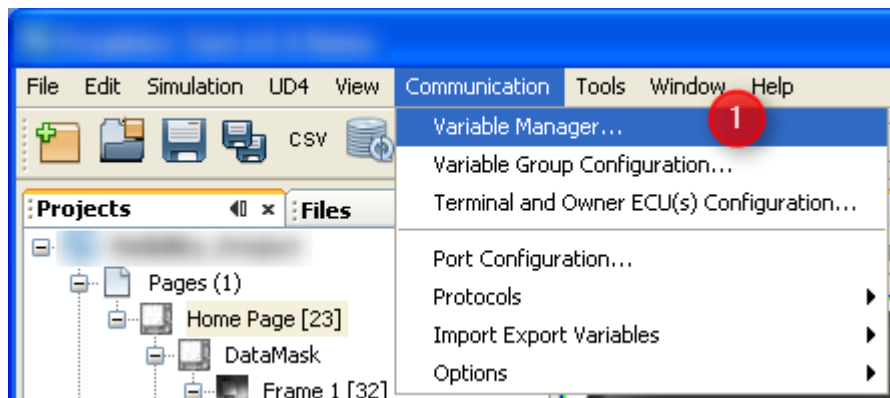
1. Open "Communication" menu.
2. Select "Variable Manager..."

Create New Variable
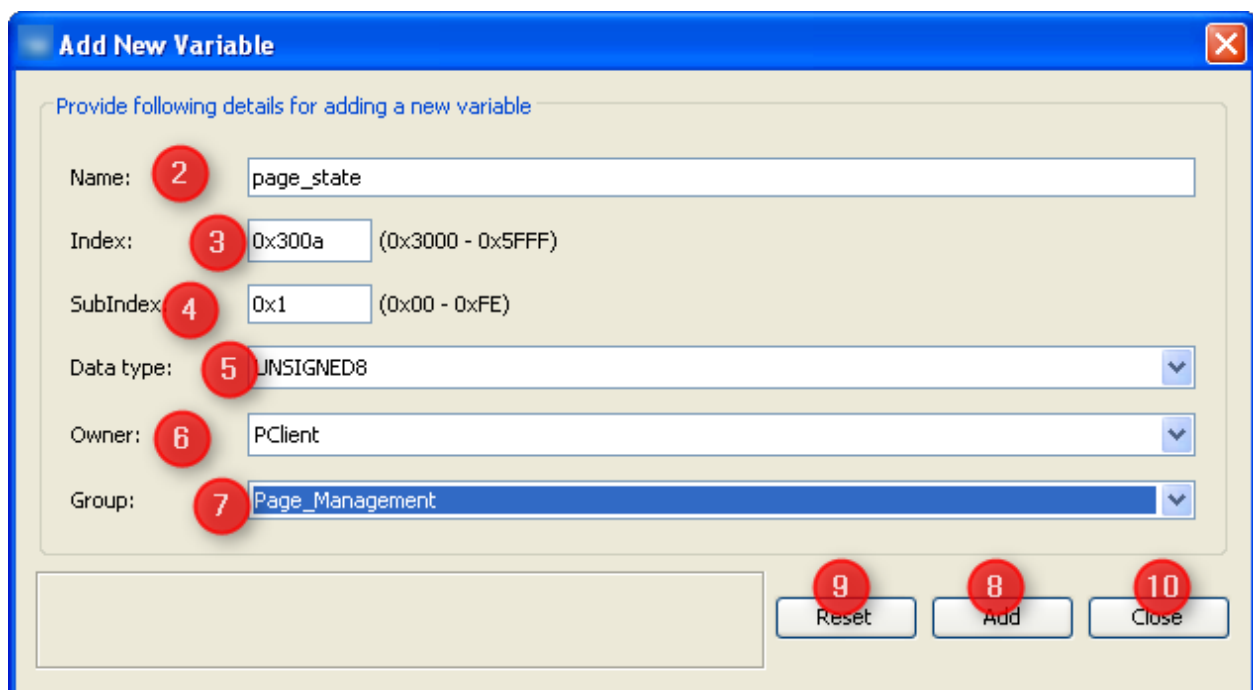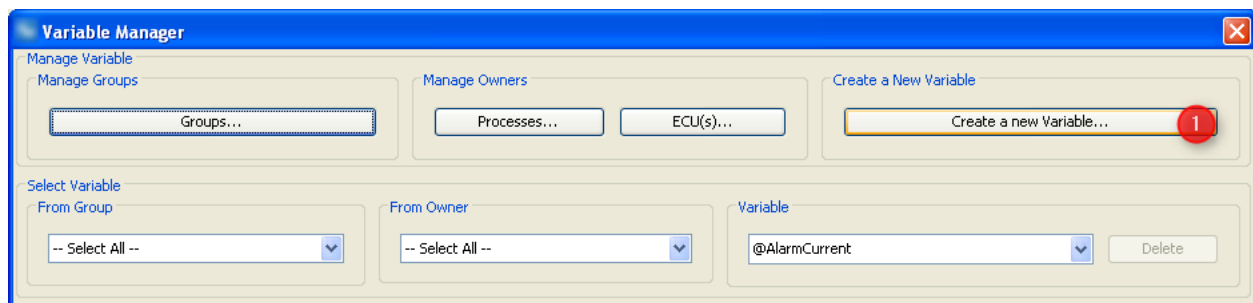
In order to add a new variable to the GUI definition

1. Press button "Create a new Variable..." in Variable Manager dialog. This opens the "Add New Variable" dialog.
2. Type a valid name that you will recognize when you want to attach this variable to DDOs later or when you want to use it in a script.

3. Define the index for the variable. If you should use CANopen this will be the index published in EDS file.
4. Define subindex for the new variable. See [dedicated section of variable concept](#)□154 for recommendations on index and subindex.
5. Choose data type for your variable only for CANopen you can not reduce the size in CAN messages. Otherwise main criteria is to make the variable big enough for all values it should transport.
6. Select owner of the variable. See section [owner concept](#)□154 in base document for criteria about which owner a variable should get.
7. Select a group for the new variable. See section [organizing principles](#)□154 in base document for some recommendations on group definitions.
8. Press "Add" button to add the new variable to GUI definition. Every press on "Add" will create a new variable - as long as name and index/subindex combination are unique.
9. If you want to create more than one variable you can either directly modify your settings OR press "Reset" first to get the dialog cleaned up first.
10. When you are finished creating variables, press "Close" button to leave the dialog and return to "Variable Manager".

Maintain and Modify Variables

Variable manager provides all the properties of all variables so that you can check the predefined ones and modify or delete your own variables.
Before actually deleting a variable it is recommended to make sure that it isn't used anymore in [variable view](#)□146.
Variable manager provides the following options and properties to maintain variables.
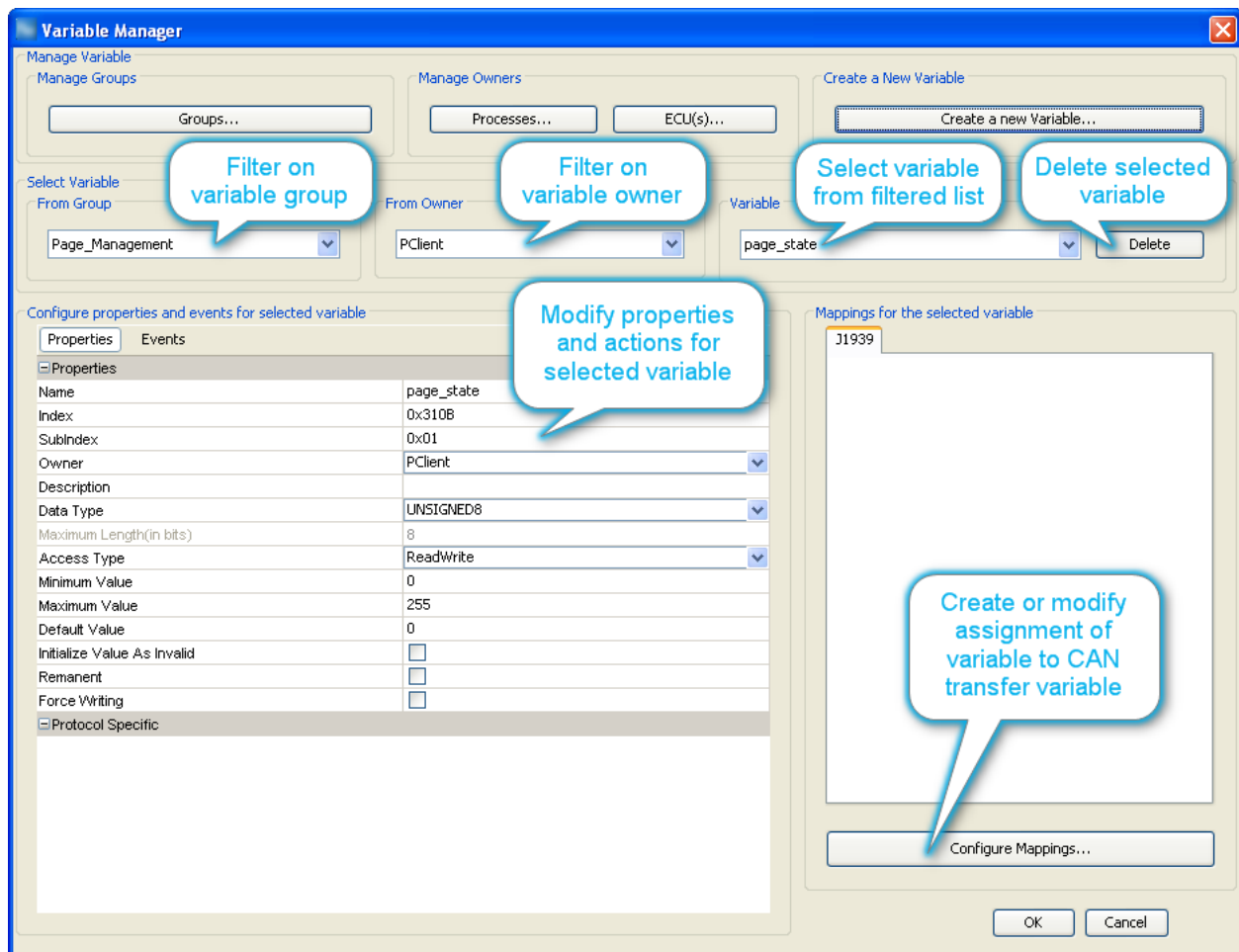
The rest of this page explains properties and actions for variables.

The filters mentioned in the image above should be self-explanatory.

The configuration of CAN mappings is explained with the CAN protocols, see CAN Freestyle[308], CANopen[328] and J1939[360].

Properties

Name

The name of a variable is used to identify the variable within the program mainly. In order to find a variable in this dialog, attach it to a DDO or use it in a script this name is required.

Therefore it is recommended to use a logic naming convention that will support working with variable. This will not only help you in your work but also others using it in the future - think of maintenance and maybe even our support team helping you with this GUI definition at some point of time.

Index

On the display side index and subindex are the main criteria to identify a variable - so it has to be unique within the GUI definition.

If you should use CANopen this value is directly exposed in the EDS file describing the object dictionary. Therefore the value range for index is limited to the proprietary range of CANopen indices. Range 0x2000 ... 0x2FFF is reserved for predefined variables so that your own variables can occupy the range 0x3000 ... 0x5FFF.

Like using variable groups proper selection of index and subindex can help to keep the GUI definition maintainable. See [dedicated section of variable concept](#)[154] for recommendations on index and subindex.

Warning: Although one can change index and subindex here they render all references to this variable useless. I.e. if you have to modify index or subindex after assigning the variable anywhere (DDO, mapping, action ...) you have to re-visit all these assignments and assign them again.

SubIndex

Subindex is directly tied to index - every subindex can only occur for one variable per index.

If you set subindex to 0 then this is the only variable for this index. If you skip subindex 0 you can define up to 254 (0x01 ... 0xFE) subindices per variable thus using the index like a group definition containing subindices that also belong together logically.

See property "index" above and [dedicated section of variable concept](#)[154] for recommendations on index and subindex.

Owner

Every variable needs an owner and only the owner is allowed to actually change the value of the variable. Here you can assign either one of the predefined or your self defined owners.

There is an extra page for [creation and maintenance of owners](#)[160]. Recommendations about owner assignment can be found on [general variable page](#)[154].

Description

This property exists for documentation and maintenance - please take the time to provide a short comment about the purpose and specific use of this variable.

Like that this property can make it easier to understand the GUI definition - especially for persons not involved in its development, e.g. maintainers or our support team.

Data Type

For CANopen this is how the variable will be maintained in object dictionary, mapped to PDOs and published in EDS. For other protocols or internal use you should choose a format covering the variable's purpose while avoiding to waste space in CAN messages.

Internally all numeric variables are stored as 32 bit anyways - so if in doubt and not using CANopen "INTEGER32" or "UNSIGNED32" should work for most applications. When adding the variable to mappings you may have to reduce its mapped size because default is to reserve the amount of bits defined here.

Custom data type creation is explained at the [top of this page](#)[165].

Maximum Length (in bits/bytes)

For numeric data types the length in bits is part of the data type and just shown here - i.e. this property is not editable for numeric types.

For strings this property defines the maximum length in bytes. This length is only enforced for user input in the program as well as on the device. Otherwise longer strings can be written to and read from string variables.

Please note that the display internally uses UTF-8 encoding which uses up to four bytes for special characters e.g. used in Russian or Asian language. So if you want the string to contain such characters you should make sure there is enough room. If you stick to

basic ASCII encoding (values 0 ... 127) then UTF-8 is the same as ASCII and every character takes exactly one byte.

Access Type

This property allows some optimization and is required for CANopen support. Generally the owner of a variable has full access and can modify its value whenever he needs to. Access type describes what others can do with the variable: If you e.g. set the variable as "ReadOnly" then a write request will not even be sent to the owner but the attempt to write will directly fail.
Full access to the variable is provided with access type "ReadWrite" - so if the variable should not need limitations this is the value to choose.

Default Value

The variable is initialized with this value - depending on "Initialize Value As Invalid" below you may never actually see this value on screen.

Initialize Value as Invalid

Set this property to consider the variable value invalid until a first value is provided by the owner. DDOs typically appear disabled (grey mask on top) while the value is invalid.
If you want to consider the "Default Value" above a valid value of the variable even though it was not set by the owner you should uncheck this property.

Remanent

Set this property if variable value should persist over power cycles. This property only actually modifies user defined variables owned by PClient. For predefined variables it's an information about what the owner does.

Force Writing

Normally there is some optimization in place which only forwards "new" variable values if the value was modified. I.e. PClient only forwards write requests to the variable owner if the value is different from the current one. The other way around it only forwards the "new" value sent by the owner if the value differs from the last one received.
This is helpful in most cases where e.g. the value is periodically transferred in a CAN message.
There are use cases though where every update or write request has to be forwarded i.e. the write itself is more meaningful than the value it transports. An example from the predefined variables is "@BeeperRepetitions" which should start a beeper sequence which could have the same amount as repetitions as the last one. For these cases "Force Writing" should be checked.

Protocol Specific

Based on the CAN protocol use to communicate with variable owner there could be additional properties required. These properties would appear here.
One example for these additional properties is index/subindex of the variable in the owner's object dictionary. This is currently only used if the owner is a CANopen ECU. Protocol specific properties are explained on the pages of the protocols, see CAN Freestyle[308], CANopen[328] and J1939[342] .

Events

The "Events" tab allows to assign actions to specific events. By default no actions are assigned.

OnValueChangeByOwner

This event is triggered whenever the owner of the variable changes its value i.e. really sends a different value or "Force Writing" is true.
If you should see any "Event Options" for this event - ignore them like PClient does.
Possible actions that can be assigned are

- "No Action" which is the default
- "Set Value" to assign a constant value to another variable, e.g. to reset a counter or trigger something attached to a variable where "Force Writing" is true.
- "Scale Value" allows modifying the new variable value before it is distributed within the display. The following calculation will be applied using integer arithmetics:

  variable_value = (((**received_value** + **offset1**) * **multiplier**) / **divisor**) + **offset2**
- "Execute Script" allows to execute a script and provides additional parameter "Execute on every Xth Event" so that it is possible to execute the script only for e.g. every third variable update. This action should only be used for variables which are rarely updated since otherwise it could slow down PClient.

OnValueChangeRequest

This event is triggered whenever an entity that is NOT the owner of the variable tries to write a value to the variable. The event is only triggered if the value differs from the one the variable currently has or "Force Writing" is true for the variable.
If you should see any "Event Options" for this event - ignore them like PClient does.
Possible actions that can be assigned are

- "No Action" which is the default
- "Set Value" to assign a constant value to another variable, e.g. to reset a counter or trigger something attached to a variable where "Force Writing" is true.
- "Scale Value" allows modifying the value to be written before it is forwarded to the owner. The following calculation will be applied using integer arithmetics:

  ```
  value_to_forward_to_owner = ((( requested_value +
  offset1 ) * multiplier ) / divisor ) + offset2
  ```
- "Execute Script" allows to execute a script and provides additional parameter "Execute on every Xth Event" so that it is possible to execute the script only for e.g. every sixth write request to the variable. This action should only be used for variables where write requests are applied rarely since otherwise it could slow down PClient.

4.3   Pre-Defined Variables

Hardware Daemon

Terminal Hardware

@TerminalHardware_NoOfEntries

"Number of entries in this group"

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as	Default Value	Min Value	Max Value

							Invalid			
0x2000	0x0	Integer	8 Bit	Read Only	disabled	disabled	enabled	6	6	6

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	Yes	Yes

@Terminal

"Type of Terminal"
Either the terminal this PClient was compiled for or that was detected / configured.

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x2000	0x3	String	255 Byte	Read Only	disabled	disabled	enabled	NO VALUE		

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	Yes	Yes

@PartNumber

"Vendor Part Number"
32 byte string as stored in EEPROM

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x2000	0x4	String	255 Byte	Read Only	disabled	disabled	enabled	NO VALUE		

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	Yes	Yes

@BrandOwnerPartNumber

"Brand Owner's Part Number"
32 byte string as stored in EEPROM

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Value	Default Value	Min Value	Max Value

							Invalid			
0x2000	0x5	String	255 Byte	Read Only	disabled	disabled	enabled	NO VALUE		

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	Yes	Yes

@SerialNumber

"Serial number of the terminal"
12 byte string as stored in EEPROM. Format is usually YYWWNNNHH where YY is the manufacturing year, WW is the manufacturing calendar week NNN the sequence number in that week and HH the hardware revision.
For EGScore devices the structure is TTT-YYWWNNNNHH-SS, where TTT is the device type and SS is the software revision, i.e. the versions of OS and application installed in the production. The rest is the same as above.

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x2000	0x6	String	255 Byte	Read Only	disabled	disabled	enabled	NO VALUE		

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	Yes	Yes

@SystemOperatingTime

"Operating Time of the device"
The operating time of the device in seconds. The time is only updated every 6 minutes.

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x2000	0x7	Integer	32 Bit	Read Only	disabled	disabled	enabled	-1	-1	2147483647

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	Yes	Yes

SD Card and USB Stick

@SDUSB_NoOfEntries

"Number of entries in this group"

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x2010	0x0	Integer	8 Bit	Read Only	disabled	disabled	enabled	12	12	12

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	Yes	Yes

@USBMemStatus

Check status of USB memory stick.Write 0 to this variable to trigger unmounting USB stick. Wait until variable reads as '0' before actually removing the stick.
Bit 0:
1 = memory stick inserted
0 = no memory stick present

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x2010	0x9	Integer	8 Bit	Read/ Write	disabled	disabled	enabled	-1	-1	1

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	Yes	Yes

@USBMemSize

"Memory size of first partition of USB stick inserted."
Resolution: 1 MB (e.g. 512 refers to a 512 MB card)

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x2010	0xa	Integer	32 Bit	Read Only	disabled	disabled	enabled	-1	-1	2147483647

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	Yes	Yes

@USBMemUsed

"Memory size used for files on first partition of USB stick inserted."
Resolution: 1 MB (e.g. 51 refers to a 51 MB used on card)

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x2010	0xb	Integer	32 Bit	Read Only	disabled	disabled	enabled	-1	-1	2147483647

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	Yes	Yes

@USBMountPath

"Filesystem path to where the USB stick is mounted"

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x2010	0xc	String	255 Byte	Read Only	disabled	disabled	enabled	NO VALUE		

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	Yes	Yes

Terminal Software

@TerminalSoftware_NoOfEntries

"Number of entries in this group"

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x2020	0x0	Integer	8 Bit	Read Only	disabled	disabled	enabled	16	16	16

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	Yes	Yes

@SystemUptime

"Time elapsed since Terminal was started"
Time in ms since Terminal was started.

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x2020	0x1	Integer	32 Bit	Read Only	disabled	disabled	enabled	-1	-1	2147483647

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	Yes	Yes

@EnableServiceMode

0: Service Mode Disabled1: Enable service Mode (after reboot OS update process will be started)
With this variable the OS update process can be started. Write a 1 to it to enable service mode. Service mode will only start after the device was rebooted (write value 0xb007 to @PWR_CurrentPowerMode). Note that the service mode can only be exited by a) installing something or b) disconnecting Ignition (clamp 15).

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x2020	0x4	Integer	2 Bit	Read/Write	disabled	disabled	enabled	0	-1	1

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	Yes	Yes

@OSKernelVersion

"Kernel version active in this terminal as reported by OS."
Complete Linux Kernel version as a string.

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x2020	0x8	String	255 Byte	Read Only	disabled	disabled	enabled	NO VALUE		

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	Yes	Yes

Power Modes

@PowerModes_NoOfEntries

"Number of entries in this group"

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x2030	0x0	Integer	8 Bit	Read Only	disabled	disabled	enabled	19	19	19

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	Yes	Yes

@PWR_TimeToSleepMode

"Define time in low power mode after which terminal switches to sleep mode (processor off, only RAM refreshed)"
Time in seconds.
Terminal switches to sleep mode after it spent this time in low power mode.
In low power mode processor keeps running, only a configurable set of components (e.g. display, backlights) can be switched off.
In sleep mode only the RAM gets refreshed while the rest of the processor is switched off.

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x2030	0x1	Integer	32 Bit	Read/ Write	enabled	disabled	enabled	60	-1	2332800

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	Yes	Yes

@PWR_TimeToPowerOff

"Define time in sleep mode after which terminal is switched off. "
Time in seconds.
Terminal switches off after it spent this time in sleep mode.
In sleep mode only the RAM gets refreshed while the rest of the processor is switched off.

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x2030	0x2	Integer	32 Bit	Read/ Write	enabled	disabled	enabled	120	-1	2332800

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	Yes	Yes

@PWR_LowPowerDisplayState

"State of the display in low power mode."
0: Off
1: On

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x2030	0x3	Integer	8 Bit	Read/Write	enabled	disabled	enabled	0	-1	1

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	Yes	Yes

@PWR_LowPowerDisplayBacklightState

"State of the display backlight in low power mode."
0: Off
1: On

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x2030	0x4	Integer	8 Bit	Read/Write	enabled	disabled	enabled	0	-1	1

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	Yes	Yes

@PWR_LowPowerKeypadBacklightState

"State of the keypad backlight in low power mode."
0: Off
1: On

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x2030	0x5	Integer	8 Bit	Read/Write	enabled	disabled	enabled	0	-1	1

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	No	Yes

@PWR_LowPower_RGB_LED_State

"State of the multi color RGB LED functionality in low power mode."
0: Off
1: On

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x2030	0x7	Integer	8 Bit	Read/Write	enabled	disabled	enabled	0	-1	1

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	Yes	Yes

@PWR_LowPowerBeeperState

"State of the Beeper functionality in low power mode."
0: Off
1: On

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x2030	0x8	Integer	8 Bit	Read/Write	enabled	disabled	enabled	0	-1	1

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	No	No	No

@PWR_LowPowerKeypadState

"State of the Keypad in low power mode."
0: Off
1: On

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x2030	0x9	Integer	8 Bit	Read/Write	enabled	disabled	enabled	0	-1	1

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	No	Yes

@PWR_LowPowerEncoderState

"State of the Encoder in low power mode."
0: Off
1: On

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x2030	0xa	Integer	8 Bit	Read/Write	enabled	disabled	enabled	0	-1	1

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	No	Yes

@PWR_LowPowerTouchState

"State of the touch screen in low power mode."
0: Off
1: On

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x2030	0xb	Integer	8 Bit	Read/Write	enabled	disabled	enabled	0	-1	1

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	Yes	Yes

@PWR_LowPowerConsoleState

"State of the serial console in low power mode."
0: Off
1: On

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x2030	0xc	Integer	8 Bit	Read/Write	enabled	disabled	enabled	1	-1	1

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	No	Yes

@PWR_LowPowerOutput1State

"State of the 1st digital output in low power mode."
0: Off
1: On

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x2030	0xd	Integer	8 Bit	Read/ Write	enabled	disabled	enabled	0	-1	1

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	No	Yes

@PWR_LowPowerOutput2State

"State of the 2nd digital output in low power mode."
0: Off
1: On

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x2030	0xe	Integer	8 Bit	Read/ Write	enabled	disabled	enabled	0	-1	1

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	No	Yes

@PWR_LowPowerOutput3State

"State of the 3rd digital output in low power mode."
0: Off
1: On

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x2030	0xf	Integer	8 Bit	Read/ Write	enabled	disabled	enabled	0	-1	1

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	No	Yes

@PWR_LowPowerCamOutputState

"State of the camera supply voltage in low power mode."
0: Off
1: On

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x2030	0x10	Integer	8 Bit	Read/Write	enabled	disabled	enabled	0	-1	1

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	No	No	No

@PWR_LowPowerVideoState

"State of the video decoder in low power mode. Note that on devices with an i.MX6 based CPU only the 1st and 3rd video input is suspended."
0: Off
1: On

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x2030	0x11	Integer	8 Bit	Read/Write	enabled	disabled	enabled	0	-1	1

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	No	Yes

@PWR_CurrentPowerMode

"Current Power Mode. See detailed description."
0: On. Can only be set if Ignition (clamp 15) is connected.
1: Low Power Mode. WARNING: Timer for going into sleep mode will only start if Ignition (clamp 15) is disconnected!
2: Low Power Mode with timer to automatically switch to sleep mode is enabled (write-only, after setting this value, the power mode will be set to 1). See @PWR_TimeToSleepMode. Only supported on certain devices.
3: Silent-on mode (read-only, can not be enabled via variable)
0xb007: Reboot. This does a complete cold reboot of the device (decimal value: 45063)
0x4a17: Halt the device (decimal value 18967). The device will shut down but the power supplies are not switched off as long as ignition is on.

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x2030	0x12	Integer	32 Bit	Read/Write	disabled	enabled	enabled	0	-1	45063

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	Yes	Yes

@PWR_SleepModePropagationTime

"This time in seconds is added to the @PWR_TimeToSleepMode before going into sleep mode. Currently there is no need to change this variable."
With this variable it is possible to delay going into sleep mode so that the application can do something (e.g. sending special CAN messages before sleep). Since currently the application will not handle the event that signals going into sleep mode there is no need to change the propagation time.

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x2030	0x13	Integer	8 Bit	Read/Write	enabled	enabled	enabled	0	0	30

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	Yes	Yes

@PWR_LowPowerEncoderBacklightState

"State of the encoder backlight in low power mode."
0: Off
1: On

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x2030	0x14	Integer	8 Bit	Read/Write	enabled	disabled	enabled	0	-1	1

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
No	Yes	No	Yes

@PWR_LowPowerDigitalInput1State

"State of the 1st digital input in low power mode."

0: Off
1: On
Only available on devices with digital inputs.

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x2030	0x15	Integer	8 Bit	Read/Write	enabled	disabled	enabled	0	-1	1

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	No	Yes

@PWR_LowPowerDigitalInput2State

"State of the 2nd digital input in low power mode."
0: Off
1: On
Only available on devices with digital inputs.

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x2030	0x16	Integer	8 Bit	Read/Write	enabled	disabled	enabled	0	-1	1

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	No	Yes

@PWR_LowPowerDigitalInput3State

"State of the 3rd digital input in low power mode."
0: Off
1: On
Only available on devices with digital inputs.

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x2030	0x17	Integer	8 Bit	Read/Write	enabled	disabled	enabled	0	-1	1

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	No	Yes

@PWR_LowPowerDigitalInput4State

"State of the 4th digital input in low power mode."
0: Off
1: On
Only available on devices with digital inputs.

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x2030	0x18	Integer	8 Bit	Read/Write	enabled	disabled	enabled	0	-1	1

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	No	Yes

@PWR_LowPower1stStatusLEDState

"State of the 1st status LED functionality in low power mode."
0: Off
1: On
Only available on devices equipped with such a status LED

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x2030	0x19	Integer	8 Bit	Read/Write	enabled	disabled	enabled	0	-1	1

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	No	Yes

@PWR_LowPower2ndStatusLEDState

"State of the 2nd status LED functionality in low power mode."
0: Off
1: On
Only available on devices equipped with such a status LED

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x2030	0x1a	Integer	8 Bit	Read/Write	enabled	disabled	enabled	0	-1	1

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	No	Yes

@PWR_TimeFromSilentOnToLowPower

"Define time spent in silent-on mode while no other wake source is enabled until power mode is switched to low power. "
Time in seconds.
When device starts in silent-on mode it will automatically switch to low power mode after this time if no other wake source is enabled (e.g. ignition). Only functional on devices that support booting into silent-on mode.

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x2030	0x1b	Integer	32 Bit	Read/ Write	enabled	disabled	enabled	65	-1	2333280

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
No	No	Yes	Yes

LEDs

@LED_NoOfEntries

"Number of entries in this group"

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x2040	0x0	Integer	8 Bit	Read Only	disabled	disabled	enabled	14	14	14

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	Yes	Yes

@LED_MultiColor

"The value of this variable represents a RGB color value"
0xffffff means white color, 0xff0000 red, 0x000000 off, 0x888888 is grey and so on.

For more details see user manual: FAQ -> How can I use the multicolor LED?

Index	Subindex	Storage	Length	Access	Remanent	Force Writing	Initialize as	Default	Min Value	Max Value

		Type	Type		g	Invalid	Value		
0x2040	0x2	Integer	32 Bit	Read/Write	disabled	disabled	enabled	0	-1	16777215

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	Yes	Yes

@LEDMultiColorRepetitions

"Number of repetitions of @LEDMultiColorPattern"
Number of intervals to perform. For more details see user manual: FAQ -> How can I use the multicolor LED?

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x2040	0xd	Integer	32 Bit	Read/Write	disabled	disabled	enabled	0	0	2147483647

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	Yes	Yes

@LEDMultiColorPattern

"Run a blink pattern on MultiColorLED"
When setting a value to this variable, a blink pattern is played on the device's multi color LED. The pattern is bitwise encoded into the value of this variable (20 values for on and off).
This pattern will be played with color set in @LED_MultiColor
e.g.: value is 5, bitwise: 00000000000000000101 resulting blink pattern:
100ms on
100ms off
100ms on
1700ms off

The pattern will be repeated for @LEDMultiColorRepetition times.

For more details see user manual: FAQ -> How can I use the multicolor LED?

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x2040	0xe	Integer	32 Bit	Read/Write	disabled	disabled	enabled	0	0	2147483647

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	Yes	Yes

Sensors

@Sensors_NoOfEntries

"Number of entries in this group"

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x2050	0x0	Integer	8 Bit	Read Only	disabled	disabled	enabled	6	6	6

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	Yes	Yes

@SensorTerminalTempC

"get estimated °C temperature inside terminal"
°C value as measured by internal temperature sensor. Value is in °C.

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x2050	0x1	Integer	32 Bit	Read Only	disabled	disabled	enabled	-1	-2000	2000

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	Yes	Yes

@SensorAmbientLight

"Standardized ambient light value"
Percentage of ambient light intensity in 10th of % (0: complete darknes (0.0%), 1000: full brightness (100.0%))

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x2050	0x3	Integer	32 Bit	Read Only	disabled	disabled	enabled	-1	-1	1000

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	Yes	Yes

@SensorSupplyVoltage

"Input voltage of power supply in mV"
Input voltage on permanent power supply (clamp 30) in mV e.g. 13294 means 13.294V

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x2050	0x5	Integer	32 Bit	Read Only	disabled	disabled	enabled	-1	-1	60000

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	Yes	Yes

Digital Inputs

@DigitalInputs_NoOfEntries

"Number of entries in this group"

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x2080	0x0	Integer	8 Bit	Read Only	disabled	disabled	enabled	11	11	11

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	No	Yes

@DigitalInput01

"Get state of digital input 01"
Only valid for terminals equipped with such an input
0: low voltage on input pin
1: high voltage on input pin

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x2080	0x2	Integer	2 Bit	Read Only	disabled	disabled	enabled	-1	-1	1

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	No	Yes

@DigitalInput02

"Get state of digital input 02"
Only valid for terminals equipped with such an input
0: low voltage on input pin
1: high voltage on input pin

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x2080	0x3	Integer	2 Bit	Read Only	disabled	disabled	enabled	-1	-1	1

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	No	Yes

@DigitalInput03

"Get state of digital input 03"
Only valid for terminals equipped with such an input
0: low voltage on input pin
1: high voltage on input pin

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x2080	0x4	Integer	2 Bit	Read Only	disabled	disabled	enabled	-1	-1	1

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	No	Yes

@DigitalInput04

"Get state of digital input 04"
Only valid for terminals equipped with such an input
0: low voltage on input pin
1: high voltage on input pin

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value

0x2080	0x5	Integer	2 Bit	Read Only	disabled	disabled	enabled	-1	-1	1

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	No	Yes

Analog Inputs

@AnalogInputs_NoOfEntries

"Number of entries in this group"

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x2090	0x0	Integer	8 Bit	Read Only	disabled	disabled	enabled	10	10	10

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	No	Yes

@AnalogInput01

"Get value of analog input 01"
Only valid for terminals equipped with such an input.
Software supports voltage on the respective input in 10 millivolts, e.g. a value of 1234 corresponds to 12,34 V.
Please refere to the hardware documentation to find out which value range and precision is supported by a specific hardware.

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x2090	0x1	Integer	32 Bit	Read Only	disabled	disabled	enabled	-1	-1	2147483647

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	No	Yes

@AnalogInput02

"Get value of analog input 02"
Only valid for terminals equipped with such an input.
Software supports voltage on the respective input in 10 millivolts, e.g. a value of 1234 corresponds to 12,34 V.

Please refer to the hardware documentation to find out which value range and precision is supported by a specific hardware.

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x2090	0x2	Integer	32 Bit	Read Only	disabled	disabled	enabled	-1	-1	2147483647

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	No	Yes

@AnalogInput03

"Get value of analog input 03"
Only valid for terminals equipped with such an input.
Software supports voltage on the respective input in 10 millivolts, e.g. a value of 1234 corresponds to 12,34 V.
Please refere to the hardware documentation to find out which value range and precision is supported by a specific hardware.

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x2090	0x3	Integer	32 Bit	Read Only	disabled	disabled	enabled	-1	-1	2147483647

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	No	Yes

@AnalogInput04

"Get value of analog input 04"
Only valid for terminals equipped with such an input.
Software supports voltage on the respective input in 10 millivolts, e.g. a value of 1234 corresponds to 12,34 V.
Please refere to the hardware documentation to find out which value range and precision is supported by a specific hardware.

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x2090	0x4	Integer	32 Bit	Read Only	disabled	disabled	enabled	-1	-1	2147483647

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	No	Yes

FrequencyInputs

@FrequencyInputs_NoOfEntries

"Number of entries in this group"

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x2091	0x0	Integer	8 Bit	Read Only	disabled	disabled	enabled	1	1	1

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
No	Yes	No	Yes

@FrequencyInput01

Measured frequency of input 1 in Hz
This variable represents the measured frequency of input 1 in Hz.

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x2091	0x1	Unsigned	16 Bit	Read Only	disabled	disabled	enabled	0	0	10000

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
No	Yes	No	Yes

Input Configuration

@Inputconfiguration_NoOfEntries

"Number of entries in this group"

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x2095	0x0	Integer	8 Bit	Read Only	disabled	disabled	enabled	22	22	22

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	No	Yes

@InputConfiguration01

"Configure input 01 for digital/analog (voltage sensor) or for current sensor operation."
Configure input 01 for digital/analog (voltage sensor) or for current sensor operation. Value 0 and 1: digital and analog operation (measured value in @DigitalInput01 and @AnalogInput01); Value 2: current sensor operation (measured value in @AnalogInput01).

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x2095	0x1	Integer	8 Bit	Read/Write	enabled	disabled	enabled	0	-1	2

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
No	Yes	No	Yes

@InputConfiguration02

"Configure input 02 for digital/analog (voltage sensor) or for current sensor operation."
Configure input 02 for digital/analog (voltage sensor) or for current sensor operation. Value 0 and 1: digital and analog operation (measured value in @DigitalInput02 and @AnalogInput02); Value 2: current sensor operation (measured value in @AnalogInput02).

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x2095	0x2	Integer	8 Bit	Read/Write	enabled	disabled	enabled	0	-1	2

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
No	Yes	No	Yes

@InputConfiguration03

"Configure input 03 for digital/analog (voltage sensor) or for current sensor operation."
Configure input 03 for digital/analog (voltage sensor) or for current sensor operation. Value 0 and 1: digital and analog operation (measured value in @DigitalInput03 and @AnalogInput03); Value 2: current sensor operation (measured value in @AnalogInput03).

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value

0x2095	0x3	Integer	8 Bit	Read/Write	enabled	disabled	enabled	0	-1	2

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
No	Yes	No	Yes

@InputConfiguration04

"Configure input 04 for digital/analog (voltage sensor) or for current sensor operation."
Configure input 04 for digital/analog (voltage sensor) or for current sensor operation. Value 0 and 1: digital and analog operation (measured value in @DigitalInput04 and @AnalogInput04); Value 2: current sensor operation (measured value in @AnalogInput04).

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x2095	0x4	Integer	8 Bit	Read/Write	enabled	disabled	enabled	0	-1	2

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
No	Yes	No	Yes

@Input1hysterese

"On devices with i.MX 35 CPU one step is approx. 11mv. On devices with i.MX 6 CPU one step is 1mV."
The hysteresis when the analog input shall signal a value change.
For devices with i.MX 35 CPU: This is a digital value of the 10Bit A/D converter betwenn 0 and 1023. One step is about 11mV.
For devices with i.MX 6 CPU: The value represents 1mV directly.

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x2095	0xb	Integer	16 Bit	Read/Write	enabled	disabled	enabled	1	-1	1023

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	No	Yes

@Input2hysterese

"On devices with i.MX 35 CPU one step is approx. 11mv. On devices with i.MX 6 CPU one step is 1mV."
The hysteresis when the analog input shall signal a value change.
For devices with i.MX 35 CPU: This is a digital value of the 10Bit A/D converter betwenn 0

and 1023. One step is about 11mV.
For devices with i.MX 6 CPU: The value represents 1mV directly.

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x2095	0xe	Integer	16 Bit	Read/Write	enabled	disabled	enabled	1	-1	1023

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	No	Yes

@Input3hysterese

"On devices with i.MX 35 CPU one step is approx. 11mv. On devices with i.MX 6 CPU one step is 1mV."
The hysteresis when the analog input shall signal a value change.
For devices with i.MX 35 CPU: This is a digital value of the 10Bit A/D converter betwenn 0 and 1023. One step is about 11mV.
For devices with i.MX 6 CPU: The value represents 1mV directly.

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x2095	0x11	Integer	16 Bit	Read/Write	enabled	disabled	enabled	1	-1	1023

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	No	Yes

@Input4hysterese

"On devices with i.MX 35 CPU one step is approx. 11mv. On devices with i.MX 6 CPU one step is 1mV."
The hysteresis when the analog input shall signal a value change.
For devices with i.MX 35 CPU: This is a digital value of the 10Bit A/D converter betwenn 0 and 1023. One step is about 11mV.
For devices with i.MX 6 CPU: The value represents 1mV directly.

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x2095	0x14	Integer	16 Bit	Read/Write	enabled	disabled	enabled	1	-1	1023

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	No	Yes

Digital Outputs

@DigitalOutputs_NoOfEntries

"Number of entries in this group"

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x20a0	0x0	Integer	8 Bit	Read Only	disabled	disabled	enabled	10	10	10

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	No	Yes

@DigitalOutput01

"Set state of digital output 1"
Only valid for terminals equipped with such an output.
The output is an open drain / open collector output. This means that when the output is set to 1, ground is connected to the output. If set to 0, the output is disconnected.

0: disconnect output
1: connect to ground

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x20a0	0x1	Integer	2 Bit	Read/Write	disabled	disabled	enabled	0	-1	1

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	No	Yes

@DigitalOutput02

"Set state of digital output 02"
Only valid for terminals equipped with such an output.
The output is an open drain / open collector output. This means that when the output is set to 1, ground is connected to the output. If set to 0, the output is disconnected.

0: disconnect output

1: connect to ground

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x20a0	0x2	Integer	2 Bit	Read/Write	disabled	disabled	enabled	0	-1	1

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	No	Yes

@DigitalOutput03

"Set state of digital output 03"
Only valid for terminals equipped with such an output.
The output is an open drain / open collector output. This means that when the output is set to 1, ground is connected to the output. If set to 0, the output is disconnected.

0: disconnect output
1: connect to ground

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x20a0	0x3	Integer	2 Bit	Read/Write	disabled	disabled	enabled	0	-1	1

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	No	Yes

@DigitalOutputLoopback01

"Value of the Digital Output 1 Loopback connection"
New generation of devices have a loopback connection for the digital outputs which can read back the real set value and detect errors with this (see @DigitalOutputFailure0x).

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x20a0	0x10	Unsigned	2 Bit	Read Only	disabled	disabled	enabled	-1	0	3

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS

No	Yes	No	Yes

@DigitalOutputLoopback02

"Value of the Digital Output 2 Loopback connection"
New generation of devices have a loopback connection for the digital outputs which can read back the real set value and detect errors with this (see @DigitalOutputFailure0x).

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x20a0	0x11	Unsigned	2 Bit	Read Only	disabled	disabled	enabled	-1	0	3

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
No	Yes	No	Yes

@DigitalOutputLoopback03

"Value of the Digital Output 3 Loopback connection"
New generation of devices have a loopback connection for the digital outputs which can read back the real set value and detect errors with this (see @DigitalOutputFailure0x).

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x20a0	0x12	Unsigned	2 Bit	Read Only	disabled	disabled	enabled	-1	0	3

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
No	Yes	No	Yes

@DigitalOutputFailure01

Shows if digital output 1 has detected a failure (set value != loopback value)
If the value of @DigitalOutput0x and @DigitalOutputLoopback0x is not the same, this variable signals the failure.
0: No failure
1: Failure detected

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x20a0	0x20	Unsigned	8 Bit	Read Only	disabled	disabled	enabled	0	0	255

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
No	Yes	No	Yes

@DigitalOutputFailure02

Shows if digital output 2 has detected a failure (set value != loopback value)
If the value of @DigitalOutput0x and @DigitalOutputLoopback0x is not the same, this variable signals the failure.
0: No failure
1: Failure detected

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x20a0	0x21	Unsigned	8 Bit	Read Only	disabled	disabled	enabled	0	0	255

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
No	Yes	No	Yes

@DigitalOutputFailure03

Shows if digital output 3 has detected a failure (set value != loopback value)
If the value of @DigitalOutput0x and @DigitalOutputLoopback0x is not the same, this variable signals the failure.
0: No failure
1: Failure detected

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x20a0	0x22	Unsigned	8 Bit	Read Only	disabled	disabled	enabled	0	0	255

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
No	Yes	No	Yes

Display Backlight

@DisplayBacklight_NoOfEntries

"Number of entries in this group"

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value

0x20b0	0x0	Integer	8 Bit	Read Only	disabled	disabled	enabled	17	17	17

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	Yes	Yes

@DispBacklightIntensity

"Set display backlight intensity directly or activate automatic mode"
0 ... 100: set backlight intensity constant to this percentage
else: Automatic backlight algorithm is active
To make it clear: To activate the automatic backlight dimming based on the ambient light: Set a value that is NOT between 0 and 100 (e.g. -1). To deactivate the auto backlight dimming simply set the static value between 0 and 100.

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x20b0	0x1	Integer	8 Bit	Read/Write	enabled	disabled	enabled	80	-1	101

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	Yes	No

@DispBacklightAmbientLowPassed

"Value calculated from @SensorAmbientLight using @DispBacklightTimeConstant and @DispBacklightFilterFactor. "
@DispBacklightAmbientLowPassed is calculated any @DispBacklightTimeConstant milliseconds as
(@DispBacklightAmbientLowPassed * @DispBacklightFilterFactor) +
(@SensorAmbientLight * (1000 - @DispBacklightFilterFactor) / 1000

Any manual write to this variable will be ignored.

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x20b0	0x2	Integer	16 Bit	Read/Write	enabled	disabled	enabled	80	0	1000

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	Yes	No

@DispBacklightTimeConstant

"Defines how often @DispBacklightAmbientLowPassed is calculated"
Value in milliseconds how often the low passed ambient light sensor is calculated for the automatic display backlight dimming. See @DispBacklightAmbientLowPassed.

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x20b0	0x3	Integer	16 Bit	Read/Write	enabled	disabled	enabled	500	100	1000

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	Yes	No

@DispBacklightFilterFactor

"Defines balance of old and new values incorporated into @DispBacklightAmbientLowPassed"
see @DispBacklightAmbientLowPassed

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x20b0	0x4	Integer	16 Bit	Read/Write	enabled	disabled	enabled	800	-1	1000

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	Yes	No

@DispBacklightIntensity0

"Intensity of display backlight when @DispBacklightAmbient-LowPassed == 0%"
Display backlight intensity between specified values is linearly interpolated.

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x20b0	0x5	Integer	8 Bit	Read/Write	enabled	disabled	enabled	10	-1	100

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	Yes	No

@DispBacklightIntensity10

"Intensity of display backlight when @DispBacklightAmbient-LowPassed == 10%"

Display backlight intensity between specified values is linearly interpolated.

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x20b0	0x6	Integer	8 Bit	Read/Write	enabled	disabled	enabled	15	-1	100

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	Yes	No

@DispBacklightIntensity20

"Intensity of display backlight when @DispBacklightAmbient-LowPassed == 20%"
Display backlight intensity between specified values is linearly interpolated.

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x20b0	0x7	Integer	8 Bit	Read/Write	enabled	disabled	enabled	20	-1	100

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	Yes	No

@DispBacklightIntensity30

"Intensity of display backlight when @DispBacklightAmbient-LowPassed == 30%"
Display backlight intensity between specified values is linearly interpolated.

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x20b0	0x8	Integer	8 Bit	Read/Write	enabled	disabled	enabled	30	-1	100

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	Yes	No

@DispBacklightIntensity40

"Intensity of display backlight when @DispBacklightAmbient-LowPassed == 40%"
Display backlight intensity between specified values is linearly interpolated.

Index	Subindex	Storage	Length	Access	Remanent	Force Writing	Initialize as	Default	Min Value	Max Value

		Type		Type		g	Invalid	Value		
0x20b0	0x9	Integer	8 Bit	Read/Write	enabled	disabled	enabled	40	-1	100

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	Yes	No

@DispBacklightIntensity50

"Intensity of display backlight when @DispBacklightAmbient-LowPassed == 50%"
Display backlight intensity between specified values is linearly interpolated.

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x20b0	0xa	Integer	8 Bit	Read/Write	enabled	disabled	enabled	50	-1	100

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	Yes	No

@DispBacklightIntensity60

"Intensity of display backlight when @DispBacklightAmbient-LowPassed == 60%"
Display backlight intensity between specified values is linearly interpolated.

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x20b0	0xb	Integer	8 Bit	Read/Write	enabled	disabled	enabled	60	-1	100

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	Yes	No

@DispBacklightIntensity70

"Intensity of display backlight when @DispBacklightAmbient-LowPassed == 70%"
Display backlight intensity between specified values is linearly interpolated.

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value

| 0x20b0 | 0xc | Integer | 8 Bit | Read/ Write | enabled | disabled | enabled | 70 | -1 | 100 |

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	Yes	No

@DispBacklightIntensity80

"Intensity of display backlight when @DispBacklightAmbient-LowPassed == 80%"
Display backlight intensity between specified values is linearly interpolated.

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x20b0	0xd	Integer	8 Bit	Read/ Write	enabled	disabled	enabled	80	-1	100

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	Yes	No

@DispBacklightIntensity90

"Intensity of display backlight when @DispBacklightAmbient-LowPassed == 90%"
Display backlight intensity between specified values is linearly interpolated.

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x20b0	0xe	Integer	8 Bit	Read/ Write	enabled	disabled	enabled	90	-1	100

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	Yes	No

@DispBacklightIntensity100

"Intensity of display backlight when @DispBacklightAmbient-LowPassed == 100%"
Display backlight intensity between specified values is linearly interpolated.

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x20b0	0xf	Integer	8 Bit	Read/ Write	enabled	disabled	enabled	100	-1	100

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	Yes	No

@DispBacklightOffset

"Offset to be configured by end user (in case values above are fixed by brand owner)"
see @DispBacklightAmbientLowPassed

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x20b0	0x10	Integer	8 Bit	Read/Write	enabled	disabled	enabled	0	-1	100

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	Yes	No

@DispBacklightStartupIntensity

"Backlight Intensity that is set by the bootloader at cold boot until hardware daemon is started."
This value is directly stored in the EEPROM of the device.
The intensity will be overwritten by the value in @DispBacklightIntensity once the project is loaded.

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x20b0	0x12	Integer	8 Bit	Read/Write	disabled	disabled	enabled	80	5	100

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	Yes	Yes

@MinDispBacklightIntensity

"This is the Minimal Backlight Intensity of the display which the hardware daemon allows to set. "
The variable is set to 5 by default to prohibit accidentally turning off the display by setting the @DispBacklightIntensity to 0 without a "way back". To be able to set @DispBacklightIntensity to 0, set this variable to 0 first.

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value

0x20b0	0x13	Integer	8 Bit	Read/ Write	enabled	disabled	enabled	5	0	100

Supported on Platforms:

i.MX 35	**i.MX 6 A-Series**	**i.MX 6 B-Series**	**ISOBUS**
Yes	Yes	Yes	No

Keyboard Backlight

@KeyboardBacklight_NoOfEntries

"Number of entries in this group"

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x20c0	0x0	Integer	8 Bit	Read Only	disabled	disabled	enabled	1	1	17

Supported on Platforms:

i.MX 35	**i.MX 6 A-Series**	**i.MX 6 B-Series**	**ISOBUS**
Yes	Yes	No	Yes

@KeybBacklightIntensity

"Manually set Keyboard Backlight intensity."
0 ... 100: set keyboard backlight intensity constant to this percentage
Automatic keyboard backlight dimming is not supported.
On devices that support @KeyBacklightColor, this variable will dimm the selected color linearily.

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x20c0	0x1	Integer	8 Bit	Read/ Write	enabled	disabled	enabled	100	-1	100

Supported on Platforms:

i.MX 35	**i.MX 6 A-Series**	**i.MX 6 B-Series**	**ISOBUS**
Yes	Yes	No	Yes

@KeyBacklightColor

Change color of display keyboard backlight
New device generations support the changing of the color of the keyboard backlight. Any RGB value can be set here to change the color. This will then be treated as the master color which cann be dimmed with @KybBacklightIntensity.
Value examples:
0xffffff: white

0xff0000: red
0x00ff00: green
0x0000ff: blue
0x00ffff: cyan

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x20c0	0x2	Unsigned	32 Bit	Read/Write	enabled	disabled	enabled	16777215	0	16777215

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
No	Yes	No	Yes

@EncoderBacklightIntensity

"Manually set Encoder Backlight intensity."
0 ... 100: set keyboard backlight intensity constant to this percentage
Automatic encoder backlight dimming is not supported.
On devices that support @EncoderBacklightColor, this variable will dimm the selected color linearily.

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x20c0	0x3	Integer	8 Bit	Read/Write	enabled	disabled	enabled	100	-1	100

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
No	Yes	No	Yes

@EncoderBacklightColor

Change color of encoder backlight
New device generations support the changing of the color of the encoder backlight. Any RGB value can be set here to change the color. This will then be treated as the master color which cann be dimmed with @EncoderBacklightIntensity.
Value examples:
0xffffff: white
0xff0000: red
0x00ff00: green
0x0000ff: blue
0x00ffff: cyan

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as	Default Value	Min Value	Max Value

							Invalid			
0x20c0	0x4	Unsigned	32 Bit	Read/Write	enabled	disabled	enabled	16777215	0	16777215

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
No	Yes	No	Yes

@KeyBacklightStartupIntensity

"Keypad Intensity that is set by the bootloader at cold boot until hardware daemon is started. This value is directly stored in the EEPROM of the device."
Keyboard Backlight Intensity that is set by the bootloader at cold boot until hardware daemon is started. This value is directly stored in the EEPROM of the device.

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x20c0	0x12	Integer	8 Bit	Read/Write	disabled	disabled	enabled	100	0	100

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	No	No	No

@KeyBacklightStartupColor

Change color AND intensity of keyboard backlight at cold boot.
New device generations support the changing of the color of the keyboard backlight at cold boot. The value that is set represents a 24Bit RGB value for color AND intensity. After application startup the variables @KeyBacklightColor, @KeyBacklightIntensity, @EncoderBacklightColor and @EncoderBacklightIntensity will be the master values for the backlight intensity and color

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x20c0	0x13	Unsigned	24 Bit	Read/Write	disabled	disabled	enabled	255	0	16777215

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
No	Yes	No	Yes

Beeper

@Beeper_NoOfEntries

"Number of entries in this group"

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x20d0	0x0	Integer	8 Bit	Read Only	disabled	disabled	enabled	17	17	18

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	Yes	Yes

@BeeperMaxVolume

"Maximum volume to be used for any beep. Also applied as maximum volume for ISO-VT."
Percentage of maximum volume the beeper can provideto be used in combination with
@BeeperClickVolume and @BeeperAlarmVolume. Typically set by end user based on
ambient noise and personal preferences.
0: beeper completely deactivated

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x20d0	0x1	Integer	8 Bit	Read/Write	enabled	disabled	enabled	40	-1	100

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	Yes	Yes

@BeeperClickVolume

"Volume to be used to acknowledge keypresses with a short beep"
Percentage of @BeeperMaxVolume to be used to when a key is pressed.
0: no beep on key click
If the value is greater 0, a click sound will appear whenever a hard-/softkey or the touch
screen is pressed.

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x20d0	0x2	Integer	8 Bit	Read/Write	enabled	enabled	enabled	0	-1	100

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	Yes	Yes

| Yes | Yes | Yes | Yes |

@BeeperAlarmVolume

"Volume to be used for alarm beeps"
Percentage of @BeeperMaxVolume to be used to alarm user. Typically set at runtime based on severity of an alarm.

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x20d0	0x3	Integer	8 Bit	Read/Write	enabled	enabled	enabled	100	-1	100

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	Yes	Yes

@BeeperEventVolume

"Volume to be used for beep pattern assigned to events"
Percentage of @BeeperMaxVolume to be used to when a beep pattern of an event is played.
This could be a user setting or a variable settting in the script triggered by the event.
0: no beep on events

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x20d0	0x4	Integer	8 Bit	Read/Write	enabled	enabled	enabled	100	-1	100

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	Yes	Yes

@BeeperMinEmergencyAlarmVolume

"Require minimum alarm volume for emergency alarms"
Percentage of maximum volume the beeper can provide. If an emergency alarm with a sound attached occurs, the beeper volume is the maximum of this value and the one calculated from @BeeperMaxVolume and @BeeperAlarmVolume.
Set this value to 0 to have full control about the alarm volume with the variables mentioned above.

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value

| 0x20d0 | 0x5 | Integer | 8 Bit | Read/Write | enabled | enabled | enabled | 0 | -1 | 100 |

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	Yes	Yes

@BeeperOnTime

"On time in one beeper interval"
Time in interval where beeper is switched on [ms]
One interval consists of @BeeperOnTime + @BeeperOffTime
Note: On devices with a speaker, short interval times (less than 300ms) might not trigger the speaker at the beginning of the pattern.

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x20d0	0x7	Integer	16 Bit	Read/Write	disabled	enabled	enabled	0	-1	32767

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	Yes	Yes

@BeeperOffTime

"Off time in one beeper interval"
Time in interval where beeper is switched off [ms]
One interval consists of @BeeperOnTime + @BeeperOffTime
Note: On devices with a speaker, short interval times (less than 300ms) might not trigger the speaker at the beginning of the pattern.

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x20d0	0x8	Integer	16 Bit	Read/Write	disabled	enabled	enabled	0	-1	32767

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	Yes	Yes

@BeeperRepetitions

"Writing: Number of intervals to perform Reading: Beeper active/non-active"
Beeper will repeat sequence of @BeeperOnTime followed by @BeeperOffTime for @BeeperRepetitions times
This will only succeed if beeper is not occupied by higher priority beeps originating from

alarms.

Beeper priorities are:

- Key click (lowest)
- Event based beep patterns
- these variables
- Control audio command (sent to ISO VT)
- Alarms (with respect to their beep priority, highest)

Value is 0 if the beeper is not active. Value is -1 if the beeeper is active.

Note: On devices with a speaker, short interval times (less than 300ms) might not trigger the speaker at the beginning of the pattern.

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x20d0	0x9	Integer	16 Bit	Read/Write	disabled	enabled	enabled	0	-1	32767

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	Yes	Yes

@BeeperClickOn

"Switch beeper on or of for a click"
There should be no need for customers to use this variable. To enable the "click" feature, just apply a proper value to @BeeperClickVolume variable.

The click sound can be enabled by setting this variable to 1 with the volume set in @BeeperClickVolume. But only if it was not turned on by a higher priority before (higher priorities are: event -> VT -> Alarm).
The sound will automatically be switched off after 200ms (this variable will be reset to 0).

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x20d0	0xa	Integer	2 Bit	Read/Write	disabled	enabled	enabled	0	-1	1

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	Yes	Yes

@BeeperAlarmOn

"Switch beeper on or of for an alarm"
Setting to 1 will switch the beeper with volume @BeeperAlarmVolume on. Switching this variable to one will turn all other "@BeeperXXXOn" variables to 0.
Setting 0 to this variable will switch the beeper off.

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x20d0	0xb	Integer	2 Bit	Read/Write	disabled	enabled	enabled	0	-1	1

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	Yes	Yes

@BeeperEventOn

"Switch beeper on or of for an event"
Setting to 1 will switch the beeper with volume @BeeperEventVolume on, but only if it was not turned on by a higher priority before (higher priorities are: VT -> Alarm).
Setting 0 to this variable will switch the beeper off (if no higher priority variable is == 1).

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x20d0	0xc	Integer	2 Bit	Read/Write	disabled	enabled	enabled	0	-1	1

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	Yes	Yes

@BeeperVariableVolume

"Set beeper volume to be used by @BeeperRepetitions"
Percentage of @BeeperMaxVolume to be used for @BeeperRepetitions
0: no beep on repetitions

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x20d0	0xd	Integer	8 Bit	Read/Write	enabled	enabled	enabled	0	-1	100

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	Yes	Yes

@BeeperAlarmPattern

"Run an alarm's beep pattern"
When setting a value to this variable, a beep pattern is played on the device's beeper. The pattern is bitwise encoded into the value of this variable (20 values for on and off).

This pattern will be played with volume set in @BeeperAlarmVolume.
e.g.: value is 5, bitwise: 00000000000000000101 resulting beep pattern:
100ms on
100ms off
100ms on
1700ms off

The pattern will be repeated for @BeeperAlarmRepetition times.
Note: On devices with a speaker, short interval times (less than 300ms) might not trigger the speaker at the beginning of the pattern.

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x20d0	0xe	Integer	32 Bit	Read/ Write	disabled	enabled	enabled	0	-1	16777215

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	Yes	Yes

@BeeperEventPattern

"Run an event's beep pattern"
When setting a value to this variable, a beep pattern is played on the device's beeper. The pattern is bitwise encoded into the value of this variable (20 values for on and off).
This pattern will be played with volume set in @BeeperEventVolume.
e.g.: value is 5, bitwise: 00000000000000000101 resulting beep pattern:
100ms on
100ms off
100ms on
1700ms off

The pattern will be repeated for @BeeperEventRepetition times.
 Note: On devices with a speaker, short interval times (less than 300ms) might not trigger the speaker at the beginning of the pattern.

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x20d0	0xf	Integer	32 Bit	Read/ Write	disabled	enabled	enabled	0	-1	16777215

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	Yes	Yes

@BeeperAlarmRepetitions

"Writing: Number of intervals to perform. Reading: Beeper active/non-active" Note: On devices with a speaker, short interval times (less than 300ms) might not trigger the speaker at the beginning of the pattern.
Defines how often @BeeperAlarmPattern will be repeated.

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x20d0	0x10	Integer	16 Bit	Read/Write	disabled	enabled	enabled	0	-1	32767

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	Yes	Yes

@BeeperEventRepetitions

"Writing: Number of intervals to perform. Reading: Beeper active/non-active" Note: On devices with a speaker, short interval times (less than 300ms) might not trigger the speaker at the beginning of the pattern.
Defines how often @BeeperEventPattern will be repeated.

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x20d0	0x11	Integer	16 Bit	Read/Write	disabled	enabled	enabled	0	-1	32767

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	Yes	Yes

@BeeperIsActive

"Provides the current state of the beeper processing."
Returns:
-1: Read Error
0: Beeper not active
1: AlarmPattern active
2: AlarmOn active
3: Repetition active
4: EventPattern active
5: EventOn active
6: ClickOn active

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value

0x20d0	0x12	Integer	8 Bit	Read Only	disabled	disabled	enabled	0	-1	6

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	Yes	Yes

@VolumeLevelSpeaker

Set the master volume of the internal speaker.
Use this variable to set the master volume level of the internal speaker (0% to 100%).

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x20d0	0x13	Integer	8 Bit	Read/ Write	enabled	enabled	enabled	100	0	100

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
No	Yes	Yes	No

@VolumeLevelLineOut

Set the master volume of the audio line out.
Use this variable to set the master volume level of the line out channel (0% to 100%).

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x20d0	0x14	Integer	8 Bit	Read/ Write	enabled	enabled	enabled	100	0	100

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
No	Yes	Yes	No

RTC

@RTC_NoOfEntries

"Number of entries in this group"

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value

0x2100	0x0	Integer	8 Bit	Read Only	disabled	disabled	enabled	7	7	11

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	Yes	Yes

@RTC_Year

"Local date: Year"
Current year from the device's RTC chip.

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x2100	0x1	Integer	16 Bit	Read/Write	disabled	disabled	enabled	-1	-1	2200

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	Yes	Yes

@RTC_Month

"Local date: Month"
Current month from the device's RTC chip.

1: January
2: February
...
12: December

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x2100	0x2	Integer	8 Bit	Read/Write	disabled	disabled	enabled	-1	-1	12

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	Yes	Yes

@RTC_Day

"Local date: Day"
Current day from the device's RTC chip.

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x2100	0x3	Integer	8 Bit	Read/Write	disabled	disabled	enabled	-1	-1	31

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	Yes	Yes

@RTC_Hours

"Local time: Hours"
Current hour from the device's RTC chip.

writing to this variable will also change @RTC_HoursAMPM and @RTC_AMPM accordingly

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x2100	0x5	Integer	8 Bit	Read/Write	disabled	disabled	enabled	-1	-1	23

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	Yes	Yes

@RTC_Minutes

"Local time: Minutes"
Current minute from the device's RTC chip.

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x2100	0x6	Integer	8 Bit	Read/Write	disabled	disabled	enabled	-1	-1	59

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	Yes	Yes

@RTC_Seconds

"Local time: Seconds"
Current second from the device's RTC chip.

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x2100	0x7	Integer	8 Bit	Read/Write	disabled	disabled	enabled	-1	-1	59

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	Yes	Yes

@RTC_HoursAMPM

"Same as @RTC_Hours, but in am/pm format"
Current hour in AMPM format from the device's RTC chip.

writing to this variable will also change @RTC_Hours accordingly

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x2100	0x8	Integer	8 Bit	Read/Write	disabled	disabled	enabled	-1	-1	12

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	Yes	Yes

@RTC_AMPM

"Am/pm flag for @RTC_HoursAMPM"
writing to this variable will also change @RTC_Hours accordingly
0: AM, 1: PM

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x2100	0x9	Integer	8 Bit	Read/Write	disabled	disabled	enabled	-1	-1	1

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	Yes	Yes

CAN 1 Controller

@CAN1Controller_NoOfEntries

"Number of entries in this group"

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x2120	0x0	Integer	32 Bit	Read Only	disabled	disabled	enabled	0	0	0

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	Yes	Yes

@CAN1_State

State of CAN controller 1
The following values describe the state of the CAN controller:
0x00: Controller switched off
0x01: Controller is active, no errors
0x02: Error active state
0x03: Error passive state
0x04: Bus off state
0xff: Device Error
It is possible to write the values 0 to this variable to switch the CAN controller off and 1 to switch it on.

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x2120	0x1	Integer	16 Bit	Read/Write	disabled	disabled	enabled	-1	-1	255

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	Yes	Yes

@CAN1_MsgReceived

"Messages received by terminal in last second."
Counter of messages read from the bus during the last second.

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x2120	0x2	Integer	32 Bit	Read Only	disabled	disabled	enabled	-1	-1	2147483647

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS

Yes	Yes	Yes	Yes

@CAN1_MsgSent

"Messages sent by terminal in last second."
Counter of massages sent to the bus by the PClient during the last second.

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x2120	0x3	Integer	32 Bit	Read Only	disabled	disabled	enabled	-1	-1	2147483647

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	Yes	Yes

@CAN1_Bitrate

Read or set the CAN bitrate of CAN Port 1. The value is in Bit per second.
If the bitrate is changed while the CAN controller is running, it will be shut down and the new bitrate will be set. If this was successful, the CAN controller will be started again with the new bitrate.
If the new bitrate could not be set, the old bitrate will be set again.
If the CAN controller is off when setting the new bitrate, it will not be started after setting the bitrate.

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x2120	0x4	Unsigned	32 Bit	Read/Write	disabled	enabled	enabled	250000	20000	1000000

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	Yes	Yes

CAN 2 Controller

@CAN2Controller_NoOfEntries

"Number of entries in this group"

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x2121	0x0	Integer	32 Bit	Read Only	disabled	disabled	enabled	0	0	0

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	Yes	Yes

@CAN2_State

State of CAN controller 2
The following values describe the state of the CAN controller:
0x00: Controller switched off
0x01: Controller is active, no errors
0x02: Error active state
0x03: Error passive state
0x04: Bus off state
0xff: Device Error
It is possible to write the values 0 to this variable to switch the CAN controller off and 1 to switch it on.

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x2121	0x1	Integer	16 Bit	Read/Write	disabled	disabled	enabled	-1	-1	255

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	Yes	Yes

@CAN2_MsgReceived

"Messages received by terminal in last second."
Counter of messages read from the bus during the last second.

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x2121	0x2	Integer	32 Bit	Read Only	disabled	disabled	enabled	-1	-1	2147483647

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	Yes	Yes

@CAN2_MsgSent

"Messages sent by terminal in last second."
Counter of massages sent to the bus by the PClient during the last second.

Index	Subindex	Storage	Length	Access	Remanent	Force Writing	Initialize as	Default	Min Value	Max Value

		Type		Type		g	Invalid	Value		
0x2121	0x3	Integer	32 Bit	Read Only	disabled	disabled	enabled	-1	-1	2147483647

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	Yes	Yes

@CAN2_Bitrate

Read or set the CAN bitrate of CAN Port 2. The value is in Bit per second.
If the bitrate is changed while the CAN controller is running, it will be shut down and the new bitrate will be set. If this was successful, the CAN controller will be started again with the new bitrate.
If the new bitrate could not be set, the old bitrate will be set again.
If the CAN controller is off when setting the new bitrate, it will not be started after setting the bitrate.

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x2121	0x4	Unsigned	32 Bit	Read/Write	disabled	enabled	enabled	250000	20000	1000000

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	Yes	Yes

CAN 3 Controller

@CAN3Controller_NoOfEntries

"Number of entries in this group"
Only supported on devices with 3 or more CAN controllers.

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x2122	0x0	Integer	32 Bit	Read Only	disabled	disabled	enabled	0	0	0

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
No	No	Yes	Yes

@CAN3_State

State of CAN controller 3
The following values describe the state of the CAN controller:
0x00: Controller switched off
0x01: Controller is active, no errors
0x02: Error active state
0x03: Error passive state
0x04: Bus off state
0xff: Device Error
It is possible to write the values 0 to this variable to switch the CAN controller off and 1 to switch it on.
Only supported on devices with 3 or more CAN controllers.

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x2122	0x1	Integer	16 Bit	Read/Write	disabled	disabled	enabled	-1	-1	255

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
No	No	Yes	Yes

@CAN3_MsgReceived

"Messages received by terminal in last second."
Counter of messages read from the bus during the last second.
Only supported on devices with 3 or more CAN controllers.

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x2122	0x2	Integer	32 Bit	Read Only	disabled	disabled	enabled	-1	-1	2147483647

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
No	No	Yes	Yes

@CAN3_MsgSent

"Messages sent by terminal in last second."
Counter of massages sent to the bus by the PClient during the last second.
Only supported on devices with 3 or more CAN controllers.

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value

0x2122	0x3	Integer	32 Bit	Read Only	disabled	disabled	enabled	-1	-1	2147483647

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
No	No	Yes	Yes

@CAN3_Bitrate

Read or set the CAN bitrate of CAN Port 3. The value is in Bit per second.
If the bitrate is changed while the CAN controller is running, it will be shut down and the new bitrate will be set. If this was successful, the CAN controller will be started again with the new bitrate.
If the new bitrate could not be set, the old bitrate will be set again.
If the CAN controller is off when setting the new bitrate, it will not be started after setting the bitrate.
Only supported on devices with 3 or more CAN controllers.

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x2122	0x4	Unsigned	32 Bit	Read/Write	disabled	enabled	enabled	250000	20000	1000000

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
No	No	Yes	Yes

CAN 4 Controller

@CAN4Controller_NoOfEntries

"Number of entries in this group"
Only supported on devices with 4 or more CAN controllers.

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x2123	0x0	Integer	32 Bit	Read Only	disabled	disabled	enabled	0	0	0

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
No	No	Yes	Yes

@CAN4_State

State of CAN controller 4

The following values describe the state of the CAN controller:
0x00: Controller switched off
0x01: Controller is active, no errors
0x02: Error active state
0x03: Error passive state
0x04: Bus off state
0xff: Device Error
It is possible to write the values 0 to this variable to switch the CAN controller off and 1 to switch it on.
Only supported on devices with 4 or more CAN controllers.

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x2123	0x1	Integer	16 Bit	Read/Write	disabled	disabled	enabled	-1	-1	255

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
No	No	Yes	Yes

@CAN4_MsgReceived

"Messages received by terminal in last second."
Counter of messages read from the bus during the last second.
Only supported on devices with 4 or more CAN controllers.

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x2123	0x2	Integer	32 Bit	Read Only	disabled	disabled	enabled	-1	-1	2147483647

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
No	No	Yes	Yes

@CAN4_MsgSent

"Messages sent by terminal in last second."
Counter of massages sent to the bus by the PClient during the last second.
Only supported on devices with 4 or more CAN controllers.

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x2123	0x3	Integer	32 Bit	Read Only	disabled	disabled	enabled	-1	-1	2147483647

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
No	No	Yes	Yes

@CAN4_Bitrate

Read or set the CAN bitrate of CAN Port 4. The value is in Bit per second.
If the bitrate is changed while the CAN controller is running, it will be shut down and the new bitrate will be set. If this was successful, the CAN controller will be started again with the new bitrate.
If the new bitrate could not be set, the old bitrate will be set again.
If the CAN controller is off when setting the new bitrate, it will not be started after setting the bitrate.
Only supported on devices with 4 or more CAN controllers.

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x2123	0x4	Unsigned	32 Bit	Read/Write	disabled	enabled	enabled	250000	20000	1000000

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
No	No	Yes	Yes

Filesystem

@Filesystem_NoOfEntries

"Number of entries in this group"

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x2130	0x0	Integer	8 Bit	Read Only	disabled	disabled	enabled	8	8	8

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	Yes	Yes

@CurrentDirectory

"Currently selected directory (full qualified file path)"
e.g.: "/opt/user_files/"

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Value	Default Value	Min Value	Max Value

							Invalid			
0x2130	0x1	String	255 Byte	Read/ Write	enabled	enabled	enabled	/proc		

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	Yes	Yes

@CurrentFile

"Currently selected file (file name only)"
e.g.: "test.txt"

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x2130	0x2	String	255 Byte	Read/ Write	enabled	enabled	enabled	cpuinfo		

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	Yes	Yes

@DestinationDirectory

"The destination directory where @CurrentDirectory and @CurrentFile can be copied or moved to"
e.g.: "/disk/usbsda1/".

Should not be written manually unless it's secured that the directory exists.

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x2130	0x3	String	255 Byte	Read/ Write	enabled	enabled	enabled	/tmp		

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	Yes	Yes

@CopyFileOrFolder

Copy or move directory or file.Also use '@USBMemStatus' variable if USB stick is involved.
0: No action
1: Copy @CurrentDirectory/@CurrentFile to @DestinationDirectory
2: Move @CurrentDirectory/@CurrentFile to @DestinationDirectory

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x2130	0x4	Integer	8 Bit	Read/Write	disabled	disabled	enabled	0	-1	2

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	Yes	Yes

@FileSystemPath

"A valid full qualified file system path"
e.g.: "/opt/user_files/"

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x2130	0x5	String	255 Byte	Read/Write	enabled	disabled	enabled	/opt		

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	Yes	Yes

@UsedMemoryPartition

"Shows the used memory in kilobytes of the partition on which @FileSystemPath is located"

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x2130	0x6	Integer	32 Bit	Read Only	disabled	disabled	enabled	-1	-1	2147483647

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	Yes	Yes

@FreeMemoryPartition

"Shows the free memory in kilobytes of the partition on which @FileSystemPath is located"

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value

0x2130	0x7	Integer	32 Bit	Read Only	disabled	disabled	enabled	-1	-1	2147483647

Supported on Platforms:

i.MX 35	**i.MX 6 A-Series**	**i.MX 6 B-Series**	**ISOBUS**
Yes	Yes	Yes	Yes

@CurrentFileSize

"Shows the file size of @CurrentFile in kilobytes"

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x2130	0x8	Integer	32 Bit	Read Only	disabled	disabled	enabled	-1	-1	2147483647

Supported on Platforms:

i.MX 35	**i.MX 6 A-Series**	**i.MX 6 B-Series**	**ISOBUS**
Yes	Yes	Yes	Yes

Ethernet

@Ethernet_NoOfEntries

"Number of entries in this group"

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x2140	0x0	Integer	8 Bit	Read Only	disabled	disabled	enabled	21	21	23

Supported on Platforms:

i.MX 35	**i.MX 6 A-Series**	**i.MX 6 B-Series**	**ISOBUS**
Yes	Yes	Yes	No

@Ethernet0_IPAddress

Currently set IpAddress (default 192.168.135.6).
Last IP address set via hardware daemon.
Please note that there are other sources for IP address such as start scripts and UD4
update via Ethernet.
The one set in a script may differ from this one but anyways be active.
So if more than one of the IP mechanisms should be used they should all be configured to
the same IP address and netmask.

Index	Subindex	Storage	Length	Access	Remanent	Force Writing	Initialize as	Default	Min Value	Max Value

		Type		Type		g	Invalid	Value		
0x2140	0x1	String	15 Byte	Read/Write	enabled	disabled	enabled	192.168.135.6		

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	Yes	No

@Ethernet0_Netmask

"Currently set Netmask, can be set via Variable (default 255.255.255.0)"

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x2140	0x2	String	15 Byte	Read/Write	enabled	disabled	enabled	255.255.255.0		

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	Yes	No

@Ethernet0_Gateway

"Currently set Gateway for Internet Connections (default 192.168.135.1)"

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x2140	0x3	String	15 Byte	Read/Write	enabled	disabled	enabled	192.168.135.1		

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	Yes	No

@Ethernet0_UpAtStart

"Should the adapter be started at device startup? 1 = yes 0 = no; Default = 0"

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value

0x2140	0x4	Integer	2 Bit	Read/Write	enabled	disabled	enabled	0	-1	1

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	Yes	No

@Ethernet1_IPAddress

"Currently set IpAddress (default 192.168.135.2)"

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x2140	0x5	String	15 Byte	Read/Write	enabled	disabled	enabled	192.168.135.2		

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	Yes	No

@Ethernet1_Netmask

"Currently set Netmask, can be set via Variable (default 255.255.255.0)"

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x2140	0x6	String	15 Byte	Read/Write	enabled	disabled	enabled	255.255.255.0		

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	Yes	No

@Ethernet1_Gateway

"Currently set Gateway for Internet Connections (default 192.168.135.1)"

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x2140	0x7	String	15 Byte	Read/Write	enabled	disabled	enabled	192.168.135.1		

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	Yes	No

@Ethernet1_UpAtStart

"Should the adapter be started at startup? 1 = yes 0 = no; Default = 0"

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x2140	0x8	Integer	2 Bit	Read/Write	enabled	disabled	enabled	0	-1	1

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	Yes	No

@Ethernet0_applynstart

"Allows applying Ethernet settings"
If set to 1 the 'new' or 'currently' available values in ip, netmask and gateway variables are applied to the network adapter. The adapter is started and the variable is set back to 0. Default is 0.

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x2140	0x14	Integer	2 Bit	Read/Write	disabled	disabled	enabled	0	-1	1

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	Yes	No

@Ethernet1_applynstart

"Allows applying Ethernet settings"
If set to 1 the 'new' or 'currently' available values in ip, netmask and gateway variables are applied to the network adapter. The adapter is started and the variable is set back to 0. Default is 0.

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x2140	0x15	Integer	2 Bit	Read/Write	disabled	disabled	enabled	0	-1	1

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	Yes	No

@Ethernet0_MasterConfig

"Master/Slave configuration for automotive ethernet devices. 1: Master, 0: Slave"
Only available on devices equipped with automotive ethernet (100BASE-TX, BroadR-Reach). If value of this variable is 1, the interface is configured as master (=> slave devices can be directly connected). If value is 0, the interface is configured as slave (=> master devices can be directly connected). All other values => invalid. Note that if ethernet is already running when this variable is changed, the ethernet interface will be switched off before the master/slave configuration is done and switched on again awhen the configuration was finished.

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x2140	0x16	Unsigned	2 Bit	Read/Write	enabled	disabled	enabled	0	0	3

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
No	No	Yes	No

RS232

@RS232_NoOfEntries

Number of variables in this group

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x2150	0x0	Integer	8 Bit	Read Only	disabled	disabled	enabled	1	0	1

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	Yes	Yes

@ConsoleMode

Change the Linux console mode of the device (new mode is active after reboot).
Possible values:
0: Console on RS232
1: Console disabled
2: Console on first USB serial adapter (use Null-Modem Cable)

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x2150	0x1	Unsigned	8 Bit	Read/Write	disabled	enabled	enabled	0	0	2

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	Yes	Yes

PClient

User Inputs

@UserInputs_NoOfEntries

"Number of entries in this group"

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x2070	0x0	Unsigned	8 Bit	Read Only	disabled	disabled	enabled	15	15	15

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	Yes	Yes

@UserEncoder1Clicks

"Amount of encoder clicks since last reset."
Counts encoder clicks
+1: encoder turned one click clockwise
-1: encoder turned one click anti-clockwise

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x2070	0x1	Integer	32 Bit	Read/Write	disabled	disabled	enabled	0	-2147483647	2147483647

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	No	Yes

@UserButtonPressed

"Keycode of button currently pressed. -1 if no button is pressed"
Please note that not all keys are available on all devices.
Key codes:
0 - ESC
1 - Home
2 - Three Arrow Key
3 - Enter (Encoder)
6 - Stop
200 - Left Key 1
201 - Left Key 2
202 - Left Key 3
203 - Left Key 4
204 - Left Key 5
205 - Left Key 6
100- Right Key 1
101 - Right Key 2
102 - Right Key 3
103 - Right Key 4
104 - Right Key 5
105 - Right Key 6
106 - Right Key 7
107 - Right Key 8

If multiple buttons are pressed at the same time, this variable only shows the last pressed key code. To handle multiple keys at the same time, use variable @UserMultiPressed_XX

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x2070	0x4	Integer	32 Bit	Read Only	disabled	disabled	enabled	-1	-1	2147483647

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	No	Yes

@UserButtonDoubleClick

"Keycode of last button double clicked."
Keycodes: see @UserButtonPressed

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x2070	0x6	Unsigned	32 Bit	Read Only	disabled	disabled	enabled	0	0	2147483647

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	No	Yes

@UserButtonReleased

"Keycode of last button released."
Keycodes: see @UserButtonPressed
-1 if there is at least one button pressed.

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x2070	0x7	Integer	32 Bit	Read Only	disabled	disabled	enabled	-1	-1	2147483647

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	No	Yes

@UserMultiButtonPressed_00

"Keys bitwise encoded so that press of multiple keys can be detected."
Bit pattern is device independent:
bit 0 ... 15: Softkeys right of display, counted from the top
bit 16 ... 32: Softkeys left of display, counted from the top

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x2070	0x8	Unsigned	32 Bit	Read Only	disabled	disabled	enabled	0	0	0

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	No	Yes

@UserMultiButtonPressed_01

"Keys bitwise encoded so that press of multiple keys can be detected. To be used if terminal has more than 32 keys or to keep bit pattern device independent as long as possible."
Bit pattern is device independent
bit 0 ... 15: Softkeys top of display, counted from the right (bit pattern similar to physical key position)
bit 16 ... 32: Softkeys bottom of display, counted from the right (bit pattern similar to physical key position)

Index	Subindex	Storage	Length	Access	Remanent	Force Writing	Initialize as	Default	Min Value	Max Value

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x2070	0x9	Unsigned	32 Bit	Read Only	disabled	disabled	enabled	0	0	0

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	No	Yes

@UserMultiButtonPressed_02

"Keys bitwise encoded so that press of multiple keys can be detected. To be used if terminal has more than 64 keys or to keep bit pattern device independent as long as possible."
Bit pattern is device dependent
Bit 0: ESC
Bit 1: home key
Bit 2: three arrow key
Bit 3: Stop key

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x2070	0xa	Unsigned	32 Bit	Read Only	disabled	disabled	enabled	0	0	0

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	No	Yes

@UserTouchPressedX

"X coordinate where touch screen was pressed last time"
Coordinates are given in display coordinates starting with (0, 0) in the upper left corner

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x2070	0xc	Unsigned	16 Bit	Read Only	disabled	disabled	enabled	0	0	1023

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	Yes	Yes

@UserTouchPressedY

"Y coordinate where touch screen was pressed last time"

Coordinates are given in display coordinates starting with (0, 0) in the upper left corner

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x2070	0xd	Unsigned	16 Bit	Read Only	disabled	disabled	enabled	0	0	1023

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	Yes	Yes

@UserTouchReleasedX

"X coordinate where touch screen was released last time"
Coordinates are given in display coordinates starting with (0, 0) in the upper left corner

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x2070	0xe	Unsigned	16 Bit	Read Only	disabled	disabled	enabled	0	0	1023

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	Yes	Yes

@UserTouchReleasedY

"Y coordinate where touch screen was released last time"
Coordinates are given in display coordinates starting with (0, 0) in the upper left corner

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x2070	0xf	Unsigned	16 Bit	Read Only	disabled	disabled	enabled	0	0	1023

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	Yes	Yes

@MultiTouchEnabled

Enable (value 1) or disable (value 0) multi touch functionality
If multi touch functionality is enabled, multiple objects can receive touch events (e.g. multiple buttons can be operated at the same time). When multi touch functionality is disabled, only one element on the screen can be touched.

Warning: When multi touch functionality gets disabled (setting the value of this variable from 1 to 0), all currently pressed objects will be cancelled.

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x2070	0x10	Integer	2 Bit	Read/Write	enabled	disabled	disabled	1	0	1

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
No	Yes	Yes	Yes

Camera1

@Camera1_NoOfEntries

"Number of entries in this group"

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x20e0	0x0	Integer	8 Bit	Read Only	disabled	disabled	enabled	10	10	10

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	Yes	No

@Cam1Brightness

"brightness setting"

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x20e0	0x1	Integer	8 Bit	Read/Write	enabled	disabled	disabled	0	-127	127

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	Yes	No

@Cam1Contrast

"contrast setting"

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x20e0	0x2	Integer	16 Bit	Read/Write	enabled	disabled	disabled	127	-1	255

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	Yes	No

@Cam1Saturation

"color saturation setting"

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x20e0	0x3	Integer	16 Bit	Read/Write	enabled	disabled	disabled	127	-1	255

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	Yes	No

@Cam1Hue

"hue setting"

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x20e0	0x4	Integer	8 Bit	Read/Write	enabled	disabled	disabled	0	-127	127

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	No	No	No

@Cam1Interface

"Path to device file that is used for video input 1. Default value is device dependent"
Names of the interface is terminal dependent. On terminals with only one camera input this variable is ignored.

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value

0x20e0	0x5	String	255 Byte	Read Only	enabled	disabled	disabled	/dev/video0		

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	Yes	No

@Cam1Name

"Name of camera position"
e.g. "rearview", "left sweeper"

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x20e0	0x6	String	255 Byte	Read/Write	enabled	disabled	disabled	Camera1		

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	Yes	No

@Cam1Rotation

"Rotation of the video in degrees.Possible values are 0 or 90 degree. Any other value will have no effect. Other rotation values can only be achieved in combination with @Cam1Mirror."
Set the video rotation. Possible values:
0, 90

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x20e0	0xc	Integer	16 Bit	Read/Write	disabled	enabled	disabled	0	0	90

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	Yes	No

@Cam1Mirror

"Set/Get Mirror settings for camera"
Possible values:
0: No mirroring
1: Horizontal mirroring
2: Vertical mirroring
3: Horizontal and Vertical mirroring

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x20e0	0xd	Integer	8 Bit	Read/Write	disabled	enabled	disabled	0	0	3

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	Yes	No

@Cam1XPosition

"Get or set the X position of the video image"
This variables controls the X position on the screen of the video image for this camera

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x20e0	0xe	Integer	16 Bit	Read/Write	disabled	enabled	disabled	0	0	65535

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	No	No

@Cam1YPosition

"Get or set the Y position of the video image"
This variables controls the Y position on the screen of the video image for this camera

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x20e0	0xf	Integer	16 Bit	Read/Write	disabled	enabled	disabled	0	0	65535

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	No	No

@Cam1Width

"Get or set the width of the video image."
This variables controls the width of the video image for this camera. On i.MX35 devices: Any value that is not a multiple of 8 might be truncated to the next smaller multiple of 8 by the hardware interface. On i.MX6 devices there is no restriction.

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x20e0	0x10	Integer	16 Bit	Read/Write	disabled	enabled	disabled	8	8	65535

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	No	No

@Cam1Height

"Get or set the height of the video image."
This variables controls the height of the video image for this camera. On i.MX35 devices: Any value that is not a multiple of 8 might be truncated to the next smaller multiple of 8 by the hardware interface. On i.MX6 devices there is no restriction.

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x20e0	0x11	Integer	16 Bit	Read/Write	disabled	enabled	disabled	8	8	65535

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	No	No

Camera2

@Camera2_NoOfEntries

"Number of entries in this group"

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x20e1	0x0	Integer	8 Bit	Read Only	disabled	disabled	enabled	10	10	10

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	Yes	No

@Cam2Brightness

"brightness setting"

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x20e1	0x1	Integer	8 Bit	Read/Write	enabled	disabled	disabled	0	-127	127

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	Yes	No

@Cam2Contrast

"contrast setting"

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x20e1	0x2	Integer	16 Bit	Read/Write	enabled	disabled	disabled	127	-1	255

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	Yes	No

@Cam2Saturation

"color saturation setting"

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x20e1	0x3	Integer	16 Bit	Read/Write	enabled	disabled	disabled	127	-1	255

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	Yes	No

@Cam2Hue

"hue setting"

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value

| 0x20e1 | 0x4 | Integer | 8 Bit | Read/Write | enabled | disabled | disabled | 0 | -127 | 127 |

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	Yes	No

@Cam2Interface

"Path to device file that is used for video input 2. Default value is device dependent"

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x20e1	0x5	String	255 Byte	Read Only	enabled	disabled	disabled	/dev/video0		

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	Yes	No

@Cam2Name

"Name of camera position"

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x20e1	0x6	String	255 Byte	Read/Write	enabled	disabled	disabled	Camera2		

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	Yes	No

@Cam2Rotation

"Rotation of the video in degrees.Possible values are 0 or 90 degree. Any other value will have no effect. Other rotation values can only be achieved in combination with @Cam2Mirror."

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x20e1	0xc	Integer	16 Bit	Read/Write	disabled	enabled	disabled	0	0	90

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	Yes	No

@Cam2Mirror

"Set/Get Mirror settings for camera"

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x20e1	0xd	Integer	8 Bit	Read/Write	disabled	enabled	disabled	0	0	3

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	Yes	No

@Cam2XPosition

"Get or set the X position of the video image"

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x20e1	0xe	Integer	16 Bit	Read/Write	disabled	enabled	disabled	0	0	65535

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	No	No

@Cam2YPosition

"Get or set the Y position of the video image"

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x20e1	0xf	Integer	16 Bit	Read/Write	disabled	enabled	disabled	0	0	65535

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	No	No

@Cam2Width

"Get or set the width of the video image."

This variables controls the width of the video image for this camera. On i.MX35 devices: Any value that is not a multiple of 8 might be truncated to the next smaller multiple of 8 by the hardware interface. On i.MX6 devices there is no restriction.

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x20e1	0x10	Integer	16 Bit	Read/Write	disabled	enabled	disabled	8	8	65535

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	No	No

@Cam2Height

"Get or set the height of the video image."
This variables controls the height of the video image for this camera. On i.MX35 devices: Any value that is not a multiple of 8 might be truncated to the next smaller multiple of 8 by the hardware interface. On i.MX6 devices there is no restriction.

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x20e1	0x11	Integer	16 Bit	Read/Write	disabled	enabled	disabled	8	8	65535

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	No	No

Camera3

@Camera3_NoOfEntries

"Number of entries in this group"

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x20e2	0x0	Integer	8 Bit	Read Only	disabled	disabled	enabled	10	10	10

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	No	No

@Cam3Brightness

"brightness setting"

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x20e2	0x1	Integer	8 Bit	Read/Write	enabled	disabled	disabled	0	-127	127

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	No	No

@Cam3Contrast

"contrast setting"

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x20e2	0x2	Integer	16 Bit	Read/Write	enabled	disabled	disabled	127	-1	255

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	No	No

@Cam3Saturation

"color saturation setting"

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x20e2	0x3	Integer	16 Bit	Read/Write	enabled	disabled	disabled	127	-1	255

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	No	No

@Cam3Hue

"hue setting"

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as	Default Value	Min Value	Max Value

						Invalid			
0x20e2	0x4	Integer	8 Bit	Read/Write	enabled	disabled	disabled	0	-127	127

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	No	No	No

@Cam3Interface

"Path to device file that is used for video input 3. Default value is device dependent"

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x20e2	0x5	String	255 Byte	Read Only	enabled	disabled	disabled	/dev/video0		

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	No	No

@Cam3Name

"Name of camera position"

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x20e2	0x6	String	255 Byte	Read/Write	enabled	disabled	disabled	Camera3		

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	No	No

@Cam3Rotation

"Rotation of the video in degrees.Possible values are 0 or 90 degree. Any other value will have no effect. Other rotation values can only be achieved in combination with @Cam3Mirror."

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value

| 0x20e2 | 0xc | Integer | 16 Bit | Read/ Write | disabled | enabled | disabled | 0 | 0 | 90 |

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	No	No

@Cam3Mirror

"Set/Get Mirror settings for camera"

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x20e2	0xd	Integer	8 Bit	Read/ Write	disabled	enabled	disabled	0	0	3

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	No	No

@Cam3XPosition

"Get or set the X position of the video image"

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x20e2	0xe	Integer	16 Bit	Read/ Write	disabled	enabled	disabled	0	0	65535

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	No	No

@Cam3YPosition

"Get or set the Y position of the video image"

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x20e2	0xf	Integer	16 Bit	Read/ Write	disabled	enabled	disabled	0	0	65535

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	No	No

@Cam3Width

"Get or set the width of the video image."
This variables controls the width of the video image for this camera. On i.MX35 devices: Any value that is not a multiple of 8 might be truncated to the next smaller multiple of 8 by the hardware interface. On i.MX6 devices there is no restriction.

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x20e2	0x10	Integer	16 Bit	Read/ Write	disabled	enabled	disabled	8	8	65535

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	No	No

@Cam3Height

"Get or set the height of the video image."
This variables controls the height of the video image for this camera. On i.MX35 devices: Any value that is not a multiple of 8 might be truncated to the next smaller multiple of 8 by the hardware interface. On i.MX6 devices there is no restriction.

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x20e2	0x11	Integer	16 Bit	Read/ Write	disabled	enabled	disabled	8	8	65535

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	No	No

CameraControl

@CameraControl_NoOfEntries

"Number of entries in this group"

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x20ef	0x0	Integer	8 Bit	Read Only	disabled	disabled	enabled	2	2	2

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS

Yes	Yes	Yes	No

@ActiveCamera

"Number of active camera as bit field."
Number of active camera as a bitfield:
Bit 1: Analog Camera 1 (0x1)
Bit 2: Analog Camera 2 (0x2)
Bit 3: Analog Camera 3 (0x4)
Bit 4: H264 Ethernet Camera (0x8)

On i.MX6 devices it is possible to show video of camera 1 and 2 or 2 and 3 at the same time. Camera 1 and 3 is not possible due to hardware limitations. A H264 video frame can be displayed any time. Note that a camera frame or page that was configured for camera 1 can not show the video of camera 2 when changing this variable and vice versa. Only a switch from camera 1 to camera 3 and back is possible (only applies to i.MX6 devices). On i.MX35 devices only one camera (and no H264) can be shown at the same time. A switch from camera 1 to 2 to 3 and back is possible (mind the size and position variables for each camera).

Invalid combinations and values set for this variable will not be accepted. Invalid values are for i.MX6 devices: 0x5, 0x7, 0xd and any value greater 0xe or less than 0x0. For i.MX35 devices: anything not equal to 0x0, 0x1, 0x2 or 0x4.

If @VideoStatus is 1 when changing this variable, the new camera will be active. If @VideoStatus is 0 when changing this variable, the new camera will not be active, @VideoStatus has to be set to 1 then.

This variable does NOT need to be used (and should not be used) if the application (e.g. PClient) handles video frames or pages automatically. This variable can in this case be seen as a debug tool.

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x20ef	0x1	Integer	8 Bit	Read/Write	disabled	enabled	enabled	0	0	14

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	Yes	No

@VideoStatus

"Global analog video status"
0: video completely off
1: video overlay active of at least 1 camera
Use this variable to manually switch video on or completely off in combination with @ActiveCamera

This variable does NOT need to be used if the application (e.g. PClient) handles video frames or pages automatically.

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x20ef	0x2	Integer	2 Bit	Read/Write	disabled	enabled	enabled	0	0	1

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	Yes	No

Language / Unit Settings

@LanguageUnits_NoOfEntries

"Number of entries in this group"

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x2200	0x0	Integer	8 Bit	Read Only	disabled	disabled	enabled	13	13	13

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	Yes	No

@SettingsLanguageEnum

"Number of language currently selected. (Set the current language with this variable)"
Enumeration is defined in the Project; directly affects @SettingsLanguageCode

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x2200	0x1	Unsigned	8 Bit	Read/Write	enabled	disabled	enabled	0	0	127

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	Yes	No

@SettingsLanguageCode

"Two byte code for language currently set Default value will match language [0] since default of '@SettingsLanguageEnum' is 0"

similar to ISO 11783-7, A.23.1, codes according to ISO 639 e.g.
"de": german
"en": english
"fr": french
"nl": dutch
As defined in the project.

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x2200	0x2	String	2 Byte	Read Only	disabled	disabled	enabled	de		

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	Yes	No

@SettingsDecimalSymbol

"Symbol used to display numbers with a fractional part"
similar to ISO 11783-7, A.23.2
0: comma
1: point
This affects all numeric fields in the project displaying decimal values with a scaling factor.

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x2200	0x3	Unsigned	2 Bit	Read/ Write	enabled	disabled	enabled	1	0	1

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	Yes	No

@SettingsDateFormat

"date format to be used"
similar to ISO 11783-7, A.23.3
0: ddmmyyyy
1: ddyyyymm
2: mmyyyydd
3: mmddyyyy
4: yyyymmdd
5: yyyyddmm

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as	Default Value	Min Value	Max Value

						Invalid			
0x2200	0x4	Unsigned	8 Bit	Read/Write	enabled	disabled	enabled	0	0	5

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	Yes	No

@SettingsTimeFormat

"time format to be used"
similar to ISO 11783-7, A.23.4
0: 24 h
1: 12 h, am/pm

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x2200	0x5	Unsigned	2 Bit	Read/Write	enabled	disabled	enabled	0	0	1

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	Yes	No

@SettingsDistanceUnits

"type of distance units to be used"
similar to ISO 11783-7, A.23.5.2
0: metric [mm, m, km]
1: imperial [in, ft, mls]

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x2200	0x6	Unsigned	2 Bit	Read/Write	enabled	disabled	enabled	0	0	1

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	Yes	No

@SettingsAreaUnits

"type of area units to be used"
similar to ISO 11783-7, A.23.5.3
0: metric [m², ha (hectares)]
1: imperial [sqare feet, acres]

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x2200	0x7	Unsigned	2 Bit	Read/Write	enabled	disabled	enabled	0	0	1

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	Yes	No

@SettingsVolumeUnits

"type of volume units to be used"
similar to ISO 11783-7, A.23.5.4
0: metric [mm³, m³, l]
1: imperial [ga (gallon)]
2: US [ga (gallon)]

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x2200	0x8	Unsigned	2 Bit	Read/Write	enabled	disabled	enabled	0	0	2

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	Yes	No

@SettingsMassUnits

"type of mass units to be used"
similar to ISO 11783-7, A.23.5.5
0: metric [g, kg, t]
1: imperial [long tonnes (UK), pounds]
2: US [short tonnes (US), pounds]

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x2200	0x9	Unsigned	2 Bit	Read/Write	enabled	disabled	enabled	0	0	2

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	Yes	No

@SettingsTemperatureUnits

"type of temperature units to be used"
similar to ISO 11783-7, A.23.5.6
0: metric [°K, °C]
1: imperial [°F]

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x2200	0xa	Unsigned	2 Bit	Read/Write	enabled	disabled	enabled	0	0	1

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	Yes	No

@SettingsPressureUnits

"type of pressure units to be used"
similar to ISO 11783-7, A.23.5.7
0: metric [hP, kP, P (Pascal)]
1: imperial [PSI]

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x2200	0xb	Unsigned	2 Bit	Read/Write	enabled	disabled	enabled	0	0	1

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	Yes	No

@SettingsForceUnits

"type of force units to be used"
similar to ISO 11783-7, A.23.5.8
0: metric [kN, N (Newton)]
1: imperial [pounds-force]

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x2200	0xc	Unsigned	2 Bit	Read/Write	enabled	disabled	enabled	0	0	1

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	Yes	No

@SettingsUnitSystem

"general unit system setting"
similar to ISO 11783-7, A.23.5.9
0: metric
1: imperial
2: US

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x2200	0xd	Unsigned	2 Bit	Read/Write	enabled	disabled	enabled	0	0	2

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	Yes	No

System Settings

@SystemSettings_NoOfEntries

"Number of entries in this group"

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x2240	0x0	Unsigned	8 Bit	Read Only	disabled	disabled	enabled	15	15	15

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	Yes	Yes

@EnableUpdater

"Enable/Disable the ClientUpdater for e.g. updating the project on the device."
Bitwise coding:

Bit 0: USB update
Bit 1: CAN update
Bit 2: ETH update

Example: 7 enables all three ways to update. 5 enables USB and ETH. 2 enables CAN only. 0 disables all update possibilities.
After setting this variable the interfaces are immediately enabled. No restart is required.

Index	Subindex	Storage	Length	Access	Remanent	Force Writing	Initialize as	Default	Min Value	Max Value

		Type		Type		g	Invalid	Value		
0x2240	0x1	Unsigned	8 Bit	Read/Write	enabled	enabled	disabled	1	0	255

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	Yes	Yes

@UpdaterCANPort

Sets the CAN Port for updates of the ClientUpdater.
0: CAN Port 1 is used
1: CAN Port 2 is used.

WARNING:
This setting is only temporary. After a restart of the ClientUpdater or a device restart the settings in the loaded project will be restored.
When a new CAN Port is set, the Clientupdater will change to the new Port directly. Please check also that the baudrate of the new CAN Port is configured properly.

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x2240	0x2	Unsigned	8 Bit	Read/Write	disabled	disabled	disabled	1	0	255

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	Yes	Yes

@EncoderDirection

Change navigation direction of encoder.
0: Select next: turn encoder counter-clockwise
1: Select next: turn encoder clockwise

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x2240	0x3	Unsigned	8 Bit	Read/Write	disabled	disabled	disabled	0	0	1

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	Yes	Yes

Projekt

@Projekt_NoOfEntries

"Number of entries in this group"

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x2400	0x0	Integer	8 Bit	Read Only	disabled	disabled	enabled	15	15	15

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	Yes	Yes

@ProjektVersion

"Projekt version defined by brand owner."
User defined string set in project to identify the version of the project that is currently loaded.

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x2400	0x1	String	255 Byte	Read Only	disabled	disabled	enabled	NO VALUE		

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	Yes	Yes

@ProjektReleaseDate

"Projekt release date defined by brand owner."
User defined string set in the project properties to identify the release date of the project currently loaded.

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x2400	0x2	String	255 Byte	Read Only	disabled	disabled	enabled	NO VALUE		

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	Yes	Yes

@ProjektCurrentPage

"get/set number of page currently visible"
Page number as defined in project.
Additionally the following numbers can be set:
0: Home Page
-1: Jump one page back

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x2400	0x3	Integer	32 Bit	Read/ Write	disabled	disabled	enabled	-1	-1	2147483647

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	Yes	Yes

@ProjektHistoryPage01

"get page that was current page before the one listed in @ProjektCurrentPage"

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x2400	0x4	Integer	32 Bit	Read Only	disabled	disabled	enabled	-1	-1	2147483647

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	Yes	Yes

@ProjektHistoryPage02

"get page that was current page before the one listed in @ProjektHistoryPage01"

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x2400	0x5	Integer	32 Bit	Read Only	disabled	disabled	enabled	-1	-1	2147483647

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	Yes	Yes

@ProjektHistoryPage03

"get page that was current page before the one listed in @ProjektHistoryPage02"

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x2400	0x6	Integer	32 Bit	Read Only	disabled	disabled	enabled	-1	-1	2147483647

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	Yes	Yes

@ProjektHistoryPage04

"get page that was current page before the one listed in @ProjektHistoryPage03"

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x2400	0x7	Integer	32 Bit	Read Only	disabled	disabled	enabled	-1	-1	2147483647

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	Yes	Yes

@ProjektHistoryPage05

"get page that was current page before the one listed in @ProjektHistoryPage04"

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x2400	0x8	Integer	32 Bit	Read Only	disabled	disabled	enabled	-1	-1	2147483647

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	Yes	Yes

@ProjektSkinSelected

"Skin currently active on screen"
This is currently not supported!

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value

| 0x2400 | 0xb | Integer | 8 Bit | Read/Write | enabled | disabled | enabled | 0 | -1 | 127 |

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	Yes	Yes

@ProjektTimePage

"Time elapsed since PClient switched to current page"
Time in ms since this page was initialized to be shown.
Could be used to control some animation.

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x2400	0xc	Integer	32 Bit	Read Only	disabled	disabled	enabled	0	-1	2147483647

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	Yes	Yes

@ProjektTakeScreenshot

"If this varaible is set to 1, a screenshot will be generated. Will automatically set back to 0."
The resulting screenshots will be saved on the device in the folder /opt/screenshots/
Reading this variable does not make sense.

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x2400	0xe	Unsigned	2 Bit	Read/Write	disabled	disabled	enabled	1	0	1

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	Yes	Yes

@PClientVersion

"Version of the PClient"

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x2400	0xf	String	255 Byte	Read Only	disabled	disabled	enabled	NO VALU		

								E		

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	Yes	Yes

@PClientSimulationMode

"Enable or disable PClient simulation mode"
Enable or disable PClient simulation mode (when enabled, variable simulation can connect to PClient via Ethernet), see also @PClientSimulationPort

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x2400	0x10	Unsigned	2 Bit	Read/ Write	disabled	disabled	disabled	0	0	1

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	Yes	Yes

@PClientSimulationPort

"The TCP/IP port that shall be used for simulation applications."
The IP address for the TCP/IP connection is taken from @Ethernet0_IPAddress

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x2400	0x11	Unsigned	16 Bit	Read/ Write	enabled	disabled	disabled	1337	256	9999

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	Yes	Yes

@ChangeSoftkeyLevel

"Use this variable to trigger the change of the softkey level on the current page. See detailed description for allowed values."
The following values are accepted:
100: Right softkey frame level down
101: Right softkey frame level up
200: Left softkey frame level down
201: Left softkey frame level up
300: Top softkey frame level down
301: Top softkey frame level up
400: Bottom softkey frame level down

401: Bottom softkey frame level up
The value will be reverted back to 0 directly after writing.

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x2400	0x12	Unsigned	8 Bit	Read/Write	disabled	enabled	disabled	0	0	401

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	Yes	Yes

@ProjektTakeScreenshotName

Provide a path and name for a screenshot that is taken with @ProjektTakeScreenshot
With this variable a path and name for a screenshot can be defined. This string is then used when @ProjektTakeScreenshot is set to 1 to take the screenshot.
The value that is set to this variable always needs to contain a path that ends with a file name. The file extension should be omitted, it is automatically created when taking the screenshot (.png).
The string is checked as soon as @ProjektTakeScreenshot is set to 1. If the string does not contain a valid path AND file name, the standard path and name are used to take the screenshot.
If a screenshot with a given name already exists, a number will be appended automatically.
A valid value for this variable is e.g. "/disk/usbsda1/myPath/myFileName" (it is important that it does not end with a /).
An invalid value is e.g. "myName" (it contains no path) or "/opt/" (it contains no name).

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x2400	0x13	String	255 Byte	Read/Write	disabled	disabled	disabled			

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	Yes	Yes

Variable Logging

@VariableLogging_NoOfEntries

"Number of entries in this group"

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value

0x2410	0x0	Integer	8 Bit	Read Only	disabled	disabled	enabled	12	12	12

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	Yes	Yes

@LogLevel

"Select Level of data currently logged."
Set loglevel so that data is logged according to this level defined in project.
0: Logging switched off

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x2410	0x1	Unsigned	8 Bit	Read/Write	enabled	disabled	enabled	0	0	10

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	Yes	Yes

@LogPath

"Path to folder where logging files are stored."

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x2410	0x2	String	255 Byte	Read/Write	enabled	disabled	enabled	/opt/pclient/logfiles		

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	Yes	Yes

@LogCopyPath

"Path where folder with logging files shall be copied/moved"
This path is intended to point to a removable media or remote computer where data is copied on demand e.g. at the end of each day

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value

| 0x2410 | 0x4 | String | 255 Byte | Read/ Write | enabled | disabled | enabled | /disk/usbsda1 | | |

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	Yes	Yes

@LogCopyNow

"Trigger copying / moving current log file and start a new file."
0: Do nothing
1: Copy log file from @LogPath to @LogCopyPath
2: Move log file from @LogPath to @LogCopyPath
Value is reset to 0 when copying/moving is finished

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x2410	0x5	Unsigned	8 Bit	Read/ Write	disabled	disabled	enabled	0	0	2

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	Yes	Yes

@LogCopyProgress

"Show progress of copying / moving log data."
Progress from 0 ... 100 percent.

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x2410	0x6	Unsigned	8 Bit	Read Only	disabled	disabled	enabled	0	0	100

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	Yes	Yes

@LogCopyAllNow

"Trigger copying / moving of all files inside @LogPath and start a new file."
0: Do nothing
1: Copy all log files from @LogPath to @LogCopyPath
2: Move all log files from @LogPath to @LogCopyPath
Value is reset to 0 when copying/moving is finished

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x2410	0x7	Unsigned	8 Bit	Read/Write	disabled	disabled	enabled	0	0	2

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	Yes	Yes

@LogCurrentLogFileName

"Name of the current log file without the leading path"
The name of the log file as configured in the program. Contained tags are replaced (see variable logging documentation)

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x2410	0x8	String	255 Byte	Read Only	disabled	disabled	enabled	NO VALUE		

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	Yes	Yes

@LogCurrentLogFileSize

"The current file size of the log file in kilobytes"

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x2410	0x9	Unsigned	32 Bit	Read Only	disabled	disabled	enabled	0	0	2147483647

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	Yes	Yes

@LogMaximumLogFileSize

"The maximum log file size in kilobytes"

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as	Default Value	Min Value	Max Value

						Invalid			
0x2410	0xa	Integer	32 Bit	Read/Write	enabled	disabled	enabled	-1	-1	10000

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	Yes	Yes

@LogCopyError

"Error Number 0 = OK"
If the copy / move feature fails in some way, you can see the error code here.

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x2410	0xb	Unsigned	32 Bit	Read/Write	enabled	disabled	enabled	0	0	2147483647

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	Yes	Yes

@LogCopyFinished

"0 = in Progress 1 = finished"
If you started the copy / move process you can determine if the process is finished. If the process is finished and @LogCopyProgress < 100 you can read out @LogCopyError

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x2410	0xc	Unsigned	1 Bit	Read/Write	enabled	disabled	enabled	0	0	1

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	Yes	Yes

Visibility

@Visibility_NoOfEntries

"Number of entries in this group"

Index	Subindex	Storage	Length	Access	Remanent	Force Writing	Initialize as	Default	Min Value	Max Value

		Type	Type		g	Invalid	Value		
0x2420	0x0	Integer	8 Bit	Read Only	disabled	disabled	enabled	66	66	66

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	Yes	Yes

@VisibilityStore

"Store visibility of one or all visibility variable(s) remanently and overwrite the initial setting"
-1: do nothing
XX == 00 ... 63: store @VisibilityXX remanently.
0xa5a5: All visibility variables are stored.
When storing is finished, value is reset to -1.
Reading this variable will always return -1

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x2420	0x1	Integer	16 Bit	Read/Write	disabled	disabled	enabled	-1	-1	42405

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	Yes	Yes

@VisibilityReadRemanent

"Reload visibility of one or all visibility variable(s) from remanently stored value and overwrite the current setting"
-1: do nothing
XX == 00 ... 63: overwrite @VisibilityXX with remanently stored value.
0xa5a5: All visibility variables are restored from remanently stored values.
When overwriting is finished, value is reset to -1.
Reading will always return -1

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x2420	0x2	Integer	16 Bit	Read/Write	disabled	disabled	enabled	-1	-1	42405

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	Yes	Yes

@Visibility00

"Represents Visibility group 00"
Where a bit is set (1) the objects assigned to it are visible.
Default is "all objects visible"

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x2420	0x3	Unsigned	16 Bit	Read/Write	disabled	disabled	enabled	65535	0	65535

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	Yes	Yes

@Visibility01

"Represents Visibility group 01"
Where a bit is set (1) the objects assigned to it are visible.
Default is "all objects visible"

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x2420	0x4	Unsigned	16 Bit	Read/Write	disabled	disabled	enabled	65535	0	65535

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	Yes	Yes

@Visibility02

"Represents Visibility group 02"
Where a bit is set (1) the objects assigned to it are visible.
Default is "all objects visible"

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x2420	0x5	Unsigned	16 Bit	Read/Write	disabled	disabled	enabled	65535	0	65535

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	Yes	Yes

@Visibility03

"Represents Visibility group 03"
Where a bit is set (1) the objects assigned to it are visible.
Default is "all objects visible"

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x2420	0x6	Unsigned	16 Bit	Read/Write	disabled	disabled	enabled	65535	0	65535

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	Yes	Yes

@Visibility04

"Represents Visibility group 04"
Where a bit is set (1) the objects assigned to it are visible.
Default is "all objects visible"

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x2420	0x7	Unsigned	16 Bit	Read/Write	disabled	disabled	enabled	65535	0	65535

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	Yes	Yes

@Visibility05

"Represents Visibility group 05"
Where a bit is set (1) the objects assigned to it are visible.
Default is "all objects visible"

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x2420	0x8	Unsigned	16 Bit	Read/Write	disabled	disabled	enabled	65535	0	65535

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	Yes	Yes

@Visibility06

"Represents Visibility group 06"
Where a bit is set (1) the objects assigned to it are visible.
Default is "all objects visible"

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x2420	0x9	Unsigned	16 Bit	Read/Write	disabled	disabled	enabled	65535	0	65535

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	Yes	Yes

@Visibility07

"Represents Visibility group 07"
Where a bit is set (1) the objects assigned to it are visible.
Default is "all objects visible"

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x2420	0xa	Unsigned	16 Bit	Read/Write	disabled	disabled	enabled	65535	0	65535

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	Yes	Yes

@Visibility08

"Represents Visibility group 08"
Where a bit is set (1) the objects assigned to it are visible.
Default is "all objects visible"

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x2420	0xb	Unsigned	16 Bit	Read/Write	disabled	disabled	enabled	65535	0	65535

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	Yes	Yes

@Visibility09

"Represents Visibility group 09"
Where a bit is set (1) the objects assigned to it are visible.
Default is "all objects visible"

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x2420	0xc	Unsigned	16 Bit	Read/Write	disabled	disabled	enabled	65535	0	65535

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	Yes	Yes

@Visibility10

"Represents Visibility group 10"
Where a bit is set (1) the objects assigned to it are visible.
Default is "all objects visible"

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x2420	0xd	Unsigned	16 Bit	Read/Write	disabled	disabled	enabled	65535	0	65535

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	Yes	Yes

@Visibility11

"Represents Visibility group 11"
Where a bit is set (1) the objects assigned to it are visible.
Default is "all objects visible"

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x2420	0xe	Unsigned	16 Bit	Read/Write	disabled	disabled	enabled	65535	0	65535

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	Yes	Yes

@Visibility12

"Represents Visibility group 12"
Where a bit is set (1) the objects assigned to it are visible.
Default is "all objects visible"

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x2420	0xf	Unsigned	16 Bit	Read/ Write	disabled	disabled	enabled	65535	0	65535

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	Yes	Yes

@Visibility13

"Represents Visibility group 13"
Where a bit is set (1) the objects assigned to it are visible.
Default is "all objects visible"

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x2420	0x10	Unsigned	16 Bit	Read/ Write	disabled	disabled	enabled	65535	0	65535

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	Yes	Yes

@Visibility14

"Represents Visibility group 14"
Where a bit is set (1) the objects assigned to it are visible.
Default is "all objects visible"

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x2420	0x11	Unsigned	16 Bit	Read/ Write	disabled	disabled	enabled	65535	0	65535

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	Yes	Yes

@Visibility15

"Represents Visibility group 15"
Where a bit is set (1) the objects assigned to it are visible.
Default is "all objects visible"

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x2420	0x12	Unsigned	16 Bit	Read/ Write	disabled	disabled	enabled	65535	0	65535

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	Yes	Yes

@Visibility16

"Represents Visibility group 16"
Where a bit is set (1) the objects assigned to it are visible.
Default is "all objects visible"

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x2420	0x13	Unsigned	16 Bit	Read/ Write	disabled	disabled	enabled	65535	0	65535

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	Yes	Yes

@Visibility17

"Represents Visibility group 17"
Where a bit is set (1) the objects assigned to it are visible.
Default is "all objects visible"

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x2420	0x14	Unsigned	16 Bit	Read/ Write	disabled	disabled	enabled	65535	0	65535

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	Yes	Yes

@Visibility18

"Represents Visibility group 18"
Where a bit is set (1) the objects assigned to it are visible.
Default is "all objects visible"

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x2420	0x15	Unsigned	16 Bit	Read/Write	disabled	disabled	enabled	65535	0	65535

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	Yes	Yes

@Visibility19

"Represents Visibility group 19"
Where a bit is set (1) the objects assigned to it are visible.
Default is "all objects visible"

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x2420	0x16	Unsigned	16 Bit	Read/Write	disabled	disabled	enabled	65535	0	65535

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	Yes	Yes

@Visibility20

"Represents Visibility group 20"
Where a bit is set (1) the objects assigned to it are visible.
Default is "all objects visible"

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x2420	0x17	Unsigned	16 Bit	Read/Write	disabled	disabled	enabled	65535	0	65535

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	Yes	Yes

@Visibility21

"Represents Visibility group 21"
Where a bit is set (1) the objects assigned to it are visible.
Default is "all objects visible"

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x2420	0x18	Unsigned	16 Bit	Read/Write	disabled	disabled	enabled	65535	0	65535

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	Yes	Yes

@Visibility22

"Represents Visibility group 22"
Where a bit is set (1) the objects assigned to it are visible.
Default is "all objects visible"

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x2420	0x19	Unsigned	16 Bit	Read/Write	disabled	disabled	enabled	65535	0	65535

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	Yes	Yes

@Visibility23

"Represents Visibility group 23"
Where a bit is set (1) the objects assigned to it are visible.
Default is "all objects visible"

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x2420	0x1a	Unsigned	16 Bit	Read/Write	disabled	disabled	enabled	65535	0	65535

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	Yes	Yes

@Visibility24

"Represents Visibility group 24"
Where a bit is set (1) the objects assigned to it are visible.
Default is "all objects visible"

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x2420	0x1b	Unsigned	16 Bit	Read/Write	disabled	disabled	enabled	65535	0	65535

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	Yes	Yes

@Visibility25

"Represents Visibility group 25"
Where a bit is set (1) the objects assigned to it are visible.
Default is "all objects visible"

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x2420	0x1c	Unsigned	16 Bit	Read/Write	disabled	disabled	enabled	65535	0	65535

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	Yes	Yes

@Visibility26

"Represents Visibility group 26"
Where a bit is set (1) the objects assigned to it are visible.
Default is "all objects visible"

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x2420	0x1d	Unsigned	16 Bit	Read/Write	disabled	disabled	enabled	65535	0	65535

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	Yes	Yes

@Visibility27

"Represents Visibility group 27"
Where a bit is set (1) the objects assigned to it are visible.
Default is "all objects visible"

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x2420	0x1e	Unsigned	16 Bit	Read/Write	disabled	disabled	enabled	65535	0	65535

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	Yes	Yes

@Visibility28

"Represents Visibility group 28"
Where a bit is set (1) the objects assigned to it are visible.
Default is "all objects visible"

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x2420	0x1f	Unsigned	16 Bit	Read/Write	disabled	disabled	enabled	65535	0	65535

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	Yes	Yes

@Visibility29

"Represents Visibility group 29"
Where a bit is set (1) the objects assigned to it are visible.
Default is "all objects visible"

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x2420	0x20	Unsigned	16 Bit	Read/Write	disabled	disabled	enabled	65535	0	65535

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	Yes	Yes

@Visibility30

"Represents Visibility group 30"
Where a bit is set (1) the objects assigned to it are visible.
Default is "all objects visible"

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x2420	0x21	Unsigned	16 Bit	Read/Write	disabled	disabled	enabled	65535	0	65535

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	Yes	Yes

@Visibility31

"Represents Visibility group 31"
Where a bit is set (1) the objects assigned to it are visible.
Default is "all objects visible"

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x2420	0x22	Unsigned	16 Bit	Read/Write	disabled	disabled	enabled	65535	0	65535

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	Yes	Yes

@Visibility32

"Represents Visibility group 32"
Where a bit is set (1) the objects assigned to it are visible.
Default is "all objects visible"

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x2420	0x23	Unsigned	16 Bit	Read/Write	disabled	disabled	enabled	65535	0	65535

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	Yes	Yes

@Visibility33

"Represents Visibility group 33"
Where a bit is set (1) the objects assigned to it are visible.
Default is "all objects visible"

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x2420	0x24	Unsigned	16 Bit	Read/Write	disabled	disabled	enabled	65535	0	65535

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	Yes	Yes

@Visibility34

"Represents Visibility group 34"
Where a bit is set (1) the objects assigned to it are visible.
Default is "all objects visible"

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x2420	0x25	Unsigned	16 Bit	Read/Write	disabled	disabled	enabled	65535	0	65535

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	Yes	Yes

@Visibility35

"Represents Visibility group 35"
Where a bit is set (1) the objects assigned to it are visible.
Default is "all objects visible"

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x2420	0x26	Unsigned	16 Bit	Read/Write	disabled	disabled	enabled	65535	0	65535

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	Yes	Yes

@Visibility36

"Represents Visibility group 36"
Where a bit is set (1) the objects assigned to it are visible.
Default is "all objects visible"

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x2420	0x27	Unsigned	16 Bit	Read/Write	disabled	disabled	enabled	65535	0	65535

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	Yes	Yes

@Visibility37

"Represents Visibility group 37"
Where a bit is set (1) the objects assigned to it are visible.
Default is "all objects visible"

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x2420	0x28	Unsigned	16 Bit	Read/Write	disabled	disabled	enabled	65535	0	65535

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	Yes	Yes

@Visibility38

"Represents Visibility group 38"
Where a bit is set (1) the objects assigned to it are visible.
Default is "all objects visible"

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x2420	0x29	Unsigned	16 Bit	Read/Write	disabled	disabled	enabled	65535	0	65535

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	Yes	Yes

@Visibility39

"Represents Visibility group 39"
Where a bit is set (1) the objects assigned to it are visible.
Default is "all objects visible"

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x2420	0x2a	Unsigned	16 Bit	Read/Write	disabled	disabled	enabled	65535	0	65535

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	Yes	Yes

@Visibility40

"Represents Visibility group 40"
Where a bit is set (1) the objects assigned to it are visible.
Default is "all objects visible"

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x2420	0x2b	Unsigned	16 Bit	Read/Write	disabled	disabled	enabled	65535	0	65535

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	Yes	Yes

@Visibility41

"Represents Visibility group 41"
Where a bit is set (1) the objects assigned to it are visible.
Default is "all objects visible"

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x2420	0x2c	Unsigned	16 Bit	Read/Write	disabled	disabled	enabled	65535	0	65535

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	Yes	Yes

@Visibility42

"Represents Visibility group 42"
Where a bit is set (1) the objects assigned to it are visible.
Default is "all objects visible"

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x2420	0x2d	Unsigned	16 Bit	Read/Write	disabled	disabled	enabled	65535	0	65535

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	Yes	Yes

@Visibility43

"Represents Visibility group 43"
Where a bit is set (1) the objects assigned to it are visible.
Default is "all objects visible"

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x2420	0x2e	Unsigned	16 Bit	Read/Write	disabled	disabled	enabled	65535	0	65535

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	Yes	Yes

@Visibility44

"Represents Visibility group 44"
Where a bit is set (1) the objects assigned to it are visible.
Default is "all objects visible"

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x2420	0x2f	Unsigned	16 Bit	Read/Write	disabled	disabled	enabled	65535	0	65535

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	Yes	Yes

@Visibility45

"Represents Visibility group 45"
Where a bit is set (1) the objects assigned to it are visible.
Default is "all objects visible"

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x2420	0x30	Unsigned	16 Bit	Read/Write	disabled	disabled	enabled	65535	0	65535

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	Yes	Yes

@Visibility46

"Represents Visibility group 46"
Where a bit is set (1) the objects assigned to it are visible.
Default is "all objects visible"

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x2420	0x31	Unsigned	16 Bit	Read/Write	disabled	disabled	enabled	65535	0	65535

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	Yes	Yes

@Visibility47

"Represents Visibility group 47"
Where a bit is set (1) the objects assigned to it are visible.
Default is "all objects visible"

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x2420	0x32	Unsigned	16 Bit	Read/Write	disabled	disabled	enabled	65535	0	65535

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	Yes	Yes

@Visibility48

"Represents Visibility group 48"
Where a bit is set (1) the objects assigned to it are visible.
Default is "all objects visible"

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x2420	0x33	Unsigned	16 Bit	Read/Write	disabled	disabled	enabled	65535	0	65535

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	Yes	Yes

@Visibility49

"Represents Visibility group 49"
Where a bit is set (1) the objects assigned to it are visible.
Default is "all objects visible"

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x2420	0x34	Unsigned	16 Bit	Read/Write	disabled	disabled	enabled	65535	0	65535

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	Yes	Yes

@Visibility50

"Represents Visibility group 50"
Where a bit is set (1) the objects assigned to it are visible.
Default is "all objects visible"

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x2420	0x35	Unsigned	16 Bit	Read/Write	disabled	disabled	enabled	65535	0	65535

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	Yes	Yes

@Visibility51

"Represents Visibility group 51"
Where a bit is set (1) the objects assigned to it are visible.
Default is "all objects visible"

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x2420	0x36	Unsigned	16 Bit	Read/Write	disabled	disabled	enabled	65535	0	65535

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	Yes	Yes

@Visibility52

"Represents Visibility group 52"
Where a bit is set (1) the objects assigned to it are visible.
Default is "all objects visible"

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x2420	0x37	Unsigned	16 Bit	Read/Write	disabled	disabled	enabled	65535	0	65535

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	Yes	Yes

@Visibility53

"Represents Visibility group 53"
Where a bit is set (1) the objects assigned to it are visible.
Default is "all objects visible"

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x2420	0x38	Unsigned	16 Bit	Read/Write	disabled	disabled	enabled	65535	0	65535

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	Yes	Yes

@Visibility54

"Represents Visibility group 54"
Where a bit is set (1) the objects assigned to it are visible.
Default is "all objects visible"

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x2420	0x39	Unsigned	16 Bit	Read/Write	disabled	disabled	enabled	65535	0	65535

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	Yes	Yes

@Visibility55

"Represents Visibility group 55"
Where a bit is set (1) the objects assigned to it are visible.
Default is "all objects visible"

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x2420	0x3a	Unsigned	16 Bit	Read/Write	disabled	disabled	enabled	65535	0	65535

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	Yes	Yes

@Visibility56

"Represents Visibility group 56"
Where a bit is set (1) the objects assigned to it are visible.
Default is "all objects visible"

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x2420	0x3b	Unsigned	16 Bit	Read/Write	disabled	disabled	enabled	65535	0	65535

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	Yes	Yes

@Visibility57

"Represents Visibility group 57"
Where a bit is set (1) the objects assigned to it are visible.
Default is "all objects visible"

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x2420	0x3c	Unsigned	16 Bit	Read/Write	disabled	disabled	enabled	65535	0	65535

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	Yes	Yes

@Visibility58

"Represents Visibility group 58"
Where a bit is set (1) the objects assigned to it are visible.
Default is "all objects visible"

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x2420	0x3d	Unsigned	16 Bit	Read/Write	disabled	disabled	enabled	65535	0	65535

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	Yes	Yes

@Visibility59

"Represents Visibility group 59"
Where a bit is set (1) the objects assigned to it are visible.
Default is "all objects visible"

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x2420	0x3e	Unsigned	16 Bit	Read/Write	disabled	disabled	enabled	65535	0	65535

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	Yes	Yes

@Visibility60

"Represents Visibility group 60"
Where a bit is set (1) the objects assigned to it are visible.
Default is "all objects visible"

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x2420	0x3f	Unsigned	16 Bit	Read/Write	disabled	disabled	enabled	65535	0	65535

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	Yes	Yes

@Visibility61

"Represents Visibility group 61"
Where a bit is set (1) the objects assigned to it are visible.
Default is "all objects visible"

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x2420	0x40	Unsigned	16 Bit	Read/Write	disabled	disabled	enabled	65535	0	65535

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	Yes	Yes

@Visibility62

"Represents Visibility group 62"
Where a bit is set (1) the objects assigned to it are visible.
Default is "all objects visible"

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x2420	0x41	Unsigned	16 Bit	Read/Write	disabled	disabled	enabled	65535	0	65535

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	Yes	Yes

@Visibility63

"Represents Visibility group 63"
Where a bit is set (1) the objects assigned to it are visible.
Default is "all objects visible"

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x2420	0x42	Unsigned	16 Bit	Read/Write	disabled	disabled	enabled	65535	0	65535

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	Yes	Yes

Alarms

@Alarms_NoOfEntries

"Number of entries in this group"

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x2430	0x0	Integer	8 Bit	Read Only	disabled	disabled	enabled	5	5	5

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	Yes	Yes

@AlarmShow

"Alarm to be raised or removed. This is a queued variable, so reading it is not sensible."
Alarm number: Remove this alarm.
0x8000 & Alarm number: Engage alarm.
E.g. engage alarm 10: 0x800a ==32778;
remove alarm 10: 0x000a == 10
Engaging an undefined alarm will show a default informational alarm identifying the number (removed by ACK)

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x2430	0x1	Unsigned	16 Bit	Read/Write	disabled	disabled	enabled	0	0	65535

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	Yes	Yes

@AlarmCurrent

"Alarm currently shown on screen"
Alarm number of alarm currently shown on screen.
Additionally bit 15 is set when the alarm is acknowledged by the user.
To be requested by ECRR to receive user ACK.

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x2430	0x2	Unsigned	16 Bit	Read Only	disabled	disabled	enabled	0	0	36863

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	Yes	Yes

@IsAlarmEnqueued

"Ask if alarm number is stored in alarm queue"
Write alarm number to this variable to request if it is stored in queue.
Response has to be read from @AlarmEnqueued

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x2430	0x3	Unsigned	16 Bit	Read/ Write	disabled	disabled	enabled	0	0	32767

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	Yes	Yes

@AlarmEnqueued

"Response to writing @IsAlarmEnqueued"
Alarm number requested by writing to @IsAlarmEnqueued, bit 15 NOT set if it is enqueued.
E.g. writing 10 to @IsAlarmQueued may set this variable to 0x800a (is not enqueued) or 0x000a (is enqueued)

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value

0x2430	0x4	Unsigned	16 Bit	Read Only	disabled	disabled	enabled	0	0	36863

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	Yes	Yes

@AlarmCurrentPriority

"Priority of the currently visible alarm"
Alarm properties are as followed (n is the number of defined alarm priorities, maximum is 9):
0: Emergency Alarm, high priority
1: Medium priority
2: Low priority
... (optionaly user defined priorities)
n-1: Warning Alarm
n: Informational Alarm

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x2430	0x5	Unsigned	8 Bit	Read Only	disabled	disabled	enabled	0	0	32

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	Yes	Yes

Virtual Keyboard

@VKB_NoOfEntries

"Number of entries in this group"

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x2440	0x0	Integer	8 Bit	Read Only	disabled	disabled	enabled	6	6	6

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	Yes	Yes

@VKB_EditedValue

"Value being edited by the VirtualKeyboard"

This variable stores the Value being edited by the Virtual Keyboard. This is always a string value and appropriate casting is required for the Numeric DDO.

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x2440	0x1	String	255 Byte	Read/ Write	disabled	disabled	enabled	NO VALUE		

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	Yes	Yes

@VKB_KeyPressed

"Value of the Last Key pressed on the Virtual Keyboard"
Stores the Last key that was pressed by the user in then VKB. This variable is updated on the Key press event of the VKBButtonDDO

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x2440	0x2	String	255 Byte	Read/ Write	disabled	disabled	enabled	NO VALUE		

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	Yes	Yes

@VKB_Current

"Virtual Keyboard currently being Displayed"
VKB DDO to be displayed.

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x2440	0x3	Integer	32 Bit	Read/ Write	disabled	disabled	enabled	-1	-1	2147483647

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	Yes	Yes

@VKB_CurrentLayer

"Layer of the Active/Currently displayed VKB to be displayed"

ID of the Layer(Container Object) to be displayed for the current VKB.

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x2440	0x4	Unsigned	32 Bit	Read/Write	disabled	disabled	enabled	0	0	2147483647

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	Yes	Yes

@VKB_OkCancel

"Indicates whether the OK or Cancel button was pressed"
This variable is updated on the Key Release event of the VKBButtonDDO.

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x2440	0x5	Unsigned	8 Bit	Read/Write	disabled	disabled	enabled	0	0	3

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	Yes	Yes

@VKB_EditedDDO

"ID of the DDO being edited by the VKB."
This variable is updated on the Open For Input Event of the DDO, i.e when the VKB is to be displayed for editing the ddo, with the ID of the DDO.

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x2440	0x6	Integer	32 Bit	Read/Write	disabled	disabled	enabled	-1	-1	2147483647

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	Yes	Yes

J1939

@J1939_NoOfEntries

"Number of entries in this group"

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x2450	0x0	Integer	8 Bit	Read Only	disabled	disabled	enabled	8	8	8

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	Yes	Yes

@MalfunctionIndicatorLampStatus

"Combined status and flash information of the DM1 Malfunction Indicator Lamp (calculated value from all ECUs)"
See online manual J1939 => DM1/DM2 help page

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x2450	0x1	Integer	8 Bit	Read Only	disabled	disabled	enabled	0	0	15

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	Yes	Yes

@RedStopLampStatus

"Combined status and flash information of the DM1 Red Stop Lamp (calculated value from all ECUs)"
See online manual J1939 => DM1/DM2 help page

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x2450	0x2	Integer	8 Bit	Read Only	disabled	disabled	enabled	0	0	15

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	Yes	Yes

@AmberWarningLampStatus

"Combined status and flash information of the DM1 Amber Warning Lamp (calculated value from all ECUs)"
See online manual J1939 => DM1/DM2 help page

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x2450	0x3	Integer	8 Bit	Read Only	disabled	disabled	enabled	0	0	15

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	Yes	Yes

@ProtectLampStatus

"Combined status and flash information of the DM1 Protect Lamp (calculated value from all ECUs)"
See online manual J1939 => DM1/DM2 help page

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x2450	0x4	Integer	8 Bit	Read Only	disabled	disabled	enabled	0	0	15

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	Yes	Yes

@MalfunctionIndicatorLampFlashing

"Calculated value from all flashing and status information for the Malfunction Indicator lamp. The variable is either 0 or or will automatically flash slowly or fast."
See online manual J1939 => DM1/DM2 help page

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x2450	0x5	Integer	8 Bit	Read Only	disabled	disabled	enabled	0	0	1

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	Yes	Yes

@RedStopLampFlashing

"Calculated value from all flashing and status information for the Red Stop lamp. The variable is either 0 or or will automatically flash slowly or fast."
See online manual J1939 => DM1/DM2 help page

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x2450	0x6	Integer	8 Bit	Read Only	disabled	disabled	enabled	0	0	1

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	Yes	Yes

@AmberWarningLampFlashing

"Calculated value from all flashing and status information for the Amber Warning lamp. The variable is either 0 or or will automatically flash slowly or fast."
See online manual J1939 => DM1/DM2 help page

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x2450	0x7	Integer	8 Bit	Read Only	disabled	disabled	enabled	0	0	1

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	Yes	Yes

@ProtectLampFlashing

"Calculated value from all flashing and status information for the Protect lamp. The variable is either 0 or or will automatically flash slowly or fast."
See online manual J1939 => DM1/DM2 help page

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x2450	0x8	Integer	8 Bit	Read Only	disabled	disabled	enabled	0	0	1

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	Yes	Yes

Theme Settings

@Theme_NoOfEntries

"Number of entries in this group"

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x2460	0x0	Unsigned	8 Bit	Read Only	disabled	disabled	enabled	1	0	1

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	Yes	Yes

@CurrentThemeIndex

"Index of current shown theme. E.g.: index 0 => Day Mode / index 1 => Night Mode"
Every DDO has its own definition about what theme or dedicated settings are shown for the respective index value.
Currently PC tool only provides fixed settings for 0 == day and 1 == night. Basically the mechanism could support more different values though.

Index	Subindex	Storage Type	Length	Access Type	Remanent	Force Writing	Initialize as Invalid	Default Value	Min Value	Max Value
0x2460	0x1	Unsigned	8 Bit	Read/Write	enabled	disabled	disabled	0	0	127

Supported on Platforms:

i.MX 35	i.MX 6 A-Series	i.MX 6 B-Series	ISOBUS
Yes	Yes	Yes	Yes

CAN

5   CAN

The communication between the terminal and other devices is realized with the help of CAN protocols. Each kind of protocol supported has its special properties to be configured according to the requirements.

Supported are 2 prevalent and 1 proprietary CAN protocol:
- SAE J1939[342]
- CANopen[328]
- CANFreestyle[308] (proprietary)

 Tutorial video about setting up the basic CAN settings
https://www.youtube.com/watch?v=DO0tPCmaB1w

Firstly, you always need to set up the bit rate that you want to use. Open the menu Communication -> Port Configuration...
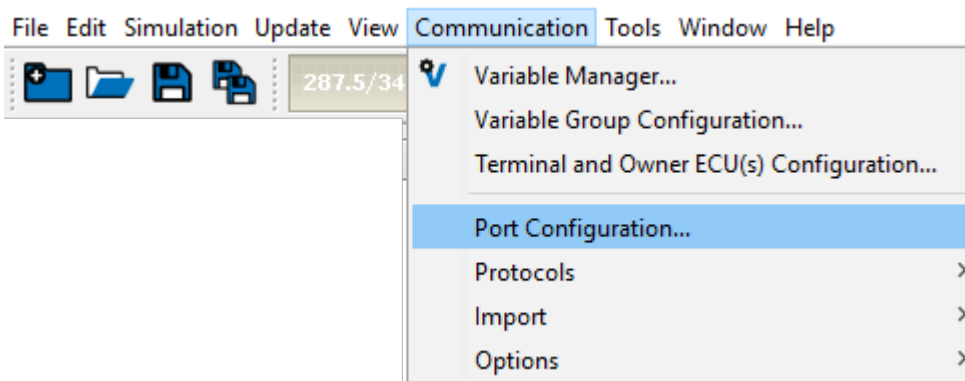
In the dialog, choose the CAN port you want to use (1) and then the bit rate / baud rate / bus speed that you want to use in your project for that port (2).
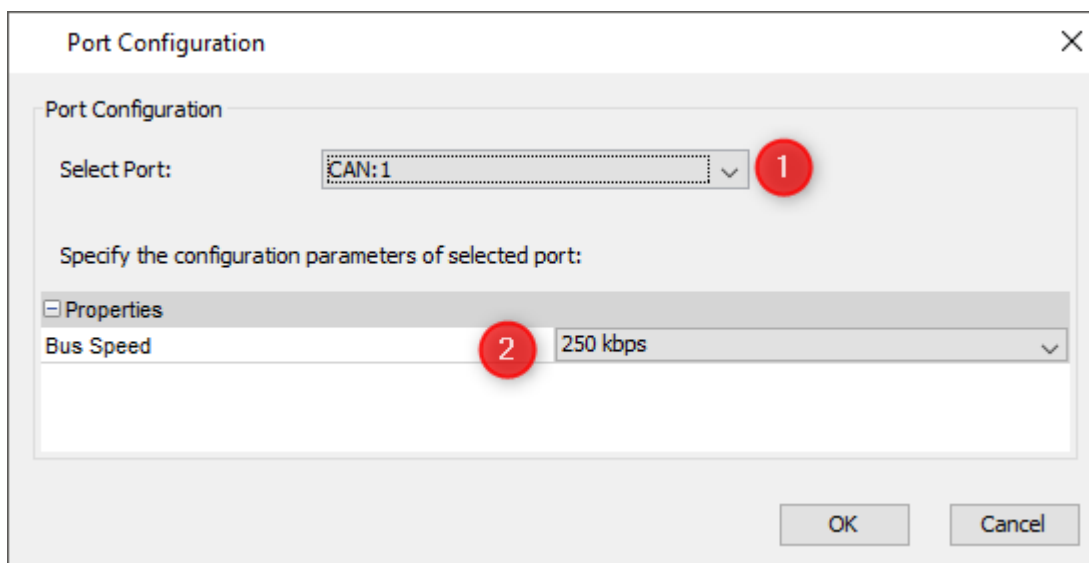
5.1 CANFreestyle

General Introduction

CANFreestyle is a simple proprietary protocol, without e.g. the complexity of a transport protocol. So this protocol is very easy to use.
Anyway you can modify CAN values that you want to send on the bus as well as the CAN values that you received from the CAN bus. The modifiers are the same as for J1939.
Actually CANFreestyle provides a way to emulate other CAN protocols that are not supported natively. To prevent confusion, CANFreestyle will be treated as a protocol in the manual.
Additionally you can create a sequence of several CAN messages which can be sent periodically.

Adding CANFreestyle to the GUI Definition

Before you can use the CANFreestyle protocol, it has to be activated. Before activating it the **Port Configuration** has to be set.

Port Configuration

To configure the port that shall be used for the communication select main menu tab

[Communication][Port Configuration...] 1 . A new dialog 'Port Configuration' will

be opened. Depending on the **Selected Port** 2 the **Bus Speed** 3 should be set.

Press "OK" 4 to confirm the settings.
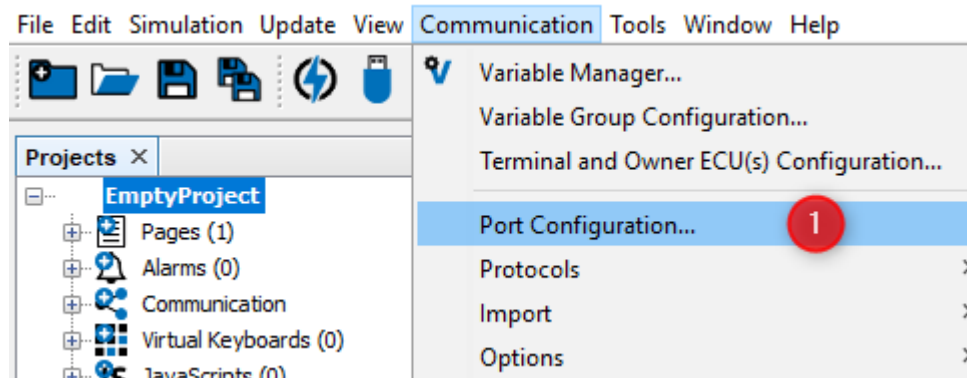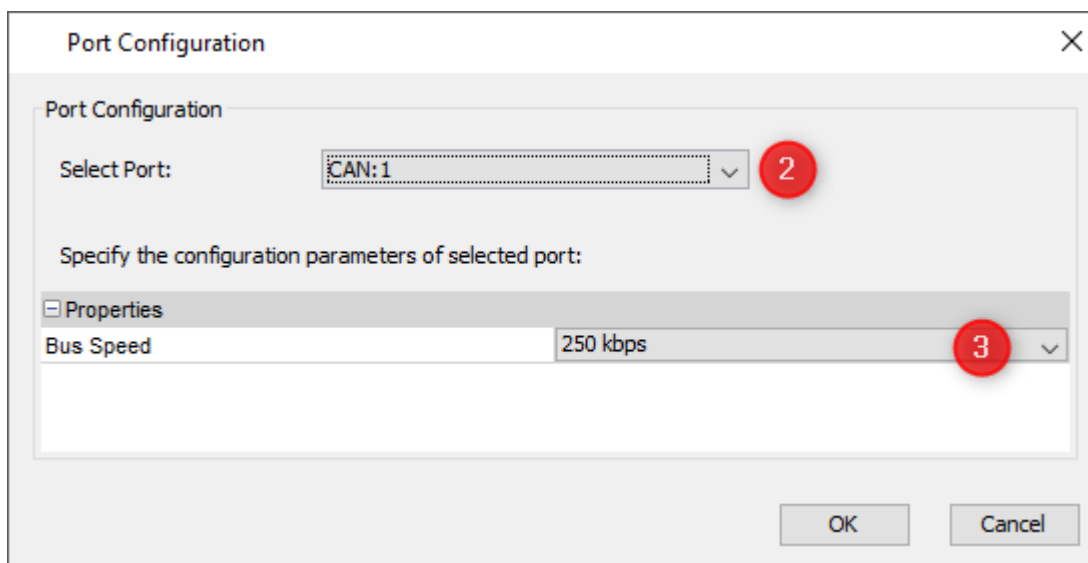
ECU Configuration

Once the port is configured the protocol itself has to be assigned to it.
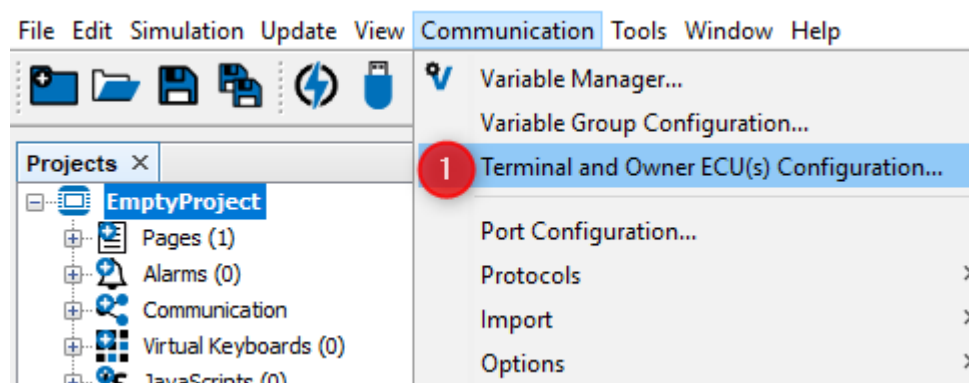
1. Open "Communication" menu and choose "Terminal and Owner ECU(s) Configuration ...". The "Terminal and Owner ECU(s) Configuration" dialog will appear
2. Press on the arrow and select **CANFreestyle** protocol for the CAN port you want it to run on
3. Press "Add" to use this protocol with your GUI on the display

⊞**Properties of Terminal ECU**
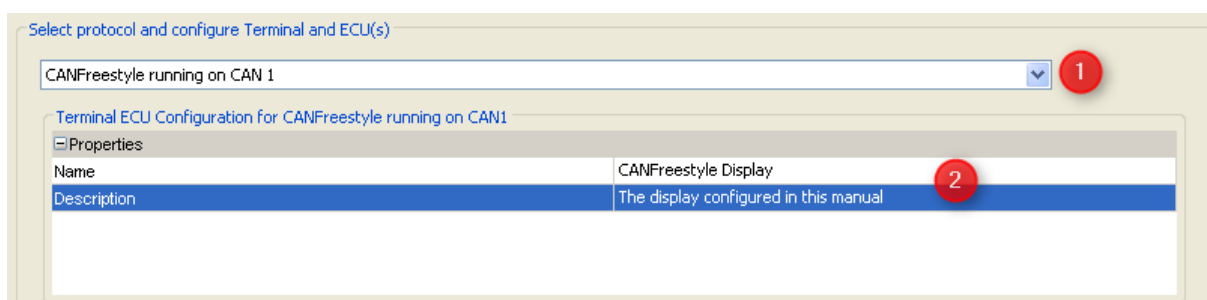
1. Select CANFreestyle on the CAN port configured before - in our example we configured it for CAN 1.
2. Type a valid name and description - these are mainly for your convenience and may help others to understand the GUI configuration in future.

⊞**Properties of Owner ECU(s)**

ECUs are configured in the same "Terminal and Owner ECU(s) Configuration" dialog that was used above to activate CANFreestyle protocol and to configure the display's address. ECUs are configured in the lower part of this dialog:

1. Type a name describing the ECU
2. Press "Add"

Then the new ECU will automatically be selected and one can modify the other parameters:

3. Provide a description that could help others to understand better which ECU is meant here
4. In case the ECU is not needed anymore it can be deleted again. If it already owns any variables, the owner will be changed to the display (PClient) when you delete it.

Adding Variables to the CANFreestyle ECUs

As mentioned above we mainly configured the CANFreestyle ECUs because they own some variables whose values we want to show and/or modify on the display.
Before we can use these variables we have to do two things:
1. Create the variables and use one of the CANFreestyle ECUs as owner.
2. Define the Mappings we want to use to send and receive the variables.

⊞**Creating dedicated variable groups for variables owned by CANFreestyle ECUs**

Knowing that we could end up with a lot of variables per ECU it's a good idea to create one variable group per ECU. Of course you are free to organize variables in groups however you prefer.
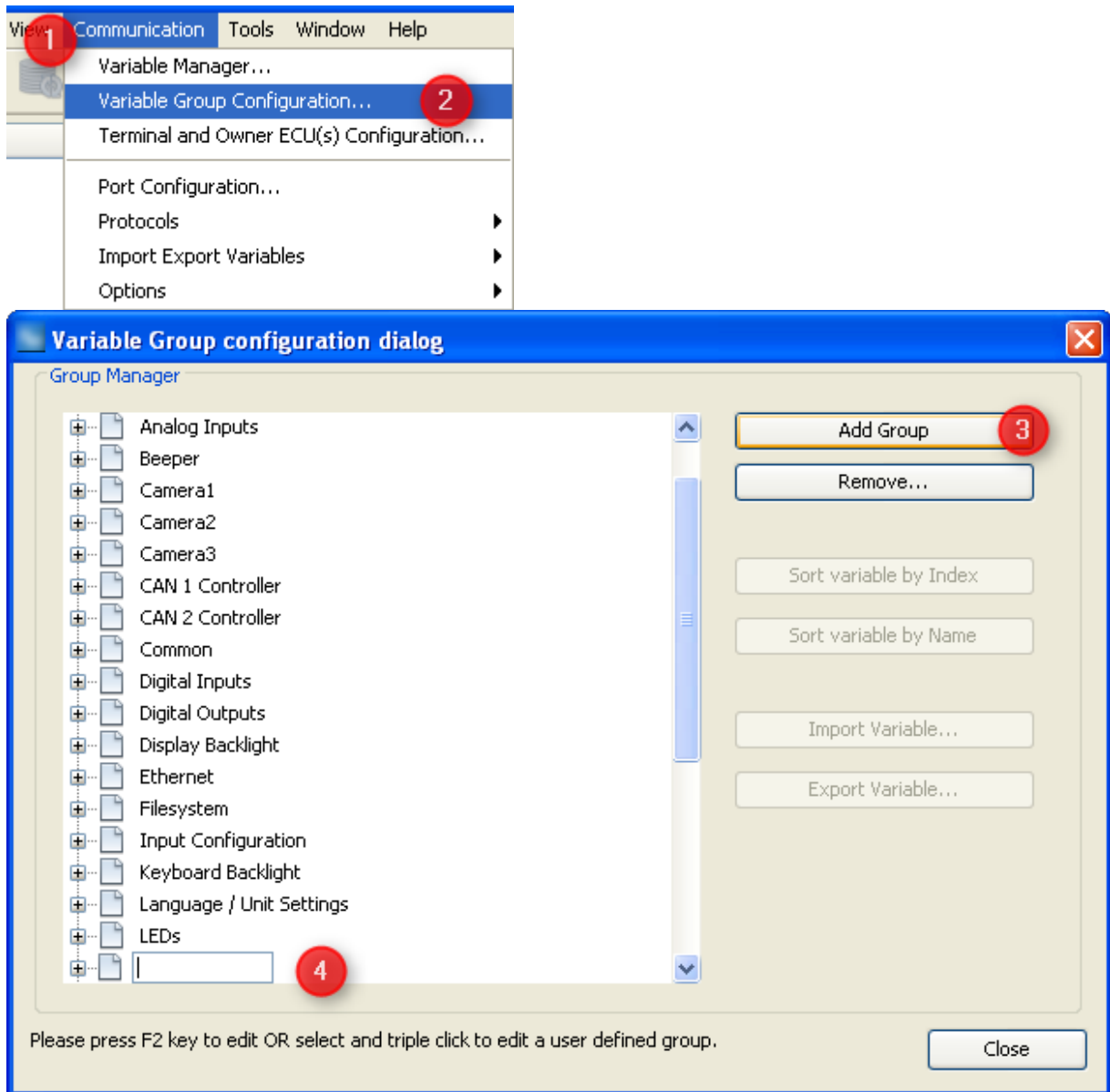
1. Open Menu "Communication"
2. Select "Variable Group Configuration ..."
3. In the dialog that opens press "Add Group" to create a new variable group
4. Type a sensible name for the new group
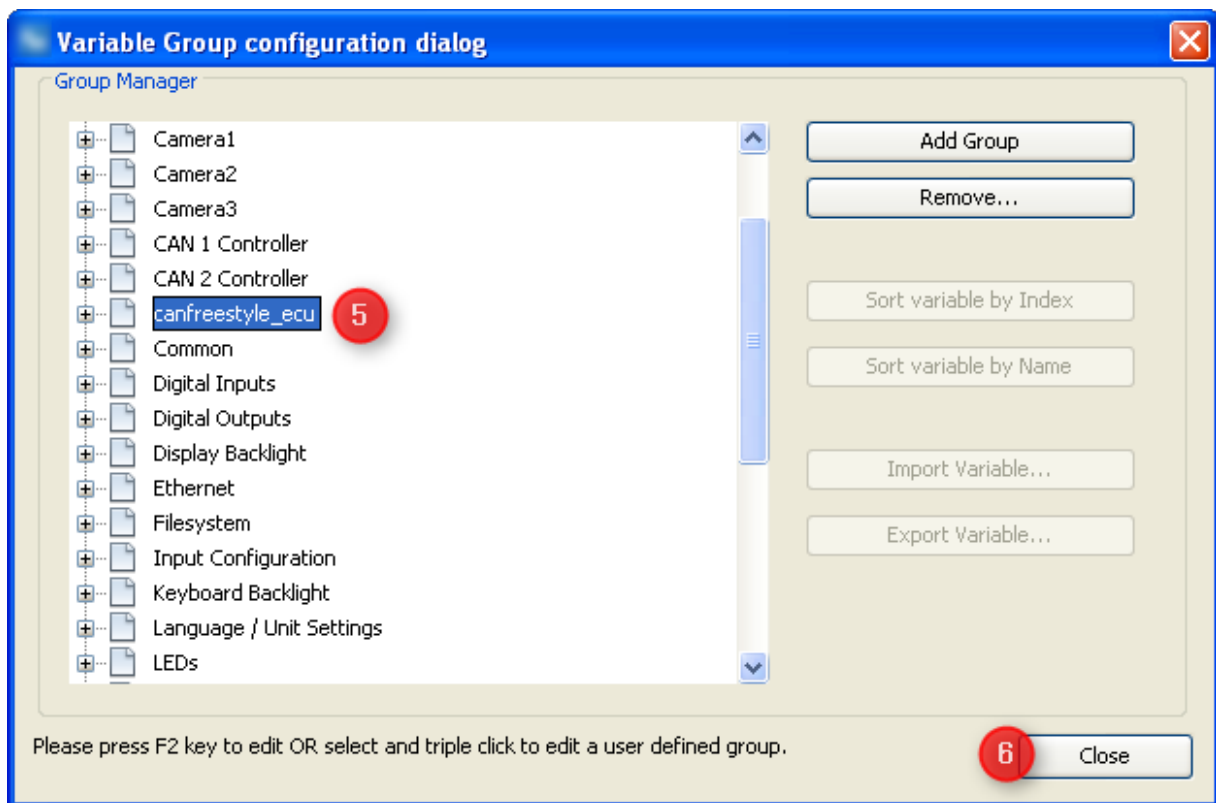
5. Once you pressed "Enter" to confirm the new group name it will move according to alphabetical order of all groups.
6. Press "close" once you're finished creating or deleting variable groups.

Creating CANFreestyle variables

Creation of CANFreestyle variables is similar to all other variables
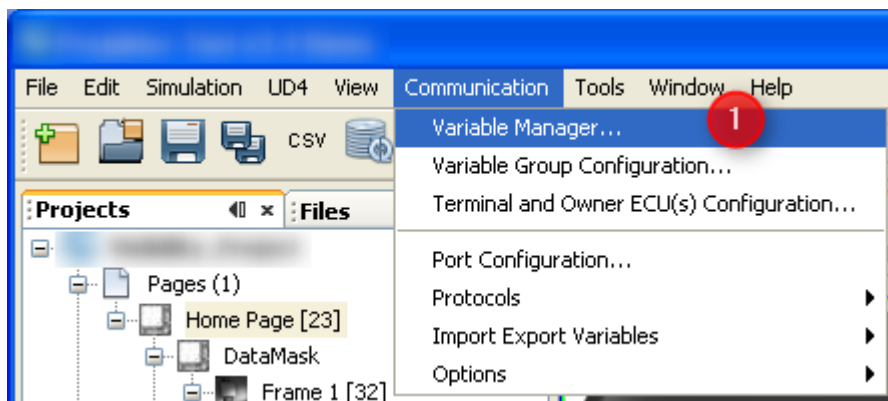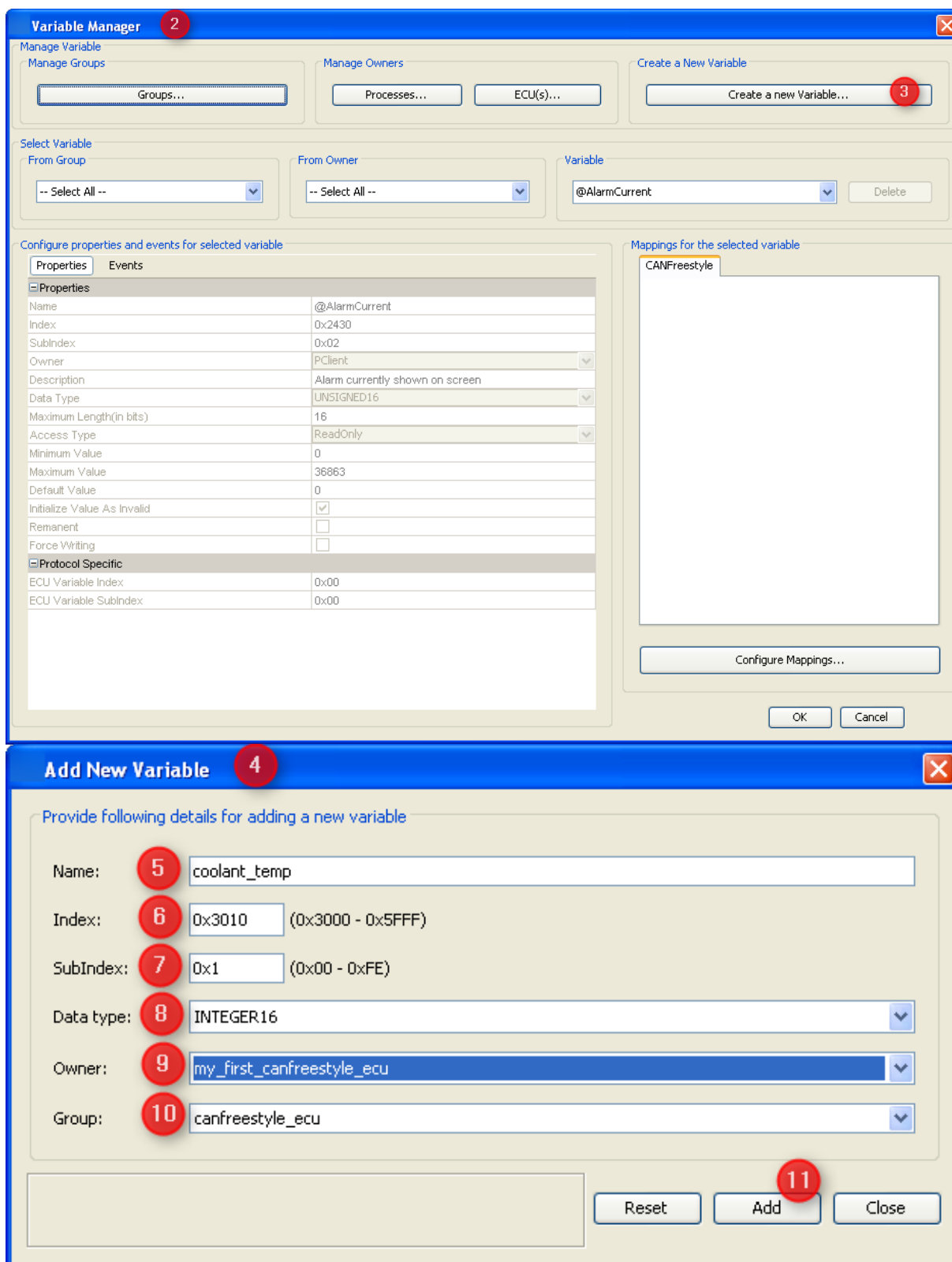
1. Open the "Communication" menu and choose "Variable Manager ..." entry
2. Variable Manager dialog opens (which can be used as shortcut to all variable related tasks)
3. Press "Create a new Variable ..." button
4. "Add new variable" dialog opens

5. Fill in a name of the variable
6. Type an index for the variable. You are free to enter the value in decimal or hexadecimal, starting with "0x". Anyways hexadecimal will be displayed. (Although the concept comes from CANopen you should use index and subindex as a means of ordering and grouping of variables)
7. Add a subindex for the new variable. Note: Combination of index and subindex has to be unique among all variables in the GUI definition.
8. Select signed or unsigned data type at least as wide as the variable will be in CAN message.
9. Change owner from default "PClient" to the newly created CANFreestyle ECU. Note: If you keep "PClient" you could handle the variable also but would not notice if the ECU is not available anymore.
10. Select the newly created variable group for the variable.
11. Press "Add" to finally create the new variable with the settings made above.

> ℹ Variable created successfully.

If the variables was created successfully the above message will be shown at the lower left box of the dialog.
Now you can either create more variables by editing the settings with or without "Reset" or leave the dialog with "Close".

After closing the "Add New Variable" dialog you can modify all the properties of the variable(s) just created or already existing in "Variable Manager" dialog:
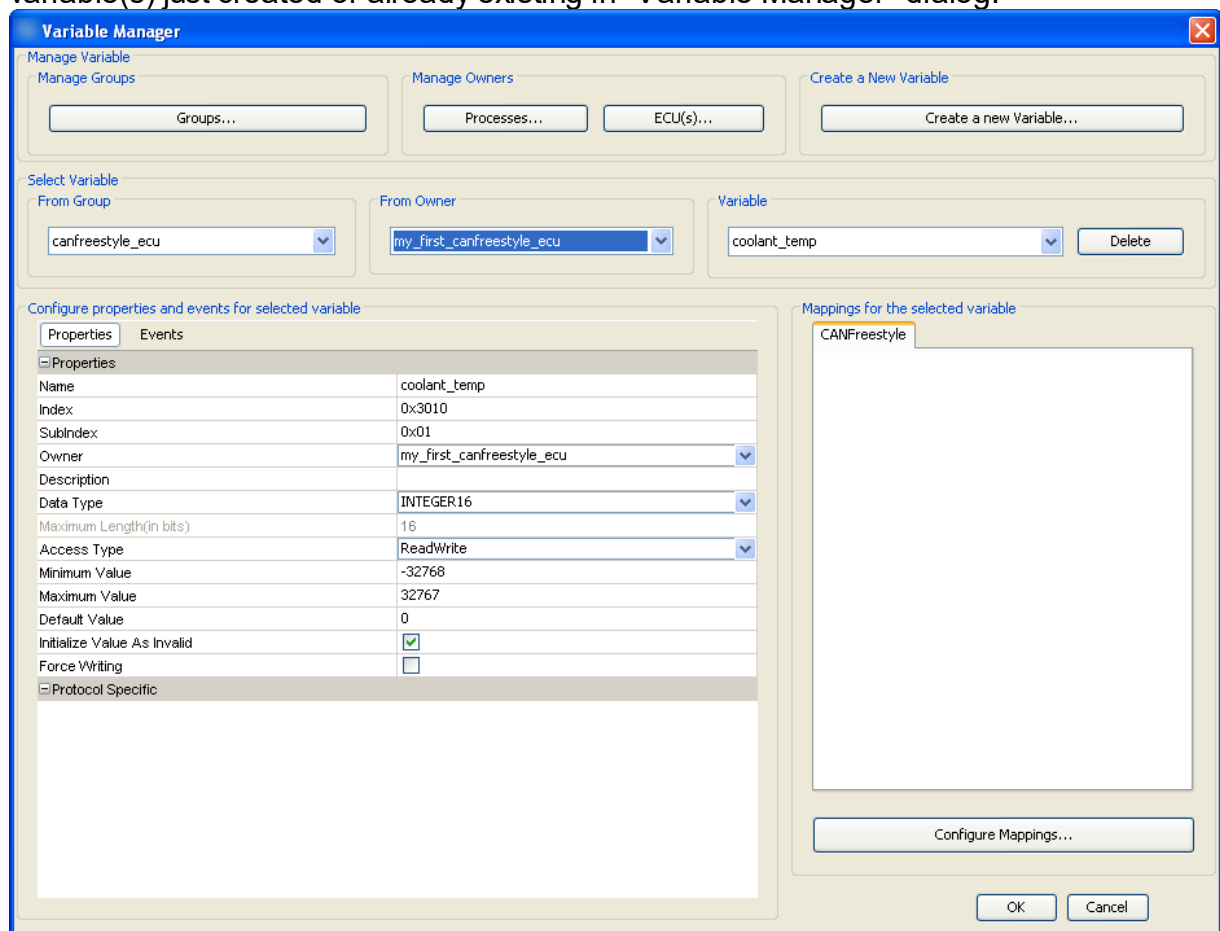
For a detailed description of all the variable properties and events please refer to [variable view](#)[146] .

There is a second option to create variables:
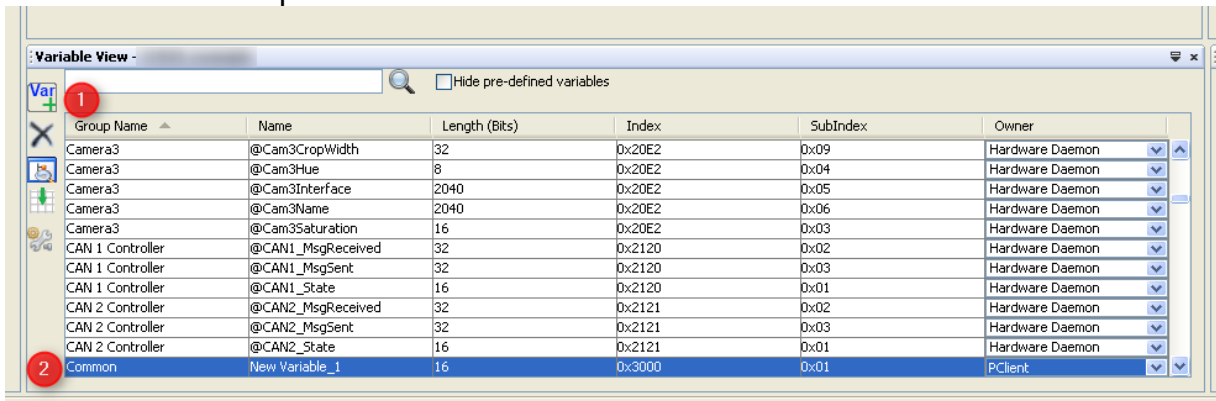
1. In variable view press the "Var +" button
2. A new variable with default settings appears
3. You can double click each property and edit it.

Warning: If you type a group name that was not defined yet, a new variable group will be created.
Warning: Depending on your sorting option the variable can "disappear" whenever a modification is confirmed.
More details on this option can be found in [variable view](#)[146] .
So the recommendation is to create variables following the first option and use variable view to manage and modify existing variables.

Defining CAN ID Masks

If you only want to evaluate specific bits of the CAN Identifier you can use **"CAN ID Masks..."** to create our own mask. This can be used for example if several ECUs are connected to the bus whose use same kind of identifier.
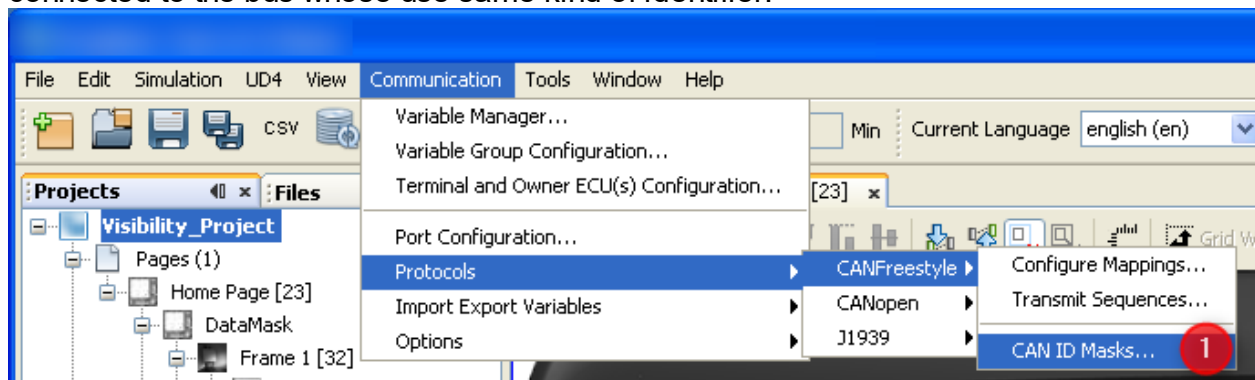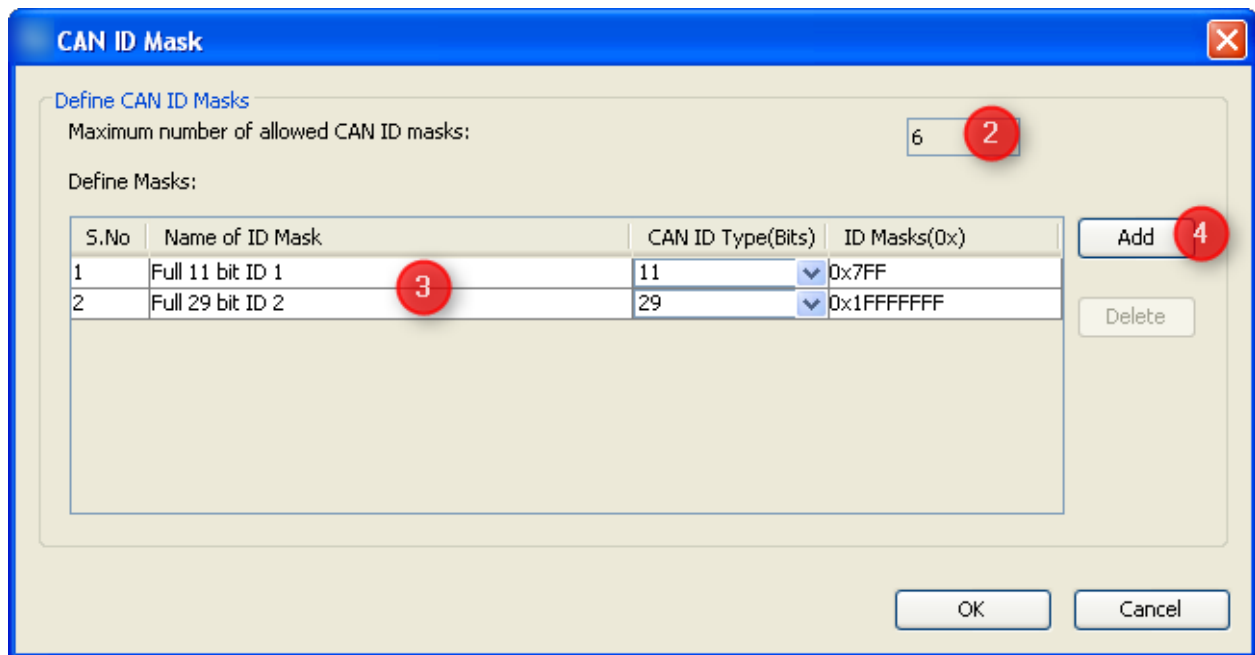
1. If you want to edit **CAN ID Masks** you can find it in the main menu like shown above. A new dialog **CAN ID Mask** will be open.
2. For performance reasons you can only define **6** CAN ID Masks
3. Two masks are predefined.**"Full 11 bit ID 1"** and **"Full 29 bit ID 2"**. So if you don't need to filter the CAN ID you can use the predefined ones.
4. If you want to add your own proprietary mask you have to press the **"Add"** button

If the **"Add"** button is pressed a new line will be added at the table which is editable.
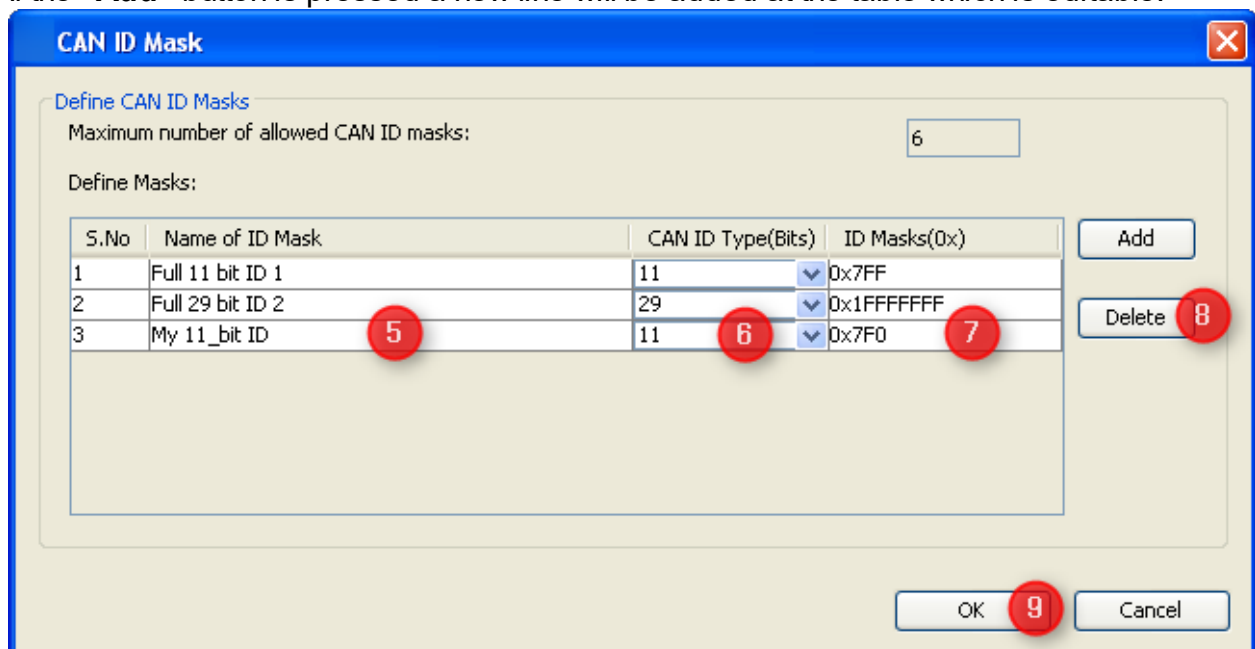
5. Here you can specify a name for the newly created mask.
6. You have to specify if the mask should be used for **11** or **29** bit CAN Identifiers.
7. In the field **ID Masks** you can specify the bits which should be evaluated for the CAN Identifier in hexadecimal format.

8. Deletes the currently selected mask. **NOTE:** The predefined mask can not be deleted.
9. To save the settings you have to press the **"OK"** button

The created Masks can be used here[319] .

Defining Variable Mappings

After having defined an ECU as variable owner and the variable itself we finally have to define the connection between the two. In other words: We need a definition about how to exchange the variable's value with the ECU.

These definitions are called "variable mappings" within this tool.

In order to define a variable mapping you can either use the shortcut button of the variable manager shown in the image above or open the respective dialog from the menu as follows:
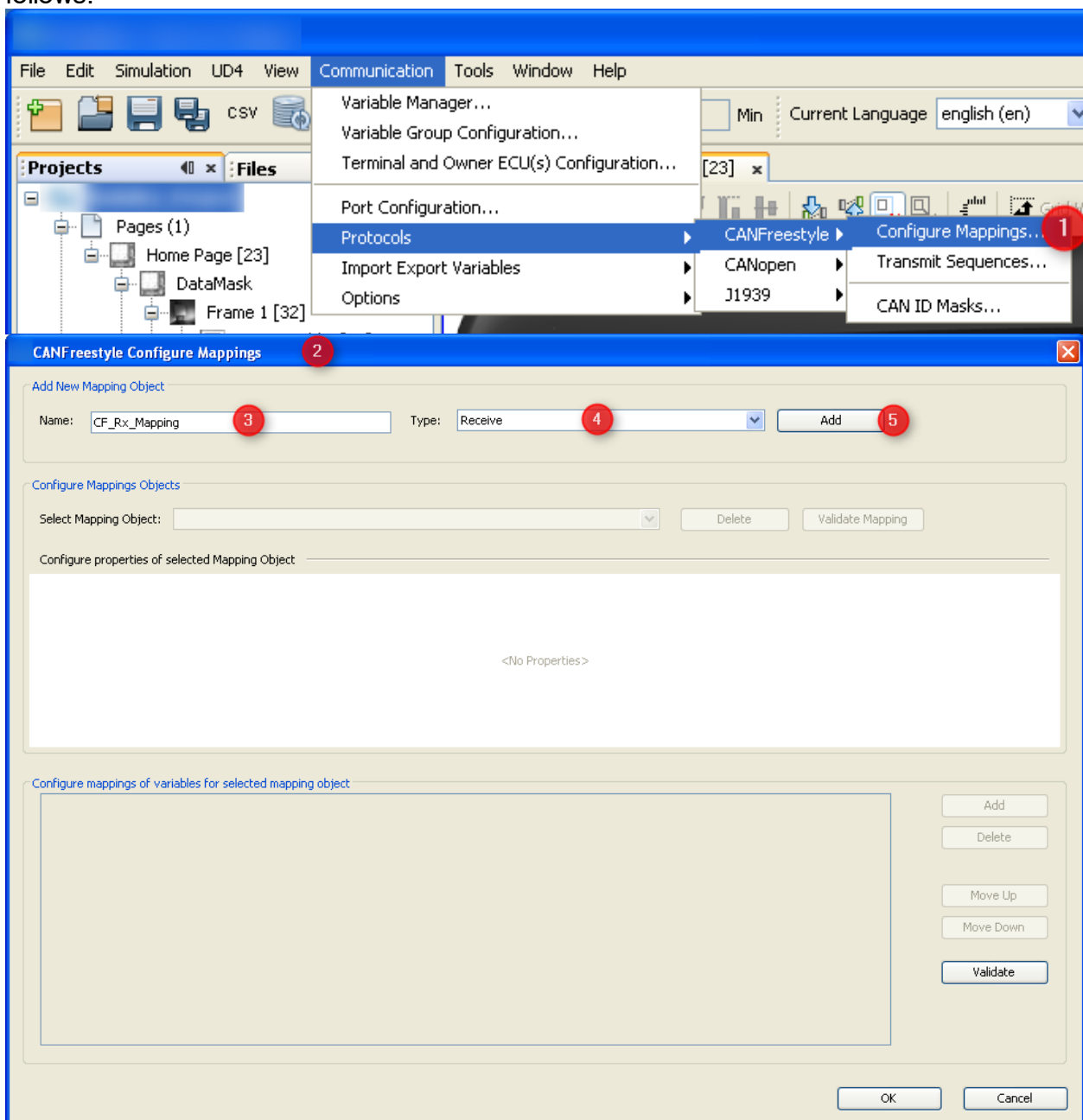
1. Open the menu "communication". Select "Protocols". Choose "CANFreestyle". Select "Configure Mappings ...".
2. The "CANFreestyle Configure Mappings" dialog will pop up.
3. Type the name of your mapping
4. Select the type "Receive" if the display shall fetch values from CAN bus and "Transmit" if the display shall send out values.
5. Press "Add" to create the new mapping.

Setting Mapping Properties

The dialog will populate with the default settings for the new mapping object which have to be modified according to your intentions:
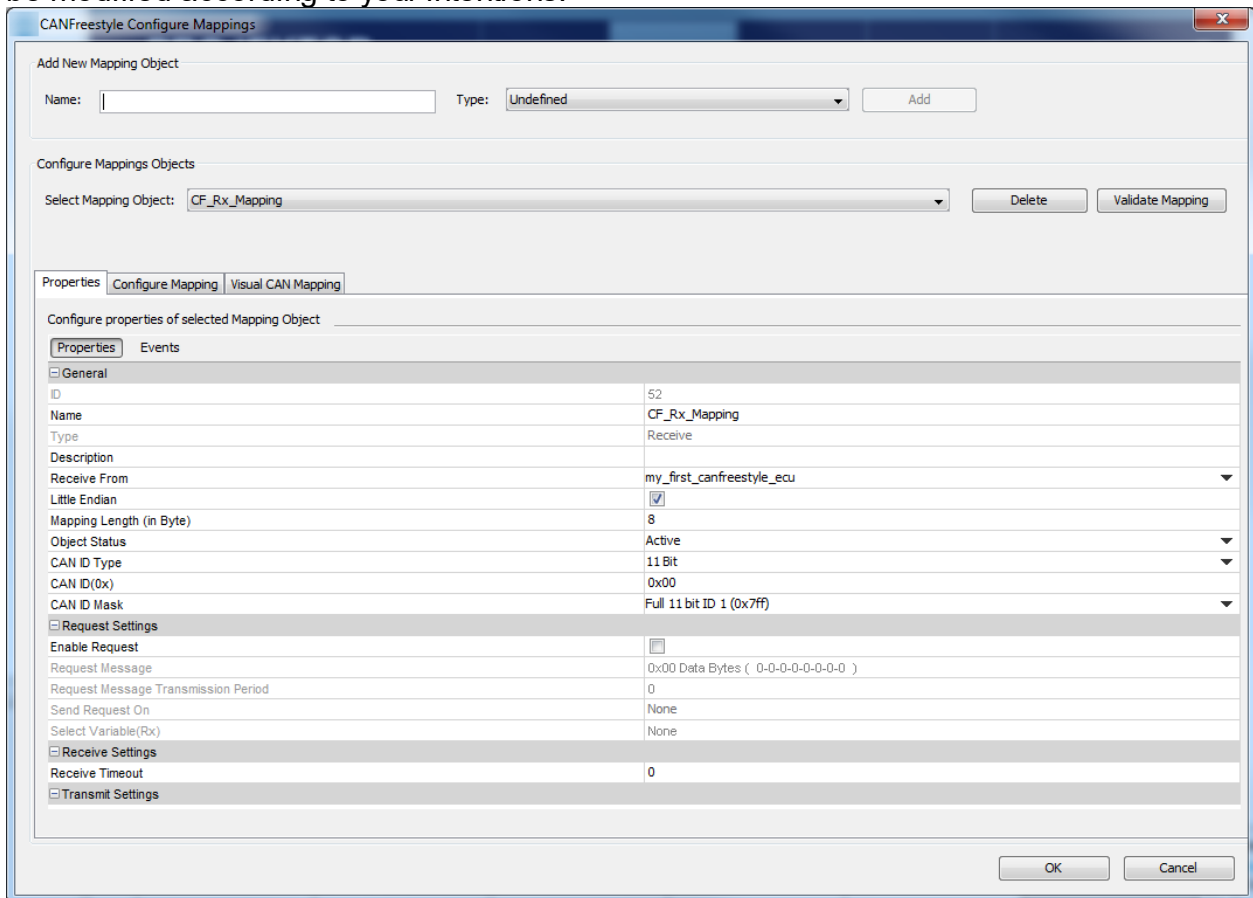
Defining Variable Mappings

General

ID

Is the object ID created internally which can not be modified.

Name

The name of the mapping as used during creation. This name can be edited here.

Type

"Receive" or "Transmit" - this property can not be modified. If this needs to be changed you have to delete and recreate the mapping.

Description

Should be a valid description of the mapping which helps understanding its purpose and context in future.

Mapping Variable Type

CANFreestyle only supports one type:

- **Length** based mappings find every variable (or constant) by position and length within the CAN message. If length based mappings are less than 8 bytes long, their variables or constants can be placed at any position. If they shall be longer, the variables and constants have to be byte aligned.

Receive from / Transmit to

For receive mappings: Select the ECU which will send the Mapping where the variable should be extracted.
For transmit mappings this property describes the destination address to be used (i.e. the ECU that shall receive the values).

Little Endian/Big Endian

CANFreestyle values are transferred least significant byte first (Little Endian) by default. Here you can change this to most significant byte first (Big Endian). If you have a Mapping which contains Little and Big Endian values you have to create two separate mappings.
This setting only applies to numeric values. Strings are expected to be "big endian" i.e. the first byte of the word or sentence is the first one on the bus.

Mapping Length

Enter the length of the message you want to configure. The length has to be between 0 and 8 bytes.

Object Status

With Object Status Active the mapping is working, i.e. it is processing received messages or sending them. With Object Status Inactive the mapping is not working. This property can be switched at run time with the [setProperty](#)[538] JavaScript function to enable / disable CAN mappings.

CAN ID Type

This setting defines whether an '11 Bit Identifier' or '29 Bit Identifier' should be used for the CAN ID.

CAN ID (0x)

This defines a CAN ID for the appropriate Mapping. The ID is **not** unique so it is possible do define several mappings with the same ID e.g. if a multiplexer is used.

CAN ID Mask

If not all bits of the Identifier should be evaluated it is possible to define a CAN ID Mask which only evaluate the needed bits of the CAN ID.
NOTE: You can define your own CAN ID Mask [here](#)[315].

Request Settings

These settings define if and when the display sends a request (for receive mappings) or answers a request for a transmit mapping.
The only pre-requisite is that the ECU sending the request is "known", i.e. defined in the ECU Configuration[309] .

Enable Request

For receive mappings this means that the display will send out a request to get the Mapping based on the following settings.
For transmit mappings this means that the request defined in the following properties will be recognized and answered with the transmit mapping.

Request message

This allows definition of a Message with fixed data section which will be sent out in order to get the receive mapping based on the following triggers.

Request Message Transmission Period

This is only enabled for receive mappings. It defines the interval in milliseconds the display will send requests for the mapping. Set to 0 if no periodic request is required.

Send Request On

- "Specific Variable Change" allows triggering the request based on the value change of any variable in the project.
- "None" means there is no trigger besides the time based one.

Select Variable (Rx)

Only enabled if this is a receive mapping, request is enabled and "Send Request On" is set to "Specific Variable Change". This allows setting the variable whose value change should trigger sending the request.

Receive Settings

Receive Timeout

This is intended for receive mappings which are expected to arrive periodically - either by default or by periodic request. If the mapping is not received in as many milliseconds as defined here all variable values of the mapping are considered invalid and hence all DDOs showing them will also be disabled (i.e. numeric fields shown with dashes, meters and such grayed out).

Transmit Settings

Use as Write Request

If checked this means that the transmit mapping will be sent out once if there is a write request to one of the variables contained in the mapping. Usually the display only sends values provided by the owners of the respective variables in transmit mappings.
This means that e.g. engine speed will only be sent out as provided by engine ECU. If in your system modifying the measured engine speed value is used to set a new desired engine speed then you can check this property and writing on the engine speed variable will send out the transmit mapping containing the new setpoint of the engine speed value.

Please note that all other values in the mapping will be the ones provided by their respective owners and even the new setpoint will in the next message be replaced by the last measured value provided by the owner.

If you should want to write to the same variable from many sites you should consider "PClient" as the variable owner which will accept all write requests but whose variables will never become invalid again once they have a value.

Send Value On

- **None:** Mapping is only sent time based (if configured) and on request.
- **Any Variable Change:** Whenever one of the mapped variables changes its value the mapping is sent out.
- **Specific Variable Change:** The mapping is sent out if the variable specified in the following property changes its value.

Select Variable (Tx)

Choose the variable whose value change should trigger sending out the mapping. This variable can be part of the mapping but doesn't have to be.

Transmission Period (in ms)

Defines the interval between messages for this transmit mapping. Set to 0 if time based sending is not desired.

Defining Data Section of CANFreestyle Mapping

After setting all the general properties of the mapping now the data part has to be defined.

Some general rules apply:

- All variable's and constant's positions define the position of the least significant bit (for little and big endian)
- Transmit mappings have to be defined completely - we can not send "gaps" on the bus.
- Receive mappings can have gaps - as long as the length can be determined.
- Receive mappings are recognized if all the following conditions are true:

```
    if (  ( (CAN-ID received AND configured CAN-ID Mask) ==
(configured CAN-ID AND configured CAN-ID Mask) )
      && ( (message length) == (message length
configured) )
      && ( (constants received AND configured mask) ==
(constants configured AND configured mask) ) )
```

Note: It is allowed to leave gaps in the data section of receive mappings. However, the message length has to be configured correctly in the Mapping Length property

Note: The last condition allows covering multiplexed Mappings by defining a separate mapping for each value of the multiplexer

To do this, use a constant value from the "Constants" group and adjust it to the length of the multiplexer region in your system.

Then map this constant to the appropriate byte/bit position. Fill the rest with the variables needed.

Create the mapping multiple times with different constant values. Each mapping will only react if its constant value equals the incoming message. The variables should match the message structure for the according multiplexer value.
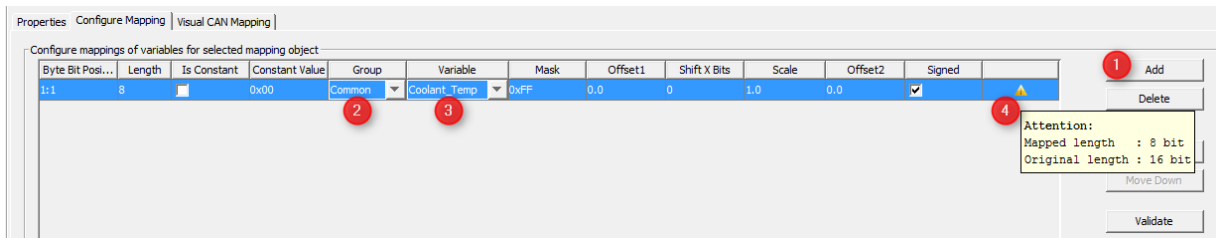
1. Press "Add" button to add a line to the mapping.
2. Select the group of the variable to map.
3. Select the variable itself.

After doing this the properties in the line adjust to the variable definitions made earlier e.g. the length is set to 16 bits and the mask is set to 0xffff meaning all the bits will be taken.

Events

This tab allows adding actions to mapping related events. It only appears for receive mappings.
The configured actions for the events are executed directly in the CAN thread when the event happens.
The CAN stacks will wait until the actions are executed.

OnReceiveTimeOut

This event is triggered if a receive mapping is not received within the period defined as "Transmission Period". If this happens all the variables mapped to the PGN are automatically considered invalid and additionally the action is executed.
If you should see any "Event Options" for this event - ignore them like the PClient does.
Possible actions that can be assigned are
- "No Action" which is the default
- "Set Value" to assign a constant value to a variable, e.g. to reset a counter, raise an alarm or trigger something attached to a variable where "Force Writing" is true.
- "Execute Script" allows to execute a script and provides additional parameter "Execute on every Xth Event" so that it is possible to execute the script only for e.g. every third timeout.

OnReceiveTimeOutResolved

This event is triggered if a receive mapping which time out before is received again. If this happens all the variables mapped to the PGN are automatically considered valid again (assuming the PGN received really contained a value for them) and additionally the action is executed.
If you should see any "Event Options" for this event - ignore them like the PClient does.
Possible actions that can be assigned are
- "No Action" which is the default

- "Set Value" to assign a constant value to a variable, e.g. to reset a counter, raise an alarm or trigger something attached to a variable where "Force Writing" is true.
- "Execute Script" allows to execute a script and provides additional parameter "Execute on every Xth Event" so that it is possible to execute the script only for e.g. every third timeout.

OnMappingReceived

The action for this event is executed EVERY TIME the mapping is received (regardles of value changes).
When a script action is executed, the variables used in the mapping will already have been updated with the values from the CAN message.
Do NOT execute complex scripts at this event if the mapping is received very often!
Monitor the performance on the device with a realistic cycle time of all CAN messages.
Monitor the cycle time of transmit messages (the CAN stack has to wait until the received action is over before taking care of other tasks like sending out transmit mappings).
If you should see any "Event Options" for this event - ignore them like the PClient does.
Possible actions that can be assigned are

- "No Action" which is the default
- "Set Value" to assign a constant value to a variable, e.g. to reset a counter, raise an alarm or trigger something attached to a variable where "Force Writing" is true.
- "Execute Script" allows to execute a script and provides additional parameter "Execute on every Xth Event" so that it is possible to execute the script only for e.g. every third timeout.

⊞ CANFreestyle mapping variable properties

The following settings can be made for each variable added to the mapping:

Byte Bit Position

Position is shown as byte#:bit# where byte 1 is the byte closest to the CAN identifier and bit 1 is the least significant bit of a byte. I.e. full CAN message starts with byte 1 directly behind the CAN ID and ends with byte 8. In a byte with value 1 bit 1 is set, in a byte with value 128 bit 8 is set.
These enumeration scheme matches the one used in J1939 standard documents so that you can directly copy the position. Engine coolant temperature is defined as the first byte - so set start position to 1:1.
Alternatively one can type the bit position directly: 1 refers to the bit closest to the CAN identifier and 64 is the bit at the end of the CAN message.
Warning: This input does not always behave as expected when typing position by hand. Finally the displayed "byte:bit" value is relevant.
The mapping is filled correctly if there are no gaps i.e. the whole message gets filled with values.
Please refer to the following image for byte and bit enumeration within a CAN message. This enumeration is valid for little and big endian variables since it only refers to the CAN message itself.

byte	CAN ID	1								2								3								4								5								6								7								8							
bit of byte		8	7	6	5	4	3	2	1	8	7	6	5	4	3	2	1	8	7	6	5	4	3	2	1	8	7	6	5	4	3	2	1	8	7	6	5	4	3	2	1	8	7	6	5	4	3	2	1	8	7	6	5	4	3	2	1	8	7	6	5	4	3	2	1
bit of CAN message		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64

Length

Defines the size of the mapped numeric variable or constant in bits. If a string is mapped this defines the length of the string in bytes.
Mapped length can be shorter than the length defined for the variable but not longer. By default the variable length is used.

Is Constant

Enable this checkbox to define a constant. Depending on the mask constants can be used to fill gaps or recognize multiplexers in receive mappings. In transmit mappings constants are required for every part of the message not filled with a variable.
Uncheck this property to map a variable.

Constant Value

Although appearance does not change this property is only valid and editable for constants i.e. lines where "Is Constant" property is set.
If valid this defines the numeric value of the constant. Please note that constants are always expected to be numeric - there is no option to define string constants.

Group

Variable group containing the variable to be mapped. Does not apply to constants.

Variable

The variable to be mapped.

Mask

Variable or constant are bitwise ANDed with mask before they are applied.
See [receive condition](#)[319] on how this can be used to receive multiplexed mappings.
See section [mapping calculations](#)[325] for details about how this property applies to receive and transmit mappings.
Mask is not used for String variables.

Offset1

Offset1 is a floating point value applied to the received value after mask and shift factor but before scale and offset 2.
See section [mapping calculations](#)[325] for details about how this property applies to receive and transmit mappings.
Offset 1 is not used for String variables.
Values can be entered in decimal form or in scientific notation / floating-point representation.

Shift X Bits

Shift received value to the right by as many bits as defined here. Negative values shift to the left.
This value is applied after the mask but before offset and scale.
See section [mapping calculations](#)[325] for details about how this property applies to receive and transmit mappings.
Shift is not used for String variables.

Scale

Floating point value to scale the received value before applying it to the mapped variable. Is applied after mask, shift and offset 1 but before offset 2.
Scale is not used for String variables.
Values can be entered in decimal form or in scientific notation / floating-point representation.

Offset2

Floating point value applied after all the other calculations are done.
See section mapping calculations[325] for details about how this property applies to receive and transmit mappings.
Offset2 is not used for String variables.
Values can be entered in decimal form or in scientific notation / floating-point representation.

Empty Field

This a simple information field. It shows a "Warning" icon if e.g a variable is not completely mapped.
In the example(4)[322] a 16 bit variable is mapped only with 8 bits, so the warning is shown.

Mapping calculations

The calculations applied to a mapping are defined mainly for receive mappings
For transmit mappings all calculations are inverted so that the same values can be applied to receive and transmit mappings.
The formula converting a mapped area of a CAN message into the variable value is as follows:

$$\text{variable\_value} = ((((\text{ CAN\_value} >> \text{shift}) \& \text{mask}) + \text{offset1}) * \text{scale}) + \text{offset2}$$

Consequently the calculation applied to transmit mappings inverts the above:

$$\text{CAN\_value} = ((((\text{ variable\_value} - \text{offset2}) / \text{scale}) - \text{offset1}) \& \text{mask}) << \text{shift}$$

Note: For constants only "*shift*" and "*mask*" are applied.
Note: Currently signed/unsigned calculation is performed based on variable definition. If variable is signed this means the CAN_value is considered negative if the most significant bit of the mapped area is set.

Sequences

Sequences can be used to send several transmit mappings periodically. For example it can be used as a simple kind of transport protocol.

Create Sequences

Before you can create Sequences you have to define transmit mappings. For more information please have a look at chapter **Defining Variable Mappings**[317]
If transmit mappings are already defined please select in the main menu **"Transmit Sequences..."** as shown below:
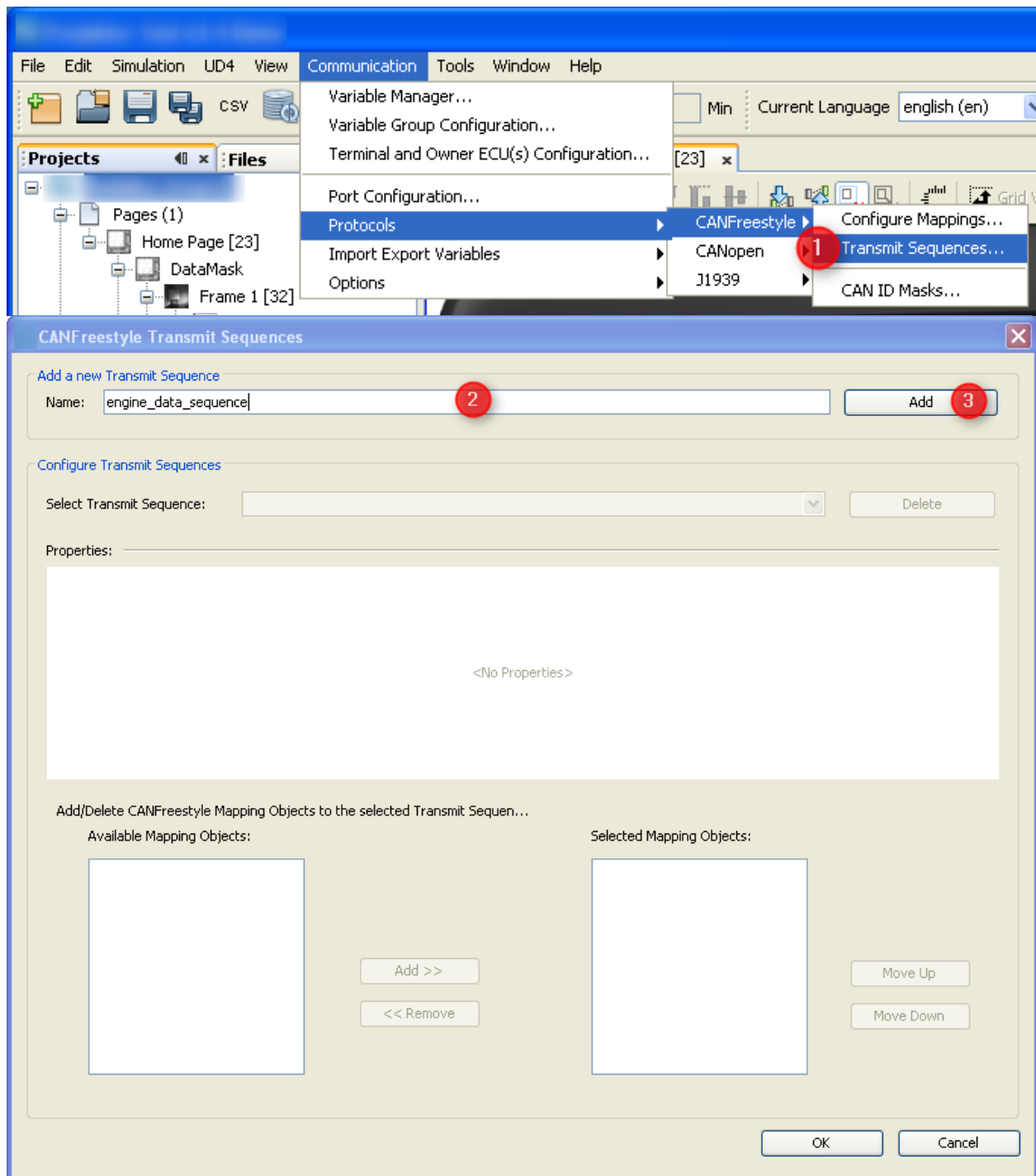
1. Open Menu **"Transmit Sequence..."**
2. Enter a **Sequence Name** in the new dialog named **CANFreestyle Transmit Sequence** which appears.
3. Press **"Add"** button

Afterwards following should be visible
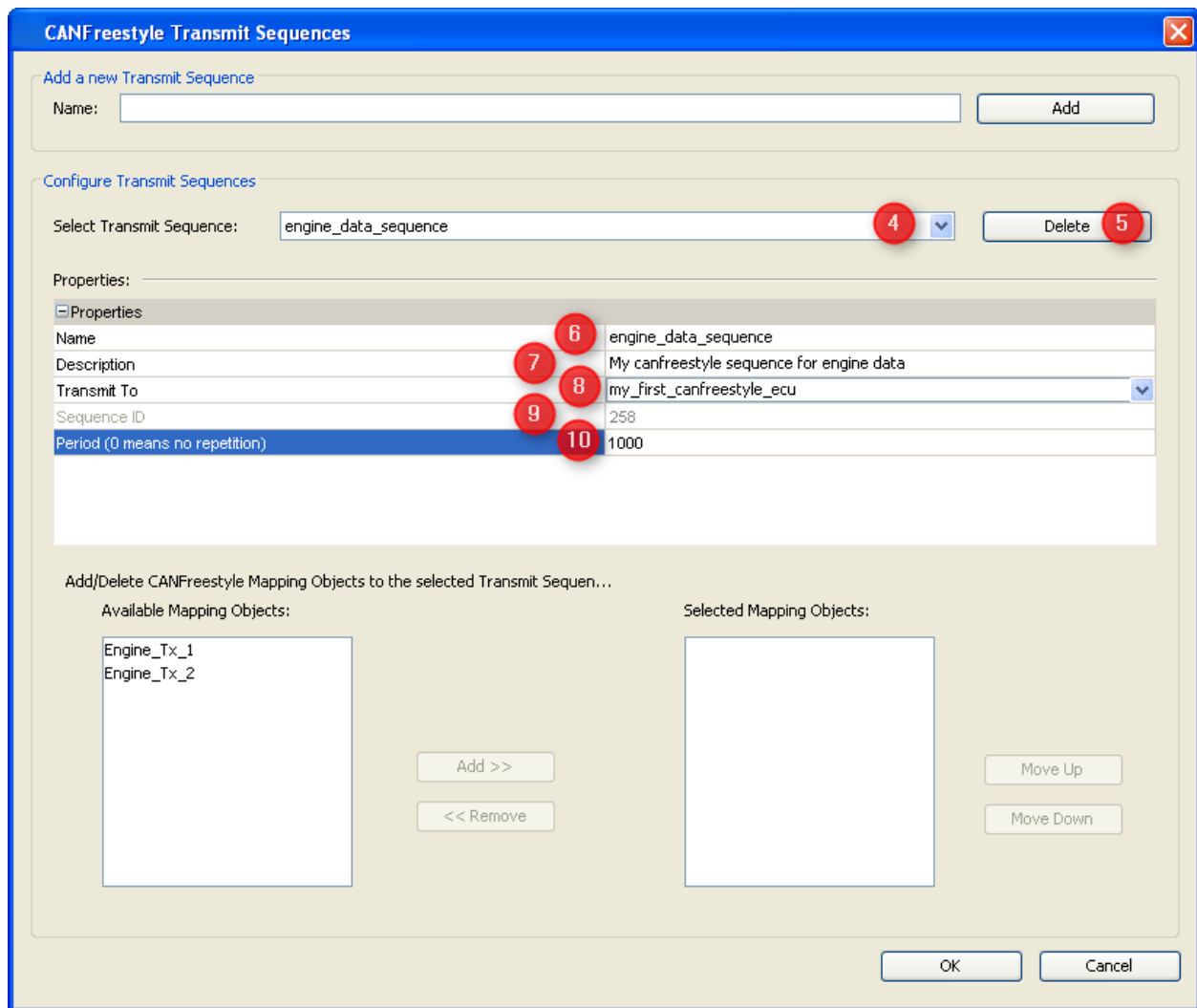
4. Combo box where you can select the sequence you want to edit.
5. Deletes the current selected sequence.
6. Name of the transmit sequence which is currently selected
7. Here you can add a description for the sequence
8. Target owner which should receive the sequence.
9. The "**Sequence ID**" is a unique internal ID, which is not editable.
10. The **"Period" (in ms)** defines the cycle for sending the sequence. **NOTE:** This is the only way to trigger the sequence.
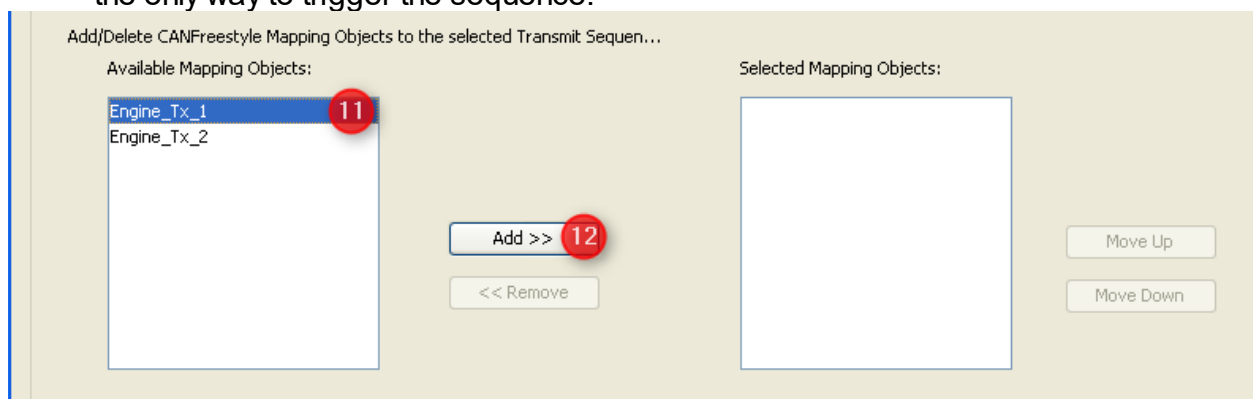
11. Select the **mapping** you want to add to the sequence.
12. Press **"Add >>"** to add the message to the sequence

After adding of mapping objects the dialog should look like this:
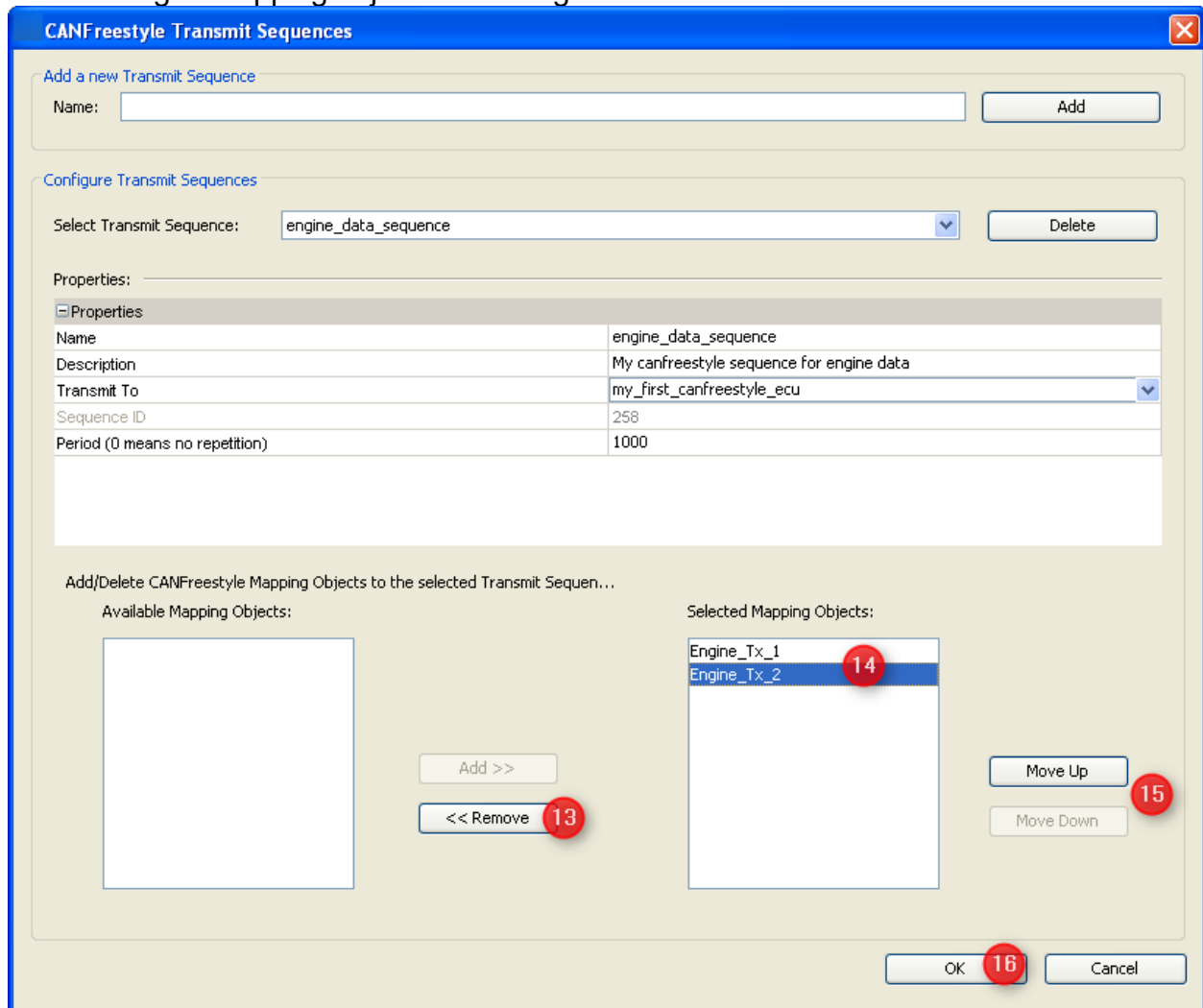
13. If a mapping was mapped by mistake you can select the mapping and remove it again with the **"<< Remove"** button
14. The list box contains the transmit mapping which should be transmitted via the sequence
15. With the help of the buttons **"Move Up"** and **"Move Down"** you can change the order of the transmit mapping at the sequence

Finally the dialog should be confirmed with **"OK"** to save the sequences.

5.2   CANopen

Introduction

Aside from CANFreestyle and J1939 protocol, it is possible to connect the device into a CANopen network.
The CANopen stack is implemented on the standard CIA 301 4.2.0 and support the following features:

- SDO Communication (Server and Client implementation)
  - Download and Upload for expedited and segmented protocol

- - Abort SDO Transfer protocol
- NMT (Slave implementation)
  - NMT transitions
  - NMT states
- PDO Communication
  - Asynchronous communication
  - acyclic and cyclic communication
- EDS File

The following Features are **NOT** implemented:
- EMCY Messages
- Node Guarding
- Block transfer SDO communication
- RTR
- Sync producer
- Auto start
- RWR, RWW Access type
- Objects

Object Dictionary: Objects list

The following objects are implemented by default in the CANopen stack of the device:

Objects
0x1000 – Device Type
0x1001 – Error Register
0x1002 – Manufacturer Status Register
0x1003 – Pre-Defined Error field
0x1005 – COB-ID Sync message
0x1008 – Manufacturer Device Name
0x1009 – Manufacturer Hardware Version
0x100a – Manufacturer Software version
0x1012 – COB-ID Time Stamp Object
0x1014 – COB-ID Emergency Object
0x1016 – Consumer heartbeat time
0x1017 – Producer Heartbeat time
0x1018 – Identity Object
0x1200 – 0x127f – Server SDO parameter
0x1280 – 0x12ff – Client SDO Parameter
0x1400 – 0x15ff – Receive PDO Communication Parameter
0x1600 – 0x17ff – Receive PDO Mapping Parameter
0x1800 – 0x19ff – Transmit PDO Communication Parameter
0x1a00 – 0x1bff – Transmit PDO Mapping Parameter

—

Communicate using CANopen protocol

To set up the CANopen protocol go to menu communication-> Terminal and Owner ECU(s) Configuration (1)
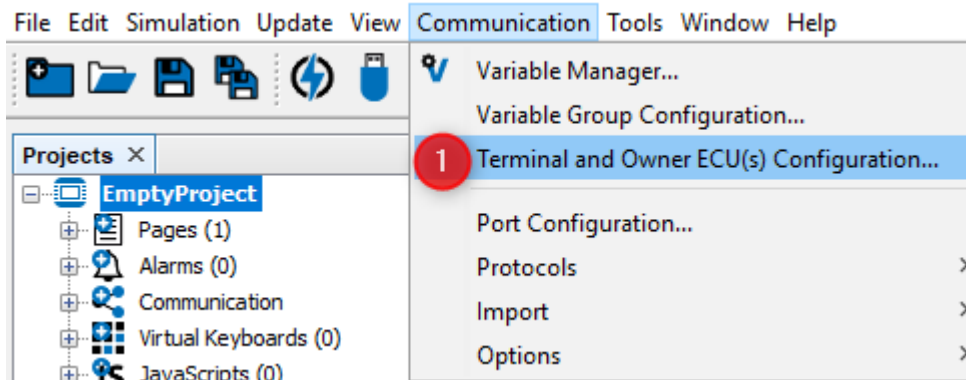
Then the user can choose the protocol CANopen on the desired CAN port
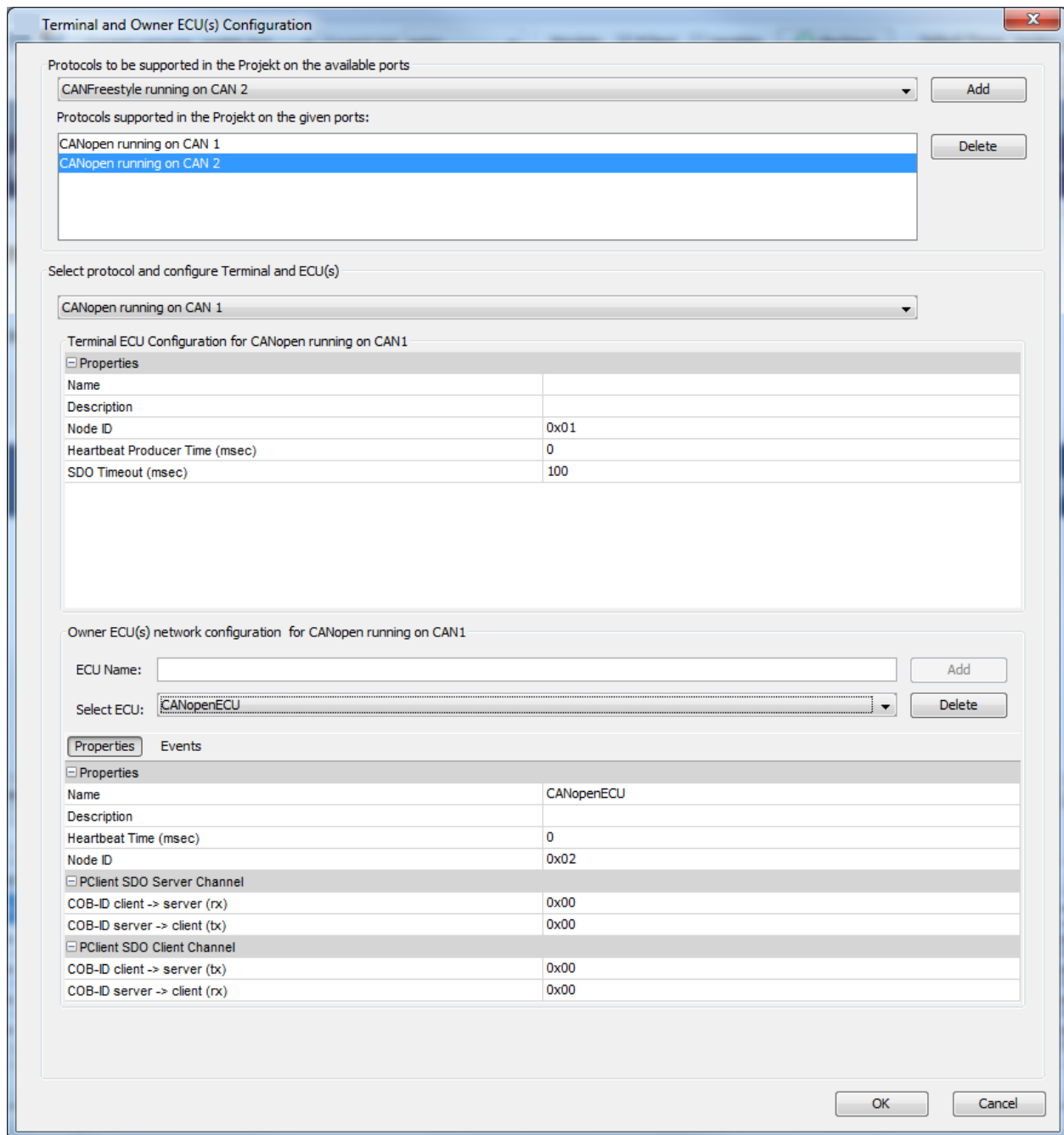
Dialog to set up the CANopen protocol on a device

CAN Port instance configuration

Step 1 : Selecting a port protocol instance
        To add a CANopen protocol instance, select the protocol to be supported on the desired port and press Add
        The added protocol instance is now selected in the view Select Protocol and configure Terminal and ECU(s)

Step 2: Selection and configuration of device node
        After adding or selecting a CANopen protocol instance on a CAN port, the user can update following properties

Properties	values / Range	Default values
Name	max 255 characters	Name ofd the selected panel
Description	max. 255 characters	
Node ID	0x1 -0x7F	0x1
HeartBeat Producer (ms) (Object 1017)	0-65535	0
SDO Timeout (msec)	100 - 1000	100

Step 3: Configuration of other nodes in the CANopen Network

After selecting and CAN port, a "partner" node shall be added in order to communicate with the device by pressig the Add button.

The new device properties are:

Properties	values / Range	Default values / Information
Name	max 255 characters	
Description	max. 255 characters	
Node ID	0x1 -0x7F	0x2
HeartBeat Timer (ms) (Object 1016 Sub1 - 127)	0-65535	0
PClient SDO Server Channel (object 1201-127F)		These COB-IDs are used if the ECU requests a value from the display device. The ECU requests a value from the display (Client -> Server (rx from the view of the device)) and the display answers (Server -> Client (tx from the view of the device))
COB-ID Client ->Server (rx)	0x601 - 0x699	0x600 + ${Node-ID} of panel on the selected CAN Port
COB-ID Server->Client (tx)	0x581 - 0x599	0x580 + ${Node-ID} of panel on the selected CAN Port
PClient SDO Client Channel (object 1280-12FF)		These COB-IDs are used if the display requests a value from the ECU. The display requests a value from the ECU (Client -> Server (tx from the view of the device)) and the ECU answers (Server -> Client (tx from the view of the device))
COB-ID Client ->Server (tx)	0x601 - 0x699	0x0
COB-ID Server->Client (rx)	0x581 - 0x599	0x0

For the new CANopen Device, events can be configured:

Events	Options (1)	actions (2)
OnHeartBeatTimeout	Sound pattern	Set Value Execute Script
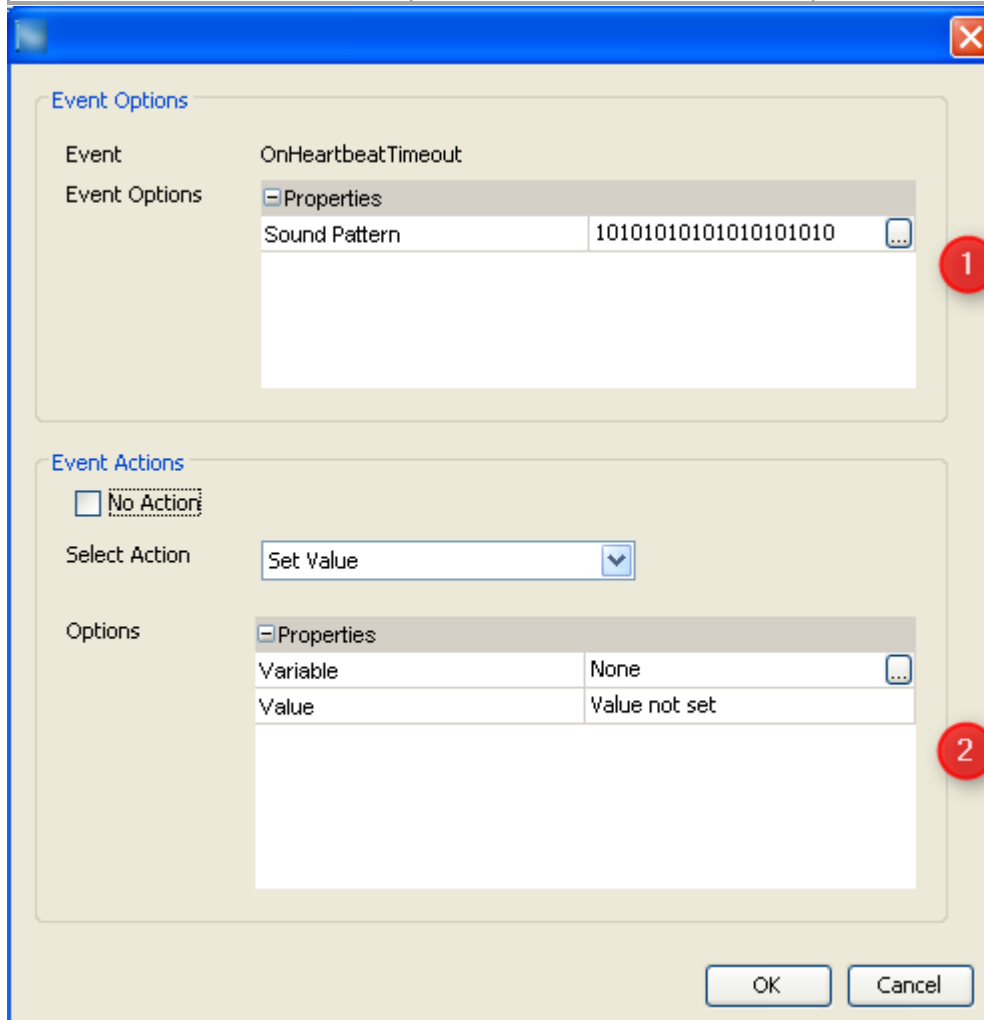
Configuration of event OnHeartBeatTimeout.
Step 4: Deleting of CANopen protocol instance

User can delete a whole CANopen protocol instance from the project. All CANopen variables will be assigned to the owner PClient. PDO configuration will be completely deleted.

Step 5: Deleting of CANopen ECU node

User can delete a CANopen ECU node. All related variable will be assigned to the PClient owner

Creating/ Updating/ Deleting of CANopen Variables

Once the user created a CANopen instance and CANopen ECU, he can create manufacturer objects (CANopen variables).
Goto menu Communication -> Variable manager
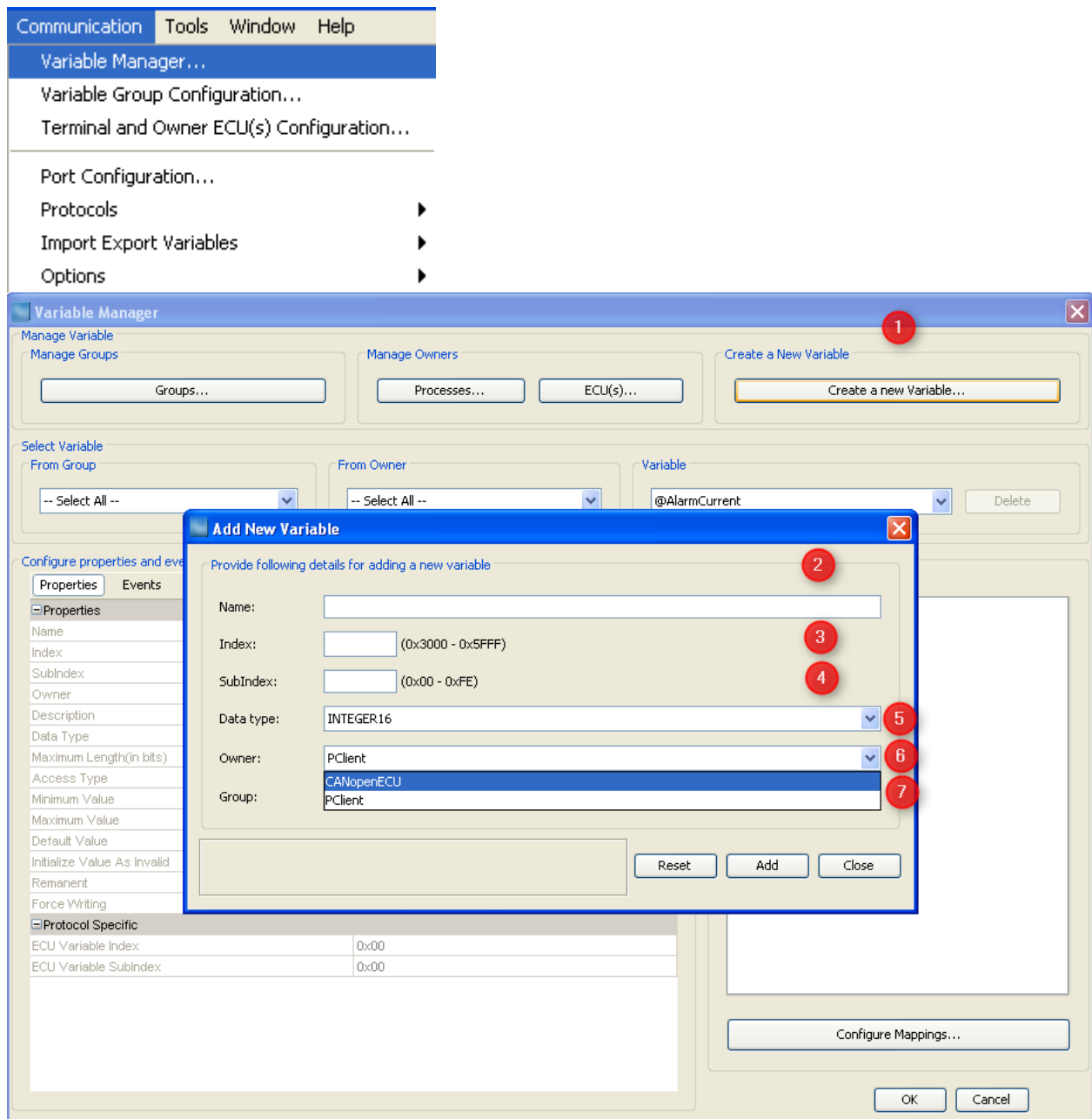
Step1: Press on button Create a new Variable

By pressing on this button, the dialog Add New Variable appears.

User can configure following properties:

Properties	values / Range	Default values
Name (2)	max 255 characters	blank
Index (3)	0x3000-0x5FFF	blank
Sub-Index (4)	0x0-0xFE	blank
Data type (5)	BOOLEAN INTEGER2 INTEGER8/UNSIGNED8 INTEGER16/UNSIGNED16 INTEGER32/UNSIGNED32 + User defined Data type (if created)	INTEGER16

Properties	values / Range	Default values
Owner (6)	List of defined Owners + PClient	PClient or last used Owner
Group (7)	User defined group + Common	Common

Step 2: By pressing Add the new variable will be created with selected configuration assuming that no issue is existing
Step 3: By pressing Close, the dialog Add a new Variable will be closed.
Alternative Step 2: By pressing Reset, all default values will be displayed.
Step 4: Once closed, other properties/ events can be updated using the variable manager properties / events views.

The last created variable will be displayed. User can select an other variable by selection Owner/Group/Variable

Following properties / Events can be configured

Properties	values / Range	Default values
Name	max. 255 characters	
Index	0x3000 - 0x5fff	as created
Sub Index	0x0-0xFE	as created
Owner	CANopen owner e.g. CANopenECU	as created
Description	blank max 255 Characters	blank
Data type	BOOLEAN INTEGER2 INTEGER8/UNSIGNED8 INTEGER16/UNSIGNED16 INTEGER32/UNSIGNED32 + User defined Data type (if created)	INTEGER16
Maximum Length (in bits)	length in bits for INTEGER/UNSIGNED variables length in bytes for STRING variables	16
Minimum Value	maximum value for selected data type	32767
Maximum Value	minimum value for selected data type	-32768
Default Value	minimum value - maximum value for selected data type	0
Initialize Value as Invalid	False|True	True
Remanent	False|True	False
Force writing	False|True	False
Protocol Specific		

Properties	values / Range	Default values
ECU Variable Index	0x0-0x9FFF	0x0
ECU Variable Subindex	0x0-0xFE	0x0

For the new CANopen Device, events can be configured:

Events	Options (1)	actions (2)
OnValueChangeByOwner	Sound pattern	Set Value Scale Value Execute Script
OnValueChangeRequest	Sound pattern	Set Value Scale Value Execute Script

Configure the SDO Communication

Once the CANopen protocol instance and CANopen ECU are created, all variables are available from the SDO communication as defined in the CIA 301 CANopen Standard
For example:
According, the user 's defined a alarm with number 1. To display, the alarm on device, following CAN message will be sent:
[601|8| 22 30 24 01 01 80 00 00]
als Response from SDO Server, the message is sent:
[581|8| 60 30 24 01 00 00 00 00]

SDO Client /Server of PClient

For each CANopen variable, user can configure the Index and Subindex of the manufacturer object on his own device with properties ECU Variable Index and ECU Variable Subindex.

The device can also send an SDO Download Write Request to the CANopenECU using this setting and PClient Client Channel as defined in the Terminal and Owner ECU(s) Configuration

Example:

The user wants to communicate via SDO Communication between his own CANopen ECU and device. By updating a CANopen variable on the device using Java Script or Encoder, a SDO
Download Write Request will be sent to the CANopen ECU.

Assuming a CANopen protocol instance is already created and a CANopenECU is added.

User configures CANopenECU Properties
PClient COB-ID Client -> Server (tx) to 0x602
and PClient COB-ID Client -> Server (rx) to 0x582 if the CANOpen ECU Node-ID is set to 0x2

User creates a CANopen Variable with CANopenVar with index 0x3000,0x1 and set the ECU Variable Index to 0x4500 and ECU Variable Subindex 0xdd (as defined on his own ECU). Other properties are set to default .

User assigns the variable into a numeric field as input for example. On the device, the variable will be be updated using the Encoder to value 0x2. The following SDO Download message will be sent by the device

[602|8| 2B 00 45 DD 02 00 00 00]
As soon as the answer is received from device, the value will be updated in the DDO.
[582|8| 60 03 24 01 00 00 00 00]

SDO Client/ Server of CANopen ECU

Since version 1.12 the CANopen stack does support SDO Upload Requests (read a value from a CANopen ECU's object dictionary).
Preconditions for using SDO Upload Requests:
- The variable which shall be read via SDO Upload request needs to be owned by a CANopen ECU
- The *PClient SDO Client Channel* COB-IDs of the CANopen ECU need to be configured
- The *ECU Variable Index* and *ECU Variable Subindex* in the *Protocol Specific* section of the variable properties need to contain the index and subindex of the variable in the ECU's object dictionary.

The only way to trigger the sending of a SDO Upload Request is the new JavaScript function requestVariableValueFromOwner[548] .
Rules that apply for SDO Upload Requests:

- There may be only one active SDO Upload Request. If more Upload Requests are triggered via requestVariableValueFromOwner, they will be queued until either a timeout happens (see below) or the ECU answers with a SDO Upload Response.
- The ECU has to answer to an SDO Upload Request within 100ms. If the answer takes longer or is not answered at all, an SDOAbort message will be sent (with error code *SDO Protocol Timeout*).
- The CANopen stack will only react to SDO Upload Responses if an SDO Upload Request has been sent before (simulating a request with e.g. CANFreestyle will not work).

Configure the PDO Communication

PDO (Process Data Communication) have a length of 8 bytes and they can be configured under
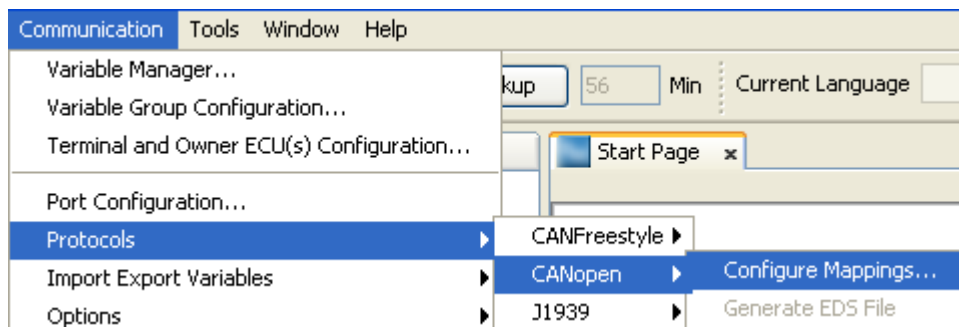
or by pressing the button Configure Mapping... in Variable Manager Dialog
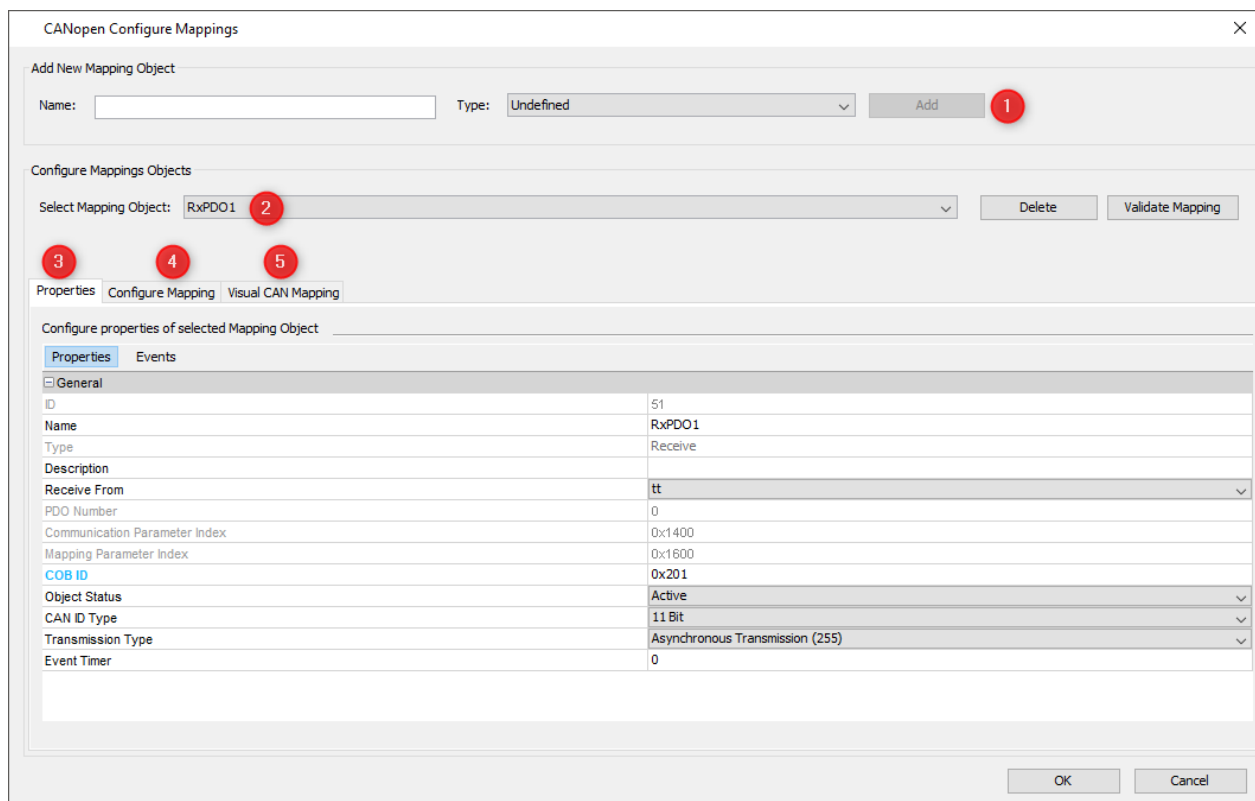
Step 1: Add a new Mapping Object

The user enters a name for the PDO and selects the PDO type Receive or Transmit.
By pressing Add , the PDO will be created and appears as selected in the field Select
Mapping Object.

Step 2: User selects the PDO

Step 3: User can update the following Properties /Events

Properties	values / Range	Default values
Name	max. 255 characters	as created
Type	Receive|not editable Transmit	as created
Description	max. 255 characters	blank
Receive From Transmit To	List of CANopen ECU + PClient	first created CANopen ECU
PDO Number	automatic incremented|not editable 0- 512	0
Communication Parameter Index	automatic incremented| not editable For Receive 0x1400- 0x15FF For Transmit 0x1800-19FF	For Receive 0x140{0-1FF} For Transmit 0x180{0-1FF}
Mapping Parameter Index	automatic incremented| not editable For Receive 0x1600- 0x17FF For Transmit 0x1A00- 0x11FF	corresponding to communication Parameter Index + 0x200
COB-ID	0x201-0x599 if a value is written outside of the CANopen COB-ID range, then a confirmation dialog will be displayed. By clicking on Yes, the new value will be accepted	
Object Status	Active|Inactive	Active
CAN ID Type	11 Bit| 29 Bit (29 bits can not be selected)	11 Bits
Transmission Type	Asynchronous Transmission (255) Asynchronous Transmission (254) Cyclic Synchronous Transmission (Only configurable for Transmit PDOs) Acylic Synchronous Transmission (requires a	as created

Properties	values / Range	Default values
	sync* message on the bus for messages to be processed)	
Event Timer	0 - 65535 For Receive: will be handled as time out For Transmit: period for sending CAN message	0
Inhibit Time	Only configurable for Transmit PDOs 0-65535 (milliseconds)	0
Sync Value	Only configurable for Transmit PDOs for cyclic synchronous 0-240	1

* The sync message is sent on COB-ID 0x80 with 0 data bytes.

Events

This tab allows adding actions to mapping related events. It only appears for receive mappings.
The configured actions for the events are executed directly in the CAN thread when the event happens.
The CAN stacks will wait until the actions are executed.

OnReceiveTimeOutAction

This event is triggered if a receive mapping is not received within the period defined as "Transmission Period". If this happens all the variables mapped to the PGN are automatically considered invalid and additionally the action is executed.
If you should see any "Event Options" for this event - ignore them like the PClient does.
Possible actions that can be assigned are
- "No Action" which is the default
- "Set Value" to assign a constant value to a variable, e.g. to reset a counter, raise an alarm or trigger something attached to a variable where "Force Writing" is true.
- "Execute Script" allows to execute a script and provides additional parameter "Execute on every Xth Event" so that it is possible to execute the script only for e.g. every third timeout.

OnReceiveTimeOutResolvedAction

This event is triggered if a receive mapping which time out before is received again. If this happens all the variables mapped to the PGN are automatically considered valid again (assuming the PGN received really contained a value for them) and additionally the action is executed.
If you should see any "Event Options" for this event - ignore them like the PClient does.
Possible actions that can be assigned are
- "No Action" which is the default

- "Set Value" to assign a constant value to a variable, e.g. to reset a counter, raise an alarm or trigger something attached to a variable where "Force Writing" is true.
- "Execute Script" allows to execute a script and provides additional parameter "Execute on every Xth Event" so that it is possible to execute the script only for e.g. every third timeout.

Step 4: Add a Variable to the Variable mapping

In the Configure Mapping tab, by pressing on Add, a new variable can be configured into the variable mapping. Alternatively, the Visual CAN Mapping[397] (5) can be used.
Once created, a PDO can be deleted by pressing the Delete button.
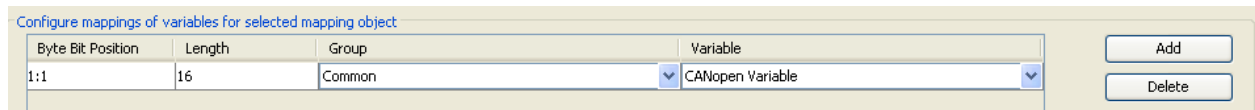
The user selects the group and then a variable. The PDO has a maximum length of 8 Bytes. For Saving the PDO, user click on OK. By pressing on Cancel, all changes will be cancelled.

The assignment of Variable into a PDO is depending on the access type of the variables and the owner of the Variable

For variables which belongs to same owner as PDO or PClient, the following rules are implemented:

- WO, RW can be added into Receive PDOs.
- RO, RW can be added into Transmit PDOs.

Create an EDS (Electronic Data File) file
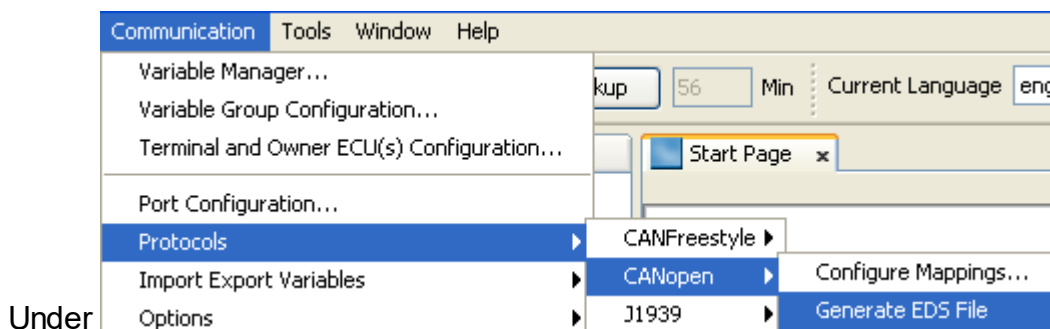
Under

After few seconds depending on the number of PDO and variables, one or several EDS Files (one per CAN instance) will be generated under the directory of the project.

5.3   J1939

General Introduction

J1939 is a CAN protocol defined by SAE (Society of Automotive Engineers) and is the basis for some other standards such as NMEA2000 for GPS data on CAN and ISO 11783 in agricultural machinery. It is also widely used in "off-the-shelf" Diesel engines.

The protocol consists of the following main elements:
- unique identification of each participant on the bus
- standardized message and number format for many automotive use cases
- means to transfer data > 8 bytes (i.e. longer than one CAN message)

These main components are described in the following sections.

Based on these definitions there are more pages describing
- [basic settings to use J1939](#)□345 ,
- [creation of J1939 ECUs and variables](#)□349 and
- [PGN mappings](#)□360

in GUI tool to use J1939 on target display.

CAN Bus Configuration

J1939 is defined to run on a CAN 2.0B network at 250 kBit/s i.e. it uses long (29 bit) CAN identifiers.

Message Format

J1939 describes each signal (i.e. each value to be transmitted) as an **SPN** (Suspect Parameter Number).

Many SPNs are combined into a **PGN** (Parameter Group Number) which gets a unique number and describes what is actually transmitted on CAN bus.

Each PGN can be identified by the CAN identifier which has the following format (source: [Wikipedia](#))

29 bit CAN-Identifier			Data	
28 … 26	25 … 8	7 … 0 bit	0 … 8 byte	
Priority	PGN (Parameter Group Number)	SA (Source Address)	PDU (Protocol Data Unit)	

		PGN		
25	24	23 … 16	15 … 8	
R (Reserved)	DP (Data Page)	PF (PDU Format)	PS (PDU Specific)	

	I.	PDU1 Format (specific) (peer to peer)		
		23 … 16	15 … 8	
		00h … EFh (0 … 239)	DA (Destination Address)	0 … 61439
or	II.	PDU2 Format (global) (broadcast)		
		23 … 16	15 … 8	
		F0h … FFh (240 … 255)	GE (Group Extension)	61440 … 65535

The last byte of this CAN ID always lists the source address of the message i.e. it identifies the ECU which sent this message.

The first three bits are always considered priority based on CAN interpretation i.e. the lower value indicates higher priority.

Located between these two is the 18 bit PGN value consisting of
1. Bit 25 was "reserved" in the past but is meanwhile considered "Extended Data Page" (0 for most predefined PGNs)
2. Bit 24 defines the "data page"

Extended data page bit	Data page bit	Description
0	0	SAE J1939 Page 0 Parameter Groups
0	1	SAE J1939 Page 1 Parameter Groups (NMEA2000 ®)*
1	0	SAE J1939 reserved *
1	1	ISO 15765-3 defined*

*not supported by our devices

3. Bits 23 ... 16 define the "PDU Format": If "PDU format" is less than 240 then "PDU specific" is 0 in the PGN definition and is replaced by the destination address in the CAN identifier (PDU1). If "PDU format" is greater or equal to 240 then "PDU specific" just extends the PGN number.
4. Bits 15 ... 8 are "PDU specific" and either define the destination address (PDU1) of the message or just extend the PGN number (PDU2).

Please note that the definitions above limit the number of valid PGNs:
PGN 0x000712 (==1810) is invalid since 0x07 == 7 is less than 240 and therefore the last byte has to be 0.
A valid PGN would be 0x000700.

Another side effect of the above definitions is that all PGNs with "PDU format" >= 240 are broadcasts because there is no room for a destination address in their CAN identifier.

Number Formats

J1939 typically transmits all values as **unsigned** with **offset** in **metric** units e.g. most temperatures are transferred in one byte with -40°C offset and 1°C/bit resolution. This means 50 is sent on CAN to indicate +10°C.
For most number formats the higher range is reserved and in almost every case "**all bits set**" indicates "**invalid**" or "**don't care**". I.e. in the example above one could not transfer temperature 215°C since the respective value on CAN would be 255 == 0xFF where all bits are set.
Value transmission of all standardized PGNs happens in "**little endian**" format i.e. "least significant byte first". Nevertheless some manufacturers use "big endian" format in their proprietary messages.

Address Arbitration

Every ECU on a J1939 network has to claim its own address. The address consists of one byte and either uses predefined suggestion from SAE, e.g. 0x00 for the engine ECU or is dynamically negotiated at runtime.
In static networks addresses can be pre-configured but should nevertheless be claimed. The tool allows to configure ECUs with static addresses where only the address is defined. Alternatively addresses can be claimed dynamically i.e. each ECU claims an address and if there should be conflicts the one with the lower J1939 name wins.
J1939 name is a 64 bit number which is supposed to uniquely identify each ECU (at least on the network they are all connected to). It is sent as the data bytes of the address claim PGN 0x00EE00 == 60928. Since the fields of this 64 bit number are not byte aligned the address claim PGN can be considered a "worst case" scenario for little endian transmission.

Transport Protocol

J1939 transport protocol is used whenever the amount of data exceeds 8 bytes so that more than one CAN message is required to transfer them. J1939 transport protocol supports transferring up to 1785 bytes.
Our program automatically handles the transport protocol but requires all data transferred to be byte aligned in this case.

5.3.1   J1939 Basic Settings

This page is a follow-up of the general introduction into J1939[342] .
Here the general setup required to use J1939 is described.
There are two more pages describing setup of J1939 ECU and variable definitions[349] and
J1939 PGN mappings[360] .

Adding J1939 to the project

Before one can use J1939 protocol it has to be activated. A good starting point is to
configure the hardware interface to be used to 250 kBit/s:
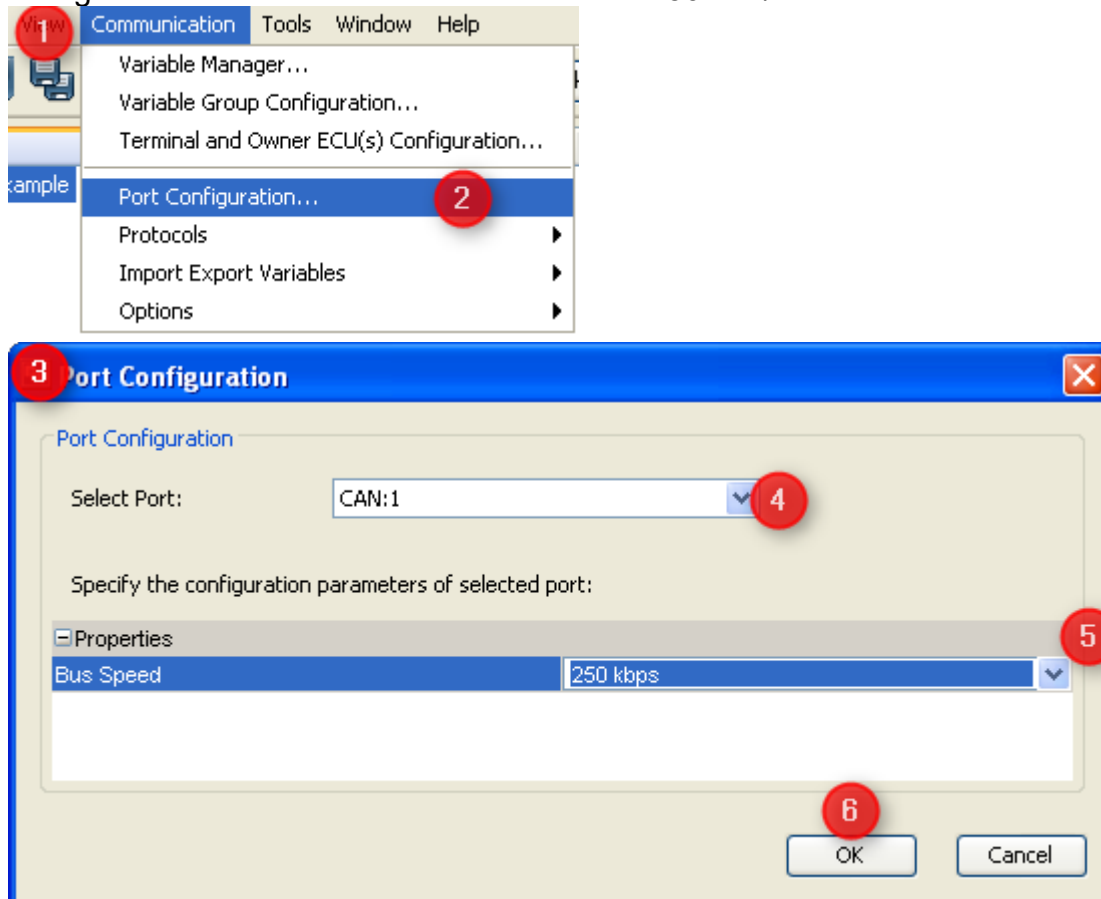
1. Open "Communication" menu
2. Choose "Port Configuration ..."
3. The "Port Configuration" dialog will appear
4. Select the CAN port you want to use for J1939
5. Set the bus speed (the standard baudrate for J1939 is 250 kBit/s)
6. Press "OK" to confirm the setting

Once the port is configured the protocol itself has to be assigned to it:
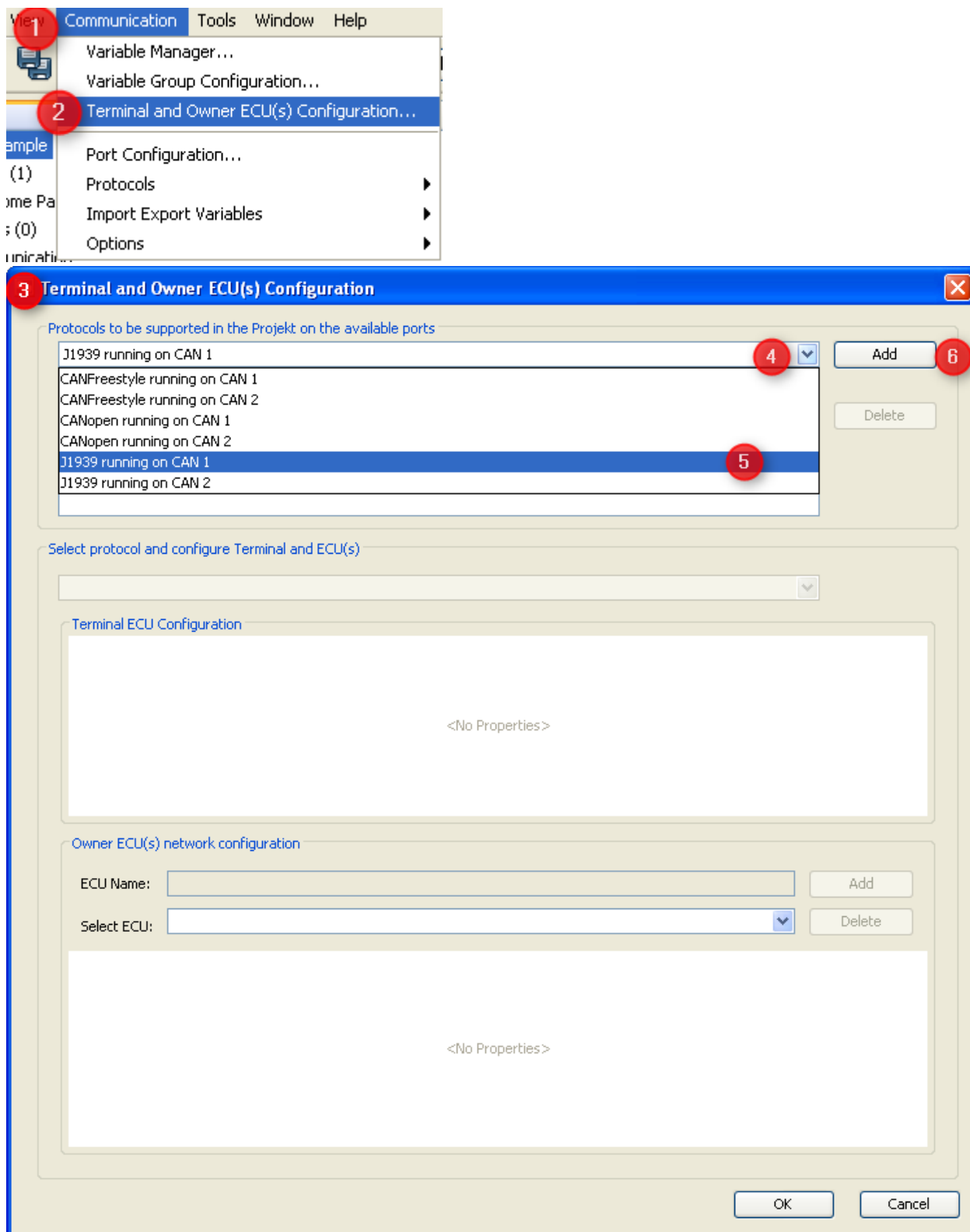
1. Open "Communication" menu
2. Choose "Terminal and Owner ECU(s) Configuration ..."
3. The "Terminal and Owner ECU(s) Configuration" dialog will appear
4. Press on the arrow to see the protocols available
5. Select J1939 protocol for the CAN port you want it to run on
6. Press "Add" to use this protocol with your GUI on the display

Defining Display's J1939 Address

Knowing that every ECU on the bus has to claim an address we now have to configure the display's own address in the same dialog:
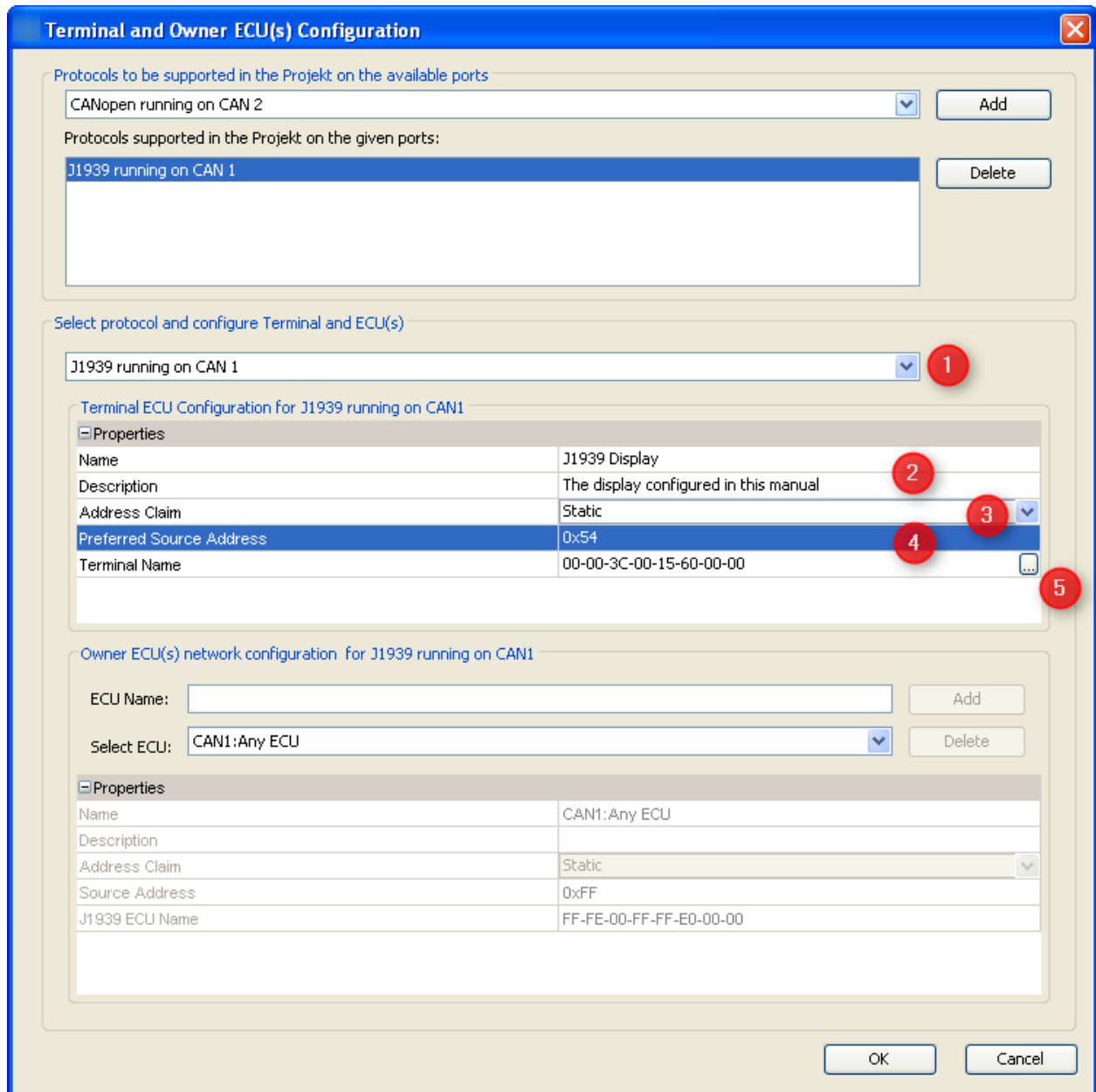
1. Select J1939 on the CAN port configured before - in our example we configured it for CAN 1.
2. Type a valid name and description - these are mainly for your convenience and may help others to understand the GUI configuration in future.
3. Choose "**static**" if you know all ECUs on your network in advance and want the display to always claim the preferred address (or none if another ECU with a higher priority claims the same address). Choose "**dynamic**" if the display should only start with the preferred address but is allowed to use another address.
4. Select a "**preferred source address**". For static address claiming this will be the display's address. For dynamic address claim the display will try to claim this

address but use another one if forced to do so by an ECU with higher priority. The default 0x54 == 84 is reserved for "Cab Display #2" in the main J1939 document.

5. Configure the display's J1939 name after pressing the button with the three dots.

Configure Display's J1939 name

The display claims a J1939 name at startup and whenever asked for it. To do so it needs a J1939 name independent of whether address claim is configured as "static" or "dynamic". Pressing the button with the three dots in the above dialog opens the following dialog where the name can be configured:

This dialog is divided into three sections all showing the same J1939 in different formats:

- On top one can edit the sections of the name as defined in J1939-81. This sections allows entering values in decimal (displayed in tooltip) and hexadecimal format (displayed).

- The second section shows the bytes as they will appear in the display's address claim message on CAN bus. This section can only be edited in hexadecimal format.
- The lower section shows the resulting J1939 as 64 bit number. Like the section above this one can only be edited in hexadecimal format.

In the following we describe the J1939 name configuration based on the fields as shown in the upper section. If desired it could also be done in the other sections and all three will always be synchronized.

1. The "Arbitrary Address Capable" bit is automatically set according to your selection of "static" or "dynamic" address claim in the dialog before.
2. "Industry Group" 0 is applicable to all industries. More specific groups can be found in table B1 of the main J1939 document.
3. "Vehicle System Instance" is defined in J1939-81 as "Vehicle System Instance is a 4-bit field that is used to identify a particular occurrence of a particular Vehicle System within a connected network." and 0 refers to the first instance.
4. "Vehicle System" could e.g. identify "tractor" or "trailer"; 0 refers to "Non-specific System" according to main J1939 document.
5. "Function" 0x3c == 60 refers to "Cab Display". Other options can be found in table B11 and B12 of the main J1939 document.
6. "Function Instance" differentiates ECUs implementing the same function. 0 refers to the first instance.
7. "ECU Instance" allows separating different ECUs belonging to the same function and function instance. 0 refers to the first ECU instance.
8. "Manufacturer Code" has a default value. Other codes can be found in table B10 of the main J1939 document.
9. "Identity Number" should allow differentiating two similar devices connected to the same network. One can keep this as "0x00" since the display will automatically fill it with a 21 bit number calculated from device part number and serial number.

Finally press "OK" to confirm the setting and close the dialog.

5.3.2   J1939 ECUs and Variables

This page is a follow-up of the general introduction into J1939[342] and the general setup required to use J1939[345] .
Here we create J1939 ECUs and their variables.
There is one more page describing setup of J1939 PGN mappings[360] .

Defining other ECUs on the J1939 Network

The display has to identify every ECU it should communicate with.
- In a static network the source address of each ECU is sufficient for this purpose.
- For networks using dynamic address claim each ECU has to be identified by its J1939 name.

The main purpose of the ECU configuration is to use these ECUs as variable owners.
Note: The display will request an address claim if it finds an unknown source address on the bus. If there should be an ECU that does not answer this request, it should be configured here even if it does not otherwise communicate with the display.

Creating an ECU with static Address

ECUs are configured in the same "Terminal and Owner ECU(s) Configuration" dialog that was used above to activate J1939 protocol and to configure the display's address.

ECUs are configured in the lower part of this dialog:
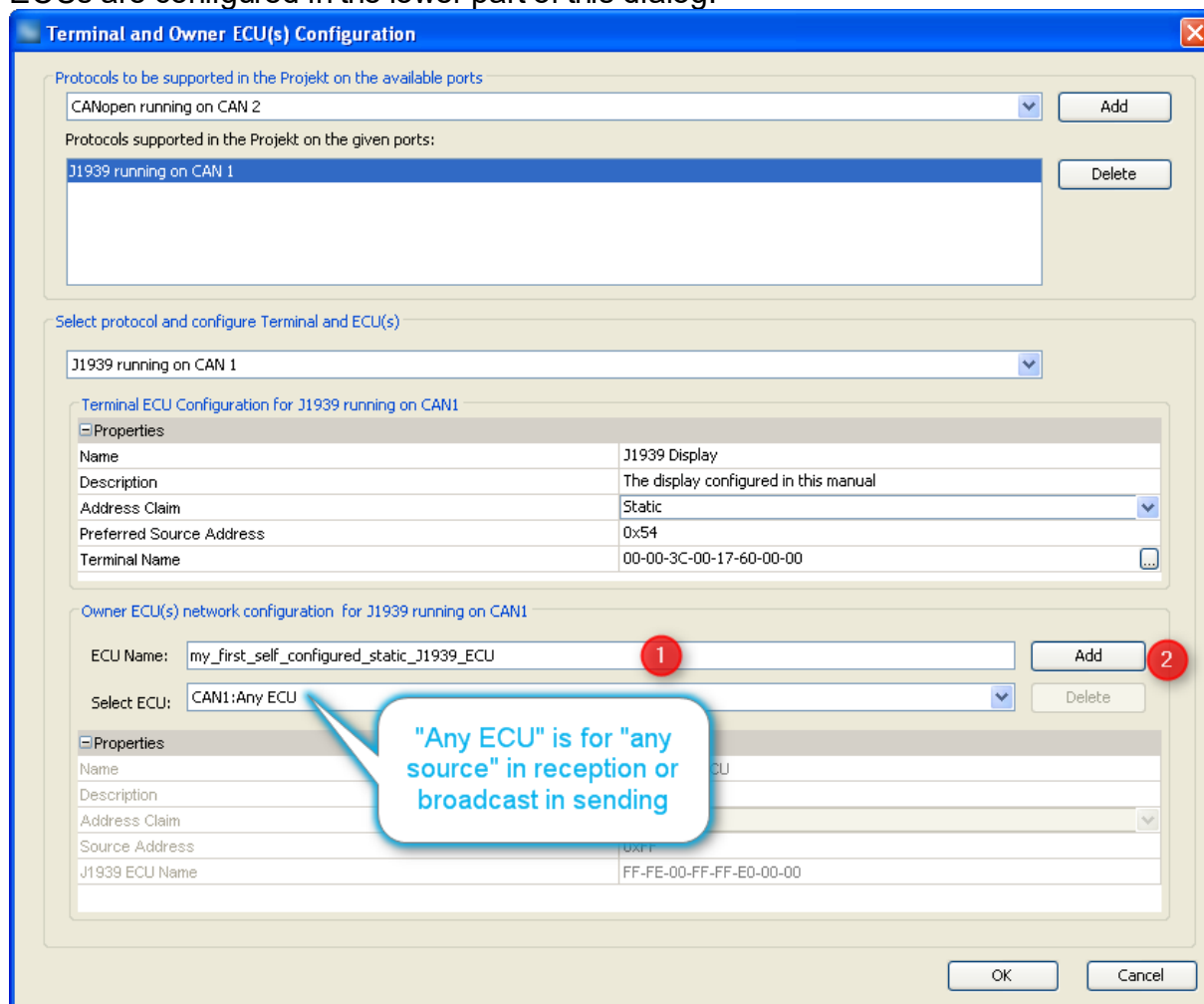
1. Type a name describing the ECU
2. Press "Add"

Then the new ECU will automatically be selected and one can modify the other parameters:
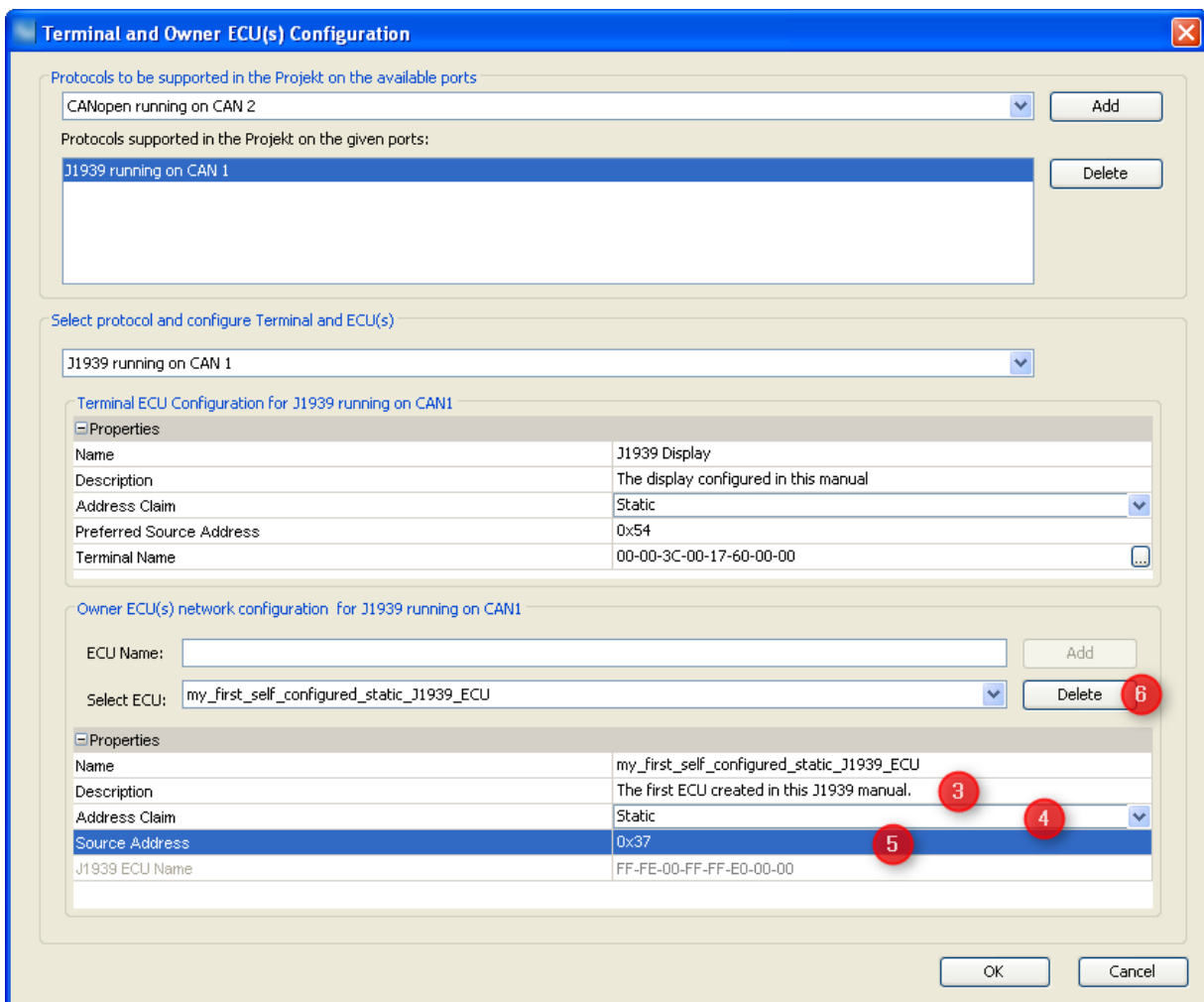
3. Provide a description that could help others to understand better which ECU is meant here.
4. Select "Static" since we want to configure an ECU with a static address here (ECUs with dynamic addresses are handled below).
5. Type the source address this ECU will use. This can be done in decimal or hexadecimal format. Anyways hexadecimal is shown.
6. In case the ECU is not needed anymore it can be deleted again. If it should already own any variable all these will be owned by the display afterwards.

Creating an ECU with dynamic Address

The first steps are the same as for the ECU with static address described above - with the exception of the "Address Claim" field which has to be set to "dynamic" and then the "J1939 ECU Name" field becomes unlocked:
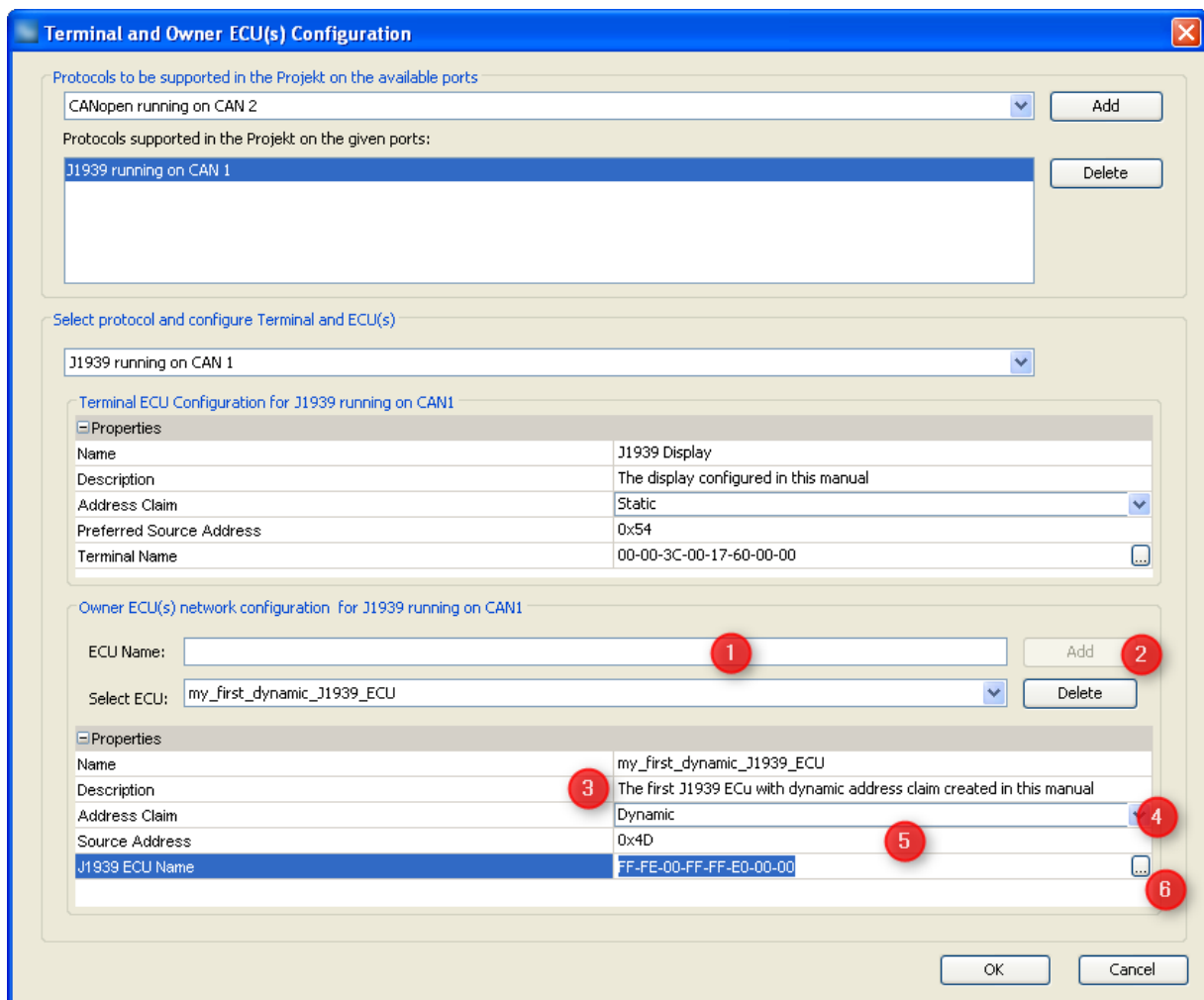
1. ... 3. Are the same steps as above.
2. Select "Dynamic" since we want to configure an ECU with dynamic address claim here
3. The "Source Address" chosen here is not really relevant - the display will identify the ECU by its J1939 name. Anyways it's a good idea to configure the preferred address of the ECU here.
4. Press the ". . ." button in the J1939 ECU Name field in order to configure the ECU's J1939 name. This will open a dialog similar to the one used to configure the display's name earlier:
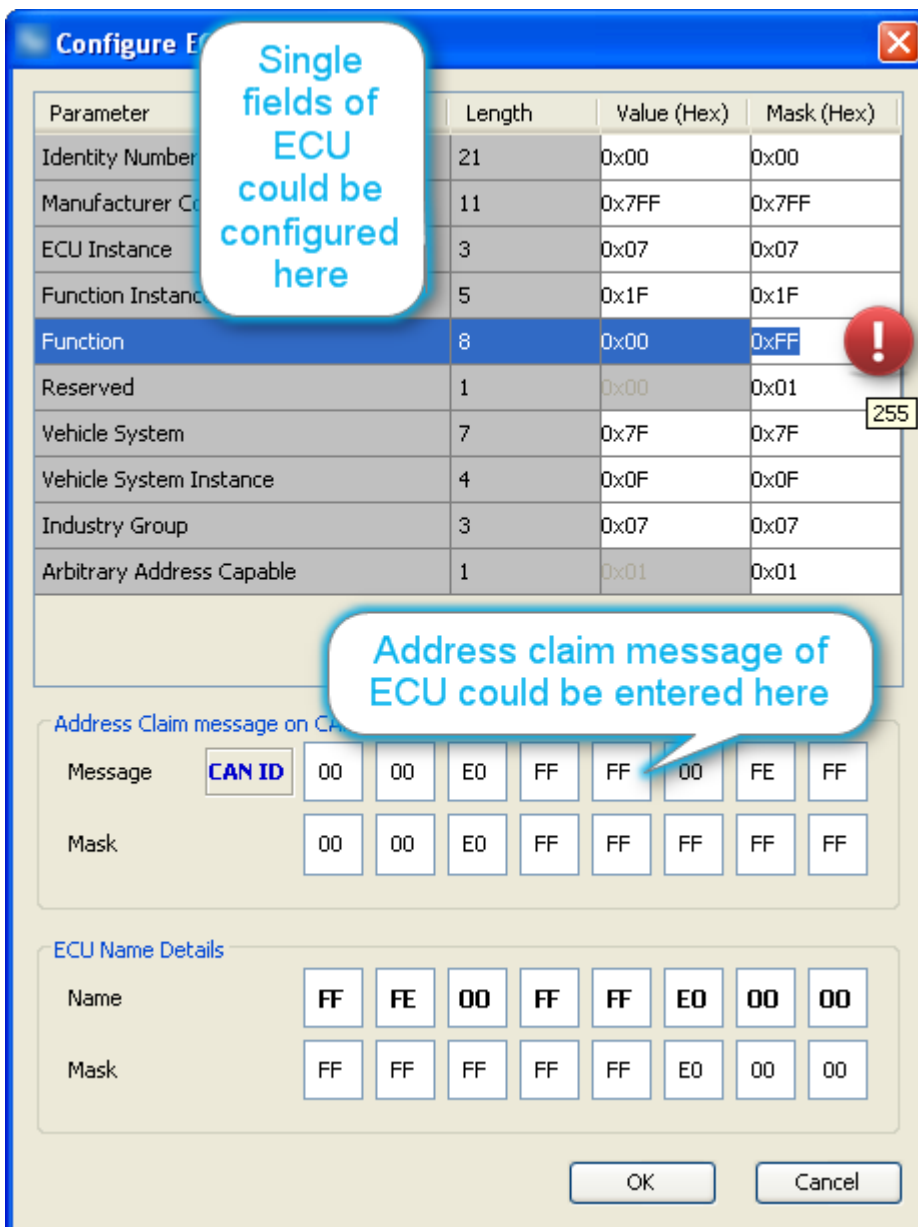
The fields of the J1939 name are explained here[348] .

One could find out the name of an ECU with a CAN tool the name is the data part of the address claim message looking like this in a CAN tool:

18EAFFFE 3 00 EE 00
18EEFF4D 8 23 47 E0 FF FF 00 FE FF

The first line shows a "request for address claimed" which you could send from your CAN tool to get the response in the second line.

This second PGN 0x00EE00 is the address claim coming from ECU with source address 0x4D and having the name we have as default in the dialog.

The most convenient way to enter this name obtained with a CAN tool is provided in the center of the dialog.

Note: In our CAN example the bytes belonging to the "Identity Number" field are "23 47" whereas the tool default is "00 00" because we expect them to be ignored anyways.

Setting Masks for the J1939 Name of an ECU

The message above shows the J1939 name of one specific ECU. The challenge now is to decide which parts of the name are relevant for the display to recognize this ECU on the bus.

Masks are used to ignore those parts of the J1939 name which change from ECU instance to ECU instance while evaluating the rest. Masks are applied via a bit-wise AND so that a bit set to 0 on the masks means to ignore the respective bit of the name.

ECU is recognized as the one configured in the dialog

```
    if ( (ECU_name_on_CAN AND configured_mask) ==
  (configured_ECU_name AND configured_mask) )
```

By default every bit of the configured name is considered - with the exception of the "Identity Number" field which should be derived from the ECU's serial number.

Note: It seems the default mask for "Function" is 0x01 and not 0xFF as it should be.

Adding Variables to the J1939 ECUs

As mentioned above we mainly configured the J1939 ECUs because they own some variables whose values we want to show and/or modify on display.
Before we can use these variables we have to do two things:
  1. Create the variables and use one of the J1939 ECUs as owner.
  2. Define the PGNs we want to use to send and receive the variables.

Creating dedicated variable groups for variables owned by J1939 ECUs

Knowing that we could end up with a lot of variables per ECU it's a good idea to create one variable group per ECU. Of course you are free to organize variables in groups however you prefer.
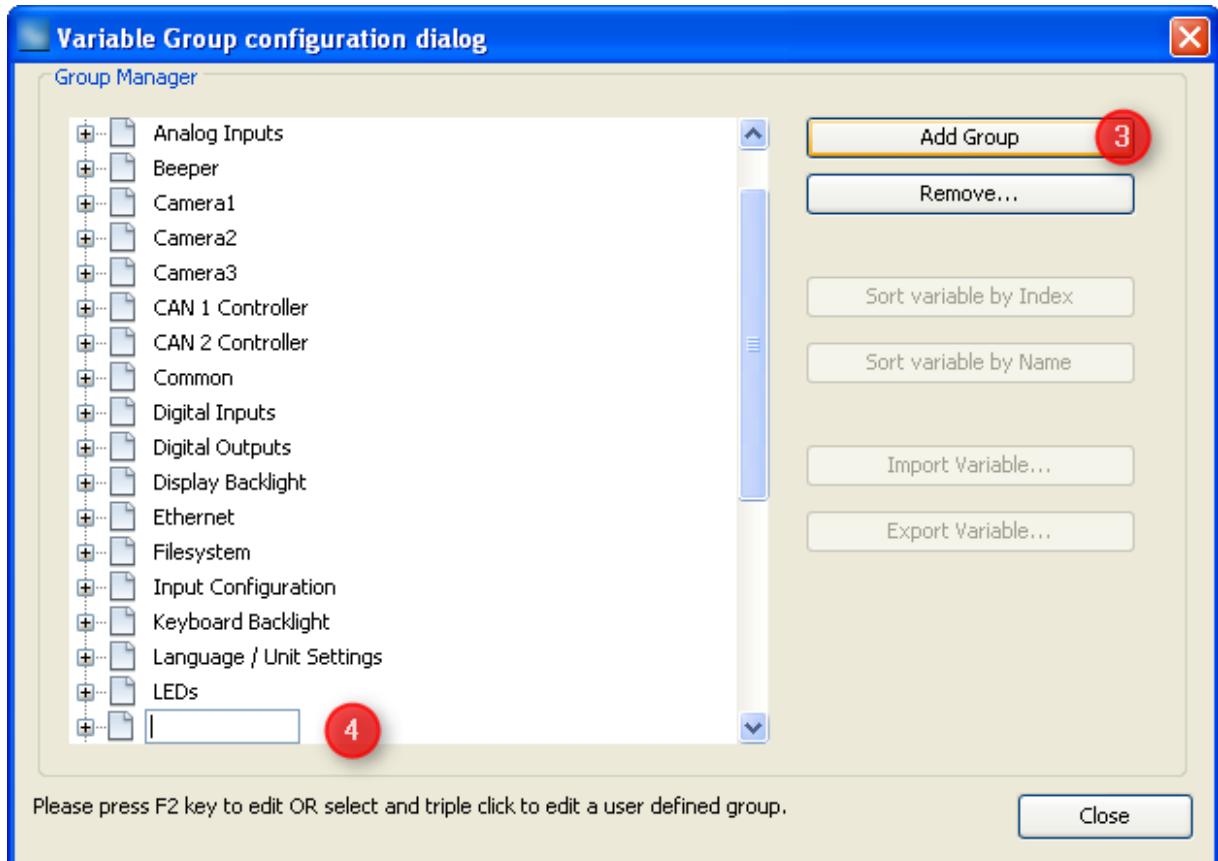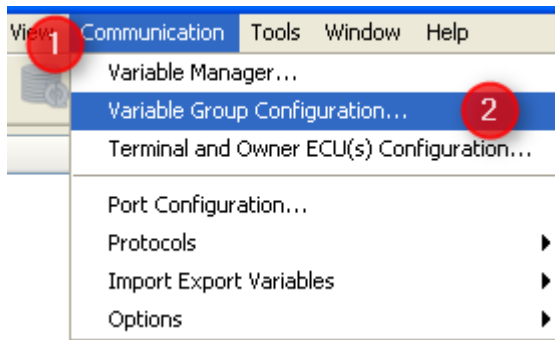
1. Open Menu "Communication"
2. Select "Variable Group Configuration ..."
3. In the dialog that opens press "Add Group" to create a new variable group
4. Type a sensible name for the new group
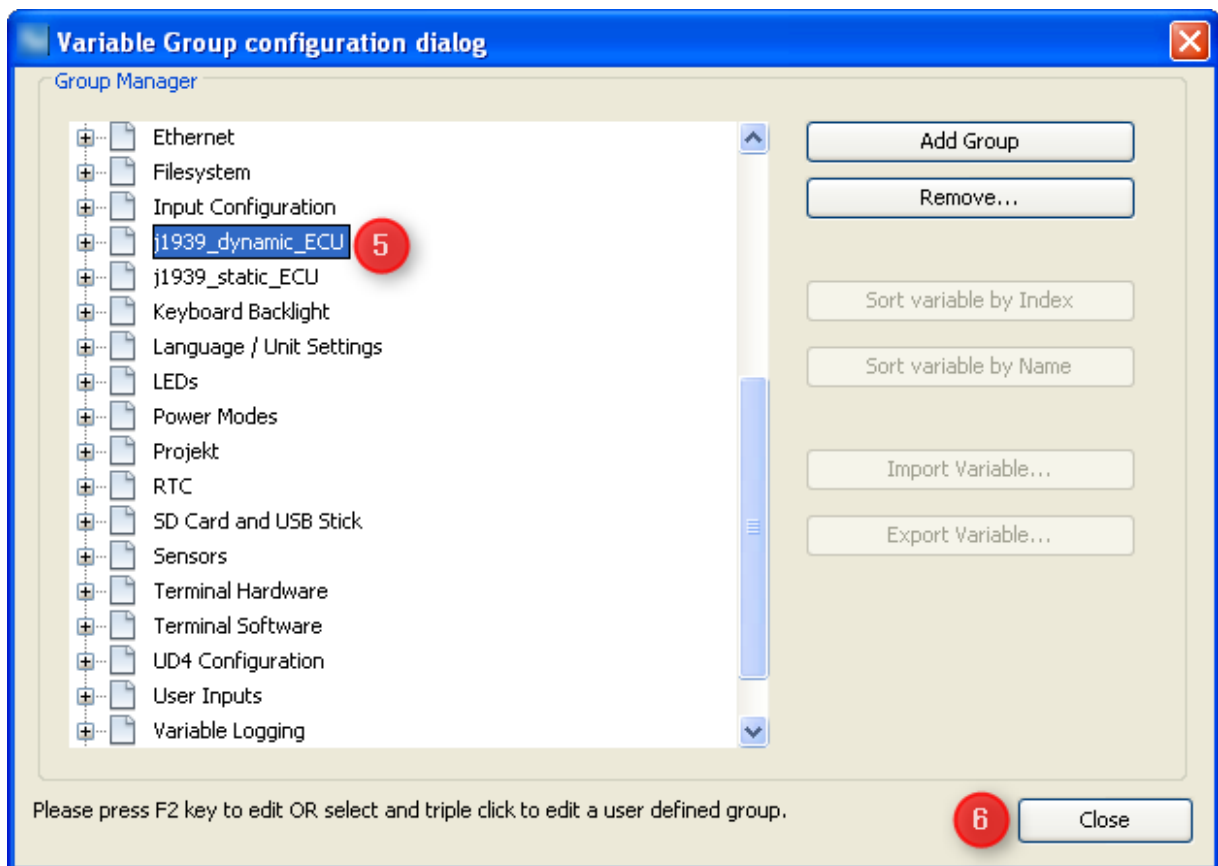
5. Once you pressed "Enter" to confirm the new group name it will move according to alphabetical order of all groups.
6. Press "close" once you're finished creating or deleting variable groups.

If you delete a variable group that is not empty, all the remaining variables of the deleted group will be moved to group "Common" which is the default for all user defined variables.

Creating J1939 variables

Creation of J1939 variables is similar to all other variables:
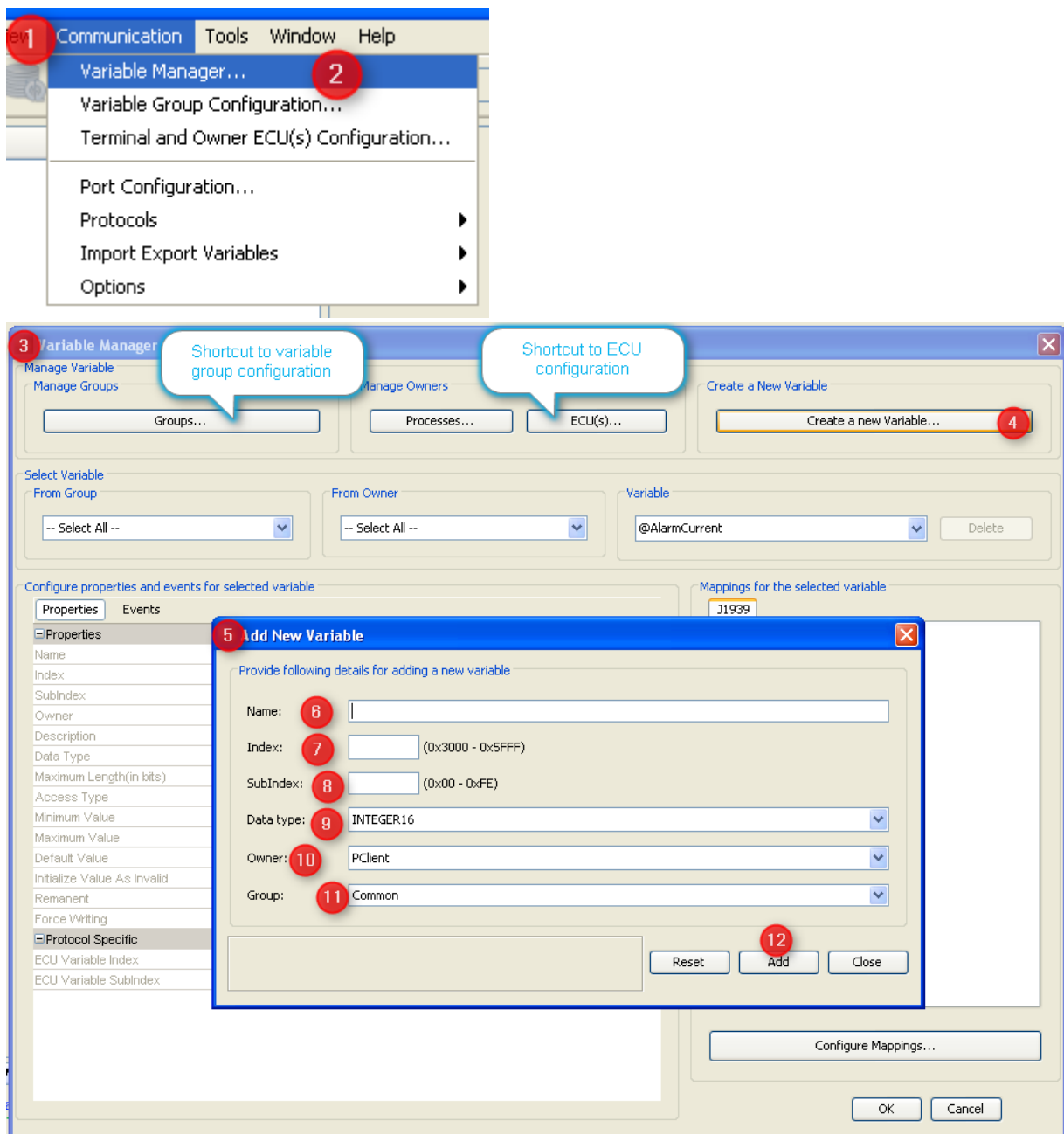
1. Open the "Communication" menu
2. Choose "Variable Manager ..." entry
3. Variable Manager dialog opens (which can be used as shortcut to all variable related tasks)
4. Press "Create a new Variable ..." button
5. "Add new variable" dialog opens
6. Fill in a name of the variable
7. Type an index for the variable. You are free to enter the value in decimal or hexadecimal, starting with "0x". Anyways hexadecimal will be displayed. (Although the concept comes from CANopen you should use index and subindex as a means of ordering and grouping of variables)
8. Add a subindex for the new variable. Note: Combination of index and subindex has to be unique among all variables in the GUI definition.

9. Select signed or unsigned data type at least as wide as the variable will be in CAN message.
10. Change owner from default "PClient" to the newly created J1939 ECU. Note: If you keep "PClient" you could handle the variable also but would not notice if the ECU is not available anymore.
11. Select the newly created variable group for the variable.
12. Press "Add" to finally create the new variable with the settings made above.

Once you did all that the dialog could look like this:

Now you can either create more variables by editing the settings with or without "Reset" or leave the dialog with "Close".

After closing the "Add New Variable" dialog you can modify all the properties of the variable(s) just created or already existing in "Variable Manager" dialog:
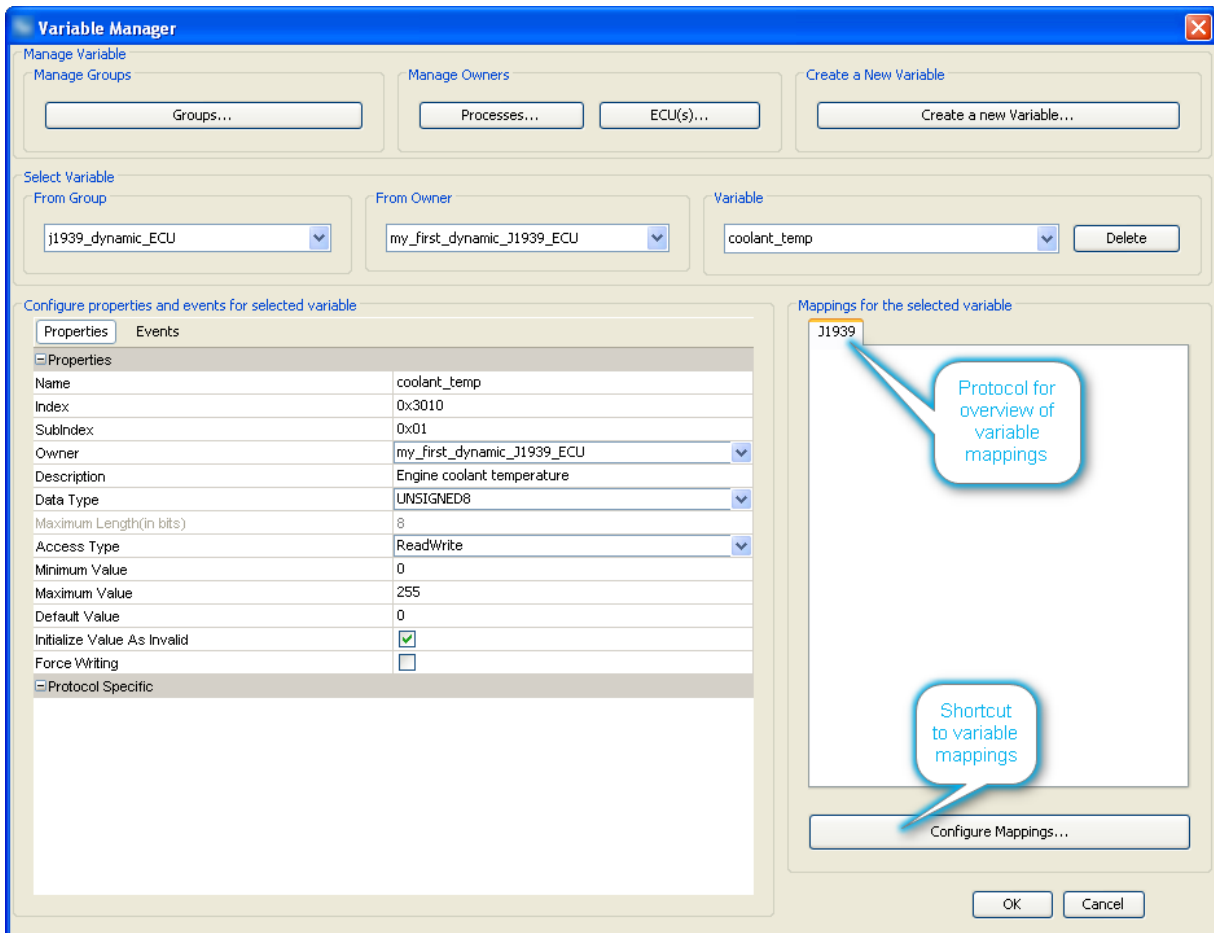
As you may have noticed the "Description" was added and the variable type was modified from "INTEGER8" to "UNSIGNED8".

For a detailed description of all the variable properties and events please refer to variables menu description [170] .

There is a second option to create variables:
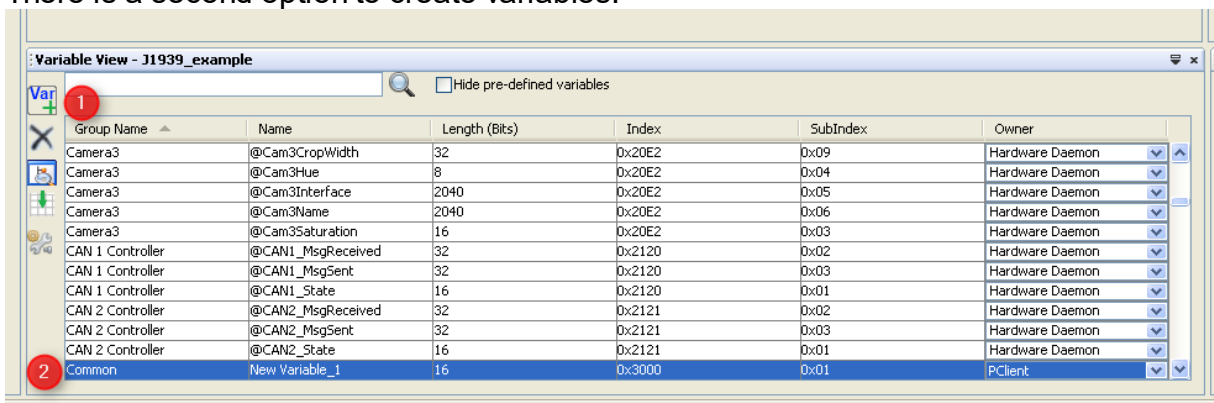
1. In variable view press the "Var +" button
2. A new variable with default settings appears
3. You can double click each property and edit it.

Warning: If you type a group name not defined yet, a new variable group will be created.

> **Warning:** Depending on your sorting option the variable can "disappear" whenever a modification is confirmed.
> More details on this option can be found in [variable view](#)[146] .
>
> So the recommendation is to create variables following the first option and use the variable view to manage and modify existing variables.

5.3.3 J1939 PGN Mappings

This page is a follow-up of the general introduction into [J1939](#)[342] , the [general setup required to use J1939](#)[345] and the [creation of J1939 ECUs and their variables](#)[349] .
Here we create J1939 PGN mappings to communicate with ECUs over CAN bus.

 Tutorial video about setting up J1939 transmit mappings
https://www.youtube.com/watch?v=AzjhoZl9y8w

 Tutorial video about setting up J1939 receive mappings
https://www.youtube.com/watch?v=GUtjp9NrCDQ

Defining PGNs/Variable Mappings

After having defined an ECU as variable owner and the variable itself we finally have to define the connection between these two. In other words: We need a definition about how to exchange the variable's value with the ECU.
These definitions are called "variable mappings" within this tool and for J1939 protocol the variable mapping defines the PGN.
In order to define a PGN or J1939 variable mapping you can either use the shortcut button of the variable manager shown in the image above or open the respective dialog from the menu as follows or, alternatively, click the J1939 mapping shortcut inthe toolbar:
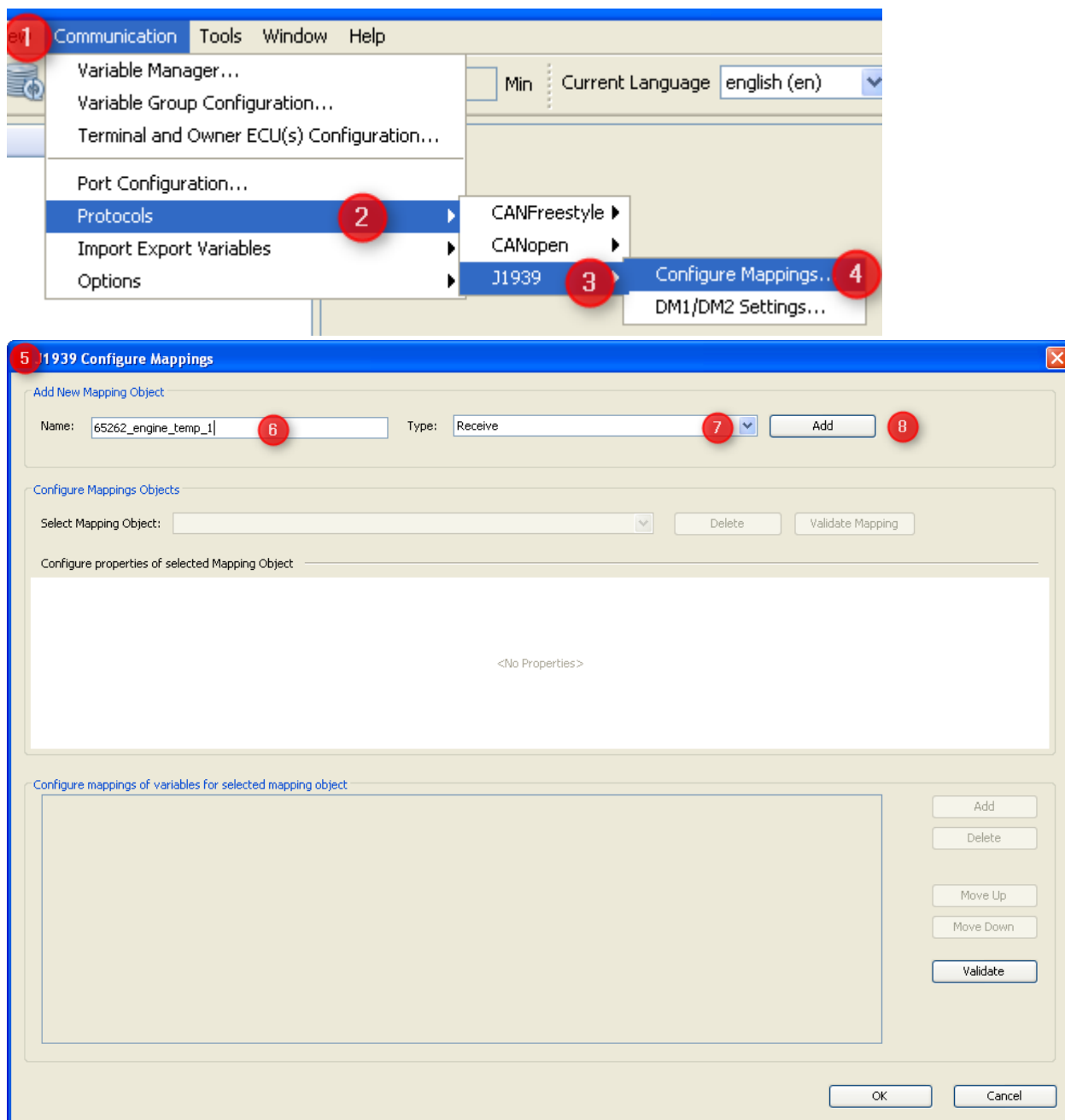
1. Open menu "communication".
2. Select "Protocols".
3. Choose "J1939"
4. Select "Configure Mappings ...".
5. The "J1939 Configure Mappings" dialog will pop up.
6. Type the name of your PGN mapping - in the example we want to define PGN 65262 "Engine Temperature 1" defined in J1939-71
7. Select type "Receive" if the display should fetch values from CAN bus and "Transmit" if the display should send out values.
8. Press "Add" to create the new mapping.

Setting Mapping Properties

The dialog will populate with the default setting for the new mapping object which have to be modified according to your intentions:
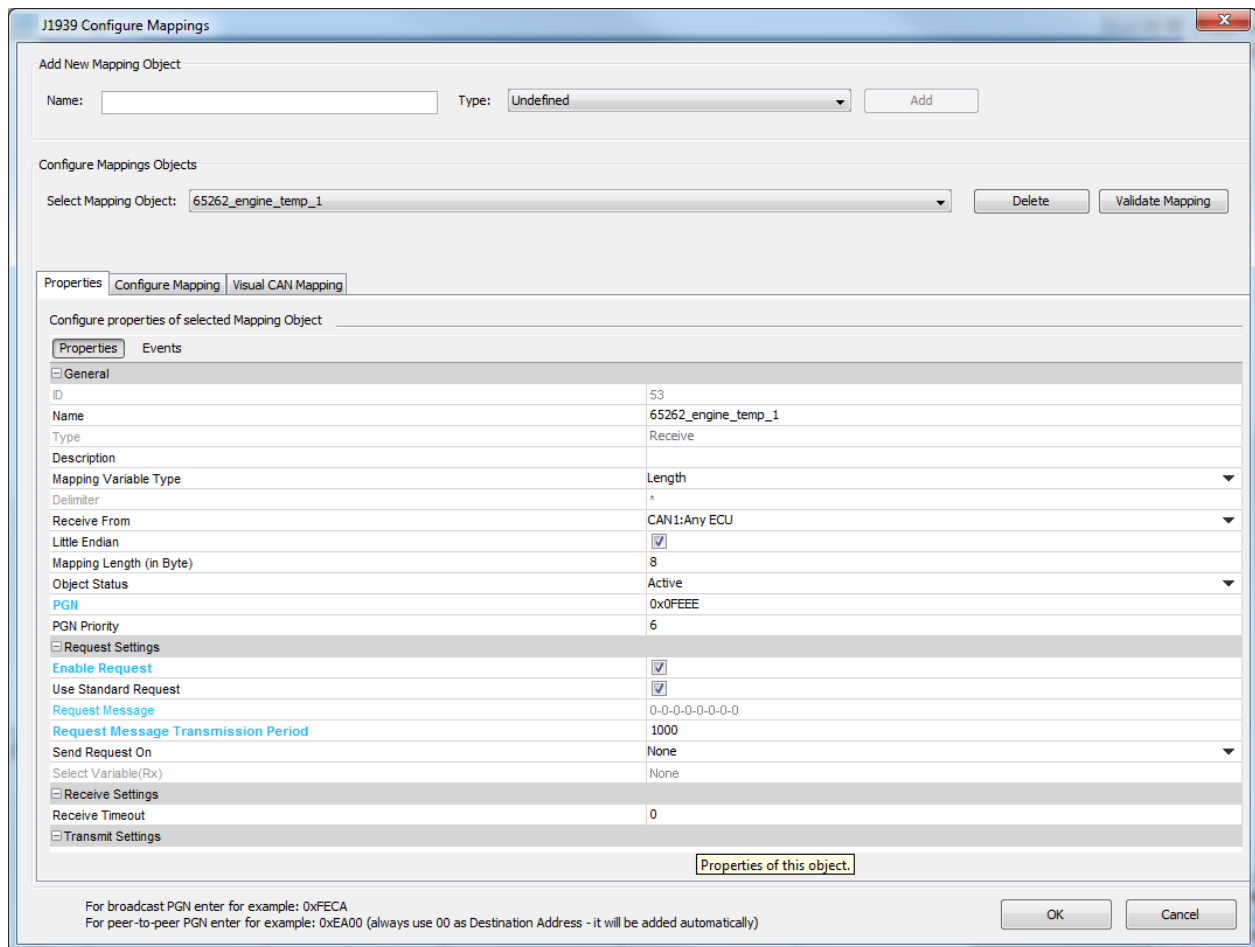
General

ID

Is the object ID internally created which can not be modified.

Name

The name of the mapping as used during creation. This name can be edited here.

Type

"Receive" or "Transmit" - this property can not be modified. If this needs to be changed you have to delete and recreate the mapping.

Description

Should be a valid description of the mapping which helps understanding its purpose and context in future.

Mapping

Variable Type

There are two types of mappings in this context:

- **Length** based mappings find every variable (or constant) by position and length within the CAN message. If length based mappings are less than 8 bytes long their variables or constants can be placed at any position. If they should be longer the variables and constants have to be byte aligned.
- **Delimiter** based mappings look for a dedicated separator character placed between the variables and find the variable based on count of these delimiters. Delimiter based mappings are typically bigger than one CAN message and have to be byte aligned. If a string variable is mapped the value is fully written to the variable even if it should be longer than the length defined for the variable.

Delimiter

Only gets enabled if "Delimiter" is chosen in "Mapping Variable Type" above. Then you can set the separator char to be found in the message.

Receive from / Transmit to

For receive mappings/PGNs: Select the ECU who has to send the PGN where the variable should be extracted. "Any ECU" is a wildcard meaning "don't care for the source address and accept this PGN from anyone".
For transmit mappings this property describes the destination address to be used in the PGN. Please note that some PGNs are broadcasts by definition so that this property will be ignored for them. See also [message format](#)[342] .

Little Endian

J1939 values are transferred least significant byte first (little endian) by default. Since some manufacturers opt for big endian format in their proprietary J1939 messages you can configure this here for the whole mapping. If you should have a PGN which contains both types you have to create two separate mappings.
This setting only applies to numeric values. Strings are expected to be "big endian" i.e. the first byte of the word or sentence is the first one on the bus.

Mapping Length

Enter the data length of the message you want to configure. The length has to be between 0 and 1785 bytes. Between 0 and 8 bytes, the data length of the mapping and the incoming message are compared. Only if they match, the data will be put into the mapped variables.

Object Status

With Object Status Active the mapping is working, i.e. it is processing received messages or sending them. With Object Status Inactive the mapping is not working. This property can be switched at run time with the [setProperty](#)[538] JavaScript function to enable / disable CAN mappings.

PGN

Enter the PGN number in decimal of hexadecimal format (starting with "0x"). The value will anyways be displayed in hexadecimal format. In our example PGN 65262 is displayed as 0xFEEE.

PGN Priority

For receive mappings this is ignored. For transmit mappings this priority is used when sending the message. Priority can be any value between 0 (highest) and 7 (lowest). See also [message format](#)[342] .

Request Settings

These settings define if and when the display sends a request (for receive mappings) or answers a request with a transmit mapping.
In any case the display answers the "request for PGN" if it has a transmit mapping configured for the requested PGN. The only pre-requisite is that the ECU sending the request is "known" i.e. it claimed an address before or is listed as owner ECU in the GUI definition.

Enable Request

For receive mappings this means that the display will send out a request to get the PGN based on the following settings.
For transmit mappings this means that the request defined in the following properties will be recognized and answered with the transmit mapping. Enabling this is only required if there is a request to be recognized that is NOT the PGN 0x00EA00 == 59904 == "request for PGN" as defined in J1939 standard.

Use standard request

Only appears for receive mappings. If enabled the display will send out a "request for PGN" based on the following triggers.

Request message

Is only enabled for receive mappings where "use standard request" is not set. This allows definition of a PGN with fixed data section which will be sent out in order to get the receive mapping based on the triggers below. Depending on the "Receive from" setting this PGN will be sent to the ECU who should provide the mapping or as a broadcast. One can select all three proprietary types of PGNs for this request.
Note: PGNs 0xff00 ... 0xffff are broadcast by definition. See also [message format](#)[342] .

Request Message Transmission Period

Is only enabled for receive mappings. This defines the interval in milliseconds the display will send requests for the mapping. Set to 0 if no periodic request is required.

Send Request On

- "Specific Variable Change" allows triggering the request based on any variable in GUI definition.
- "None" means there is no trigger besides the time based one.

Select Variable (Rx)

Only enabled if this is a receive mapping, request is enabled and "Send Request On" is set to "Specific Variable Change". This allows setting the variable whose value change should trigger sending the request.

Receive Settings

Receive Timeout

This is intended for receive mappings which are expected to arrive periodically - either by default or by periodic request. If the mapping is not received in as many milliseconds as defined here all variable values of the mapping are considered invalid and hence all DDOs showing them will also be disabled.

Transmit Settings

Use as Write Request

If checked this means that the transmit mapping will be sent out once if there is a write request to one of the variables contained in the mapping. Usually the display only sends values provided by the owners of the respective variables in transmit mappings.
This means that e.g. engine speed will only be sent out as provided by engine ECU. If in your system modifying the measured engine speed value is used to set a new desired engine speed then you can check this property and writing on the engine speed variable will send out the transmit mapping containing the new setpoint of the engine speed value. Please note that all other values in the mapping will be the ones provided by their respective owners and even the new setpoint will in the next message be replaced by the last measured value provided by the owner.
If you should want to write to the same variable from many sites you should consider "PClient" as the variable owner which will accept all write requests but whose variables will never become invalid again once they have a value.

Send Value On

- **None:** Mapping is only sent time based (if configured) and on request.
- **Any Variable Change:** Whenever one of the mapped variables changes its value the mapping is sent out.
- **Specific Variable Change:** The mapping is sent out if the variable specified in the following property changes its value.

Select Variable (Tx)

Choose the variable whose value change should trigger sending out the mapping. This variable can be part of the mapping but doesn't have to be.

Transmission Period (in ms)

Defines the interval between messages for this transmit mapping. Set to 0 if time based sending is not desired.

Events

This tab allows adding actions to mapping related events. It only appears for receive mappings.
The configured actions for the events are executed directly in the CAN thread when the event happens.
The CAN stacks will wait until the actions are executed.

OnReceiveTimeOutAction

This event is triggered if a receive mapping is not received within the period defined as "Transmission Period". If this happens all the variables mapped to the PGN are automatically considered invalid and additionally the action is executed.
If you should see any "Event Options" for this event - ignore them like the PClient does. Possible actions that can be assigned are
- "No Action" which is the default
- "Set Value" to assign a constant value to a variable, e.g. to reset a counter, raise an alarm or trigger something attached to a variable where "Force Writing" is true.

- "Execute Script" allows to execute a script and provides additional parameter "Execute on every Xth Event" so that it is possible to execute the script only for e.g. every third timeout.

OnReceiveTimeOutResolvedAction

This event is triggered if a receive mapping which time out before is received again. If this happens all the variables mapped to the PGN are automatically considered valid again (assuming the PGN received really contained a value for them) and additionally the action is executed.
If you should see any "Event Options" for this event - ignore them like the PClient does.
Possible actions that can be assigned are
- "No Action" which is the default
- "Set Value" to assign a constant value to a variable, e.g. to reset a counter, raise an alarm or trigger something attached to a variable where "Force Writing" is true.
- "Execute Script" allows to execute a script and provides additional parameter "Execute on every Xth Event" so that it is possible to execute the script only for e.g. every third timeout.

OnMappingReceived

The action for this event is executed EVERY TIME the mapping is received (regardles of value changes).
When a script action is executed, the variables used in the mapping will already have been updated with the values from the CAN message.
Do NOT execute complex scripts at this event if the mapping is received very often!
Monitor the performance on the device with a realistic cycle time of all CAN messages.
Monitor the cycle time of transmit messages (the CAN stack has to wait until the received action is over before taking care of other tasks like sending out transmit mappings).
If you should see any "Event Options" for this event - ignore them like the PClient does.
Possible actions that can be assigned are
- "No Action" which is the default
- "Set Value" to assign a constant value to a variable, e.g. to reset a counter, raise an alarm or trigger something attached to a variable where "Force Writing" is true.
- "Execute Script" allows to execute a script and provides additional parameter "Execute on every Xth Event" so that it is possible to execute the script only for e.g. every third timeout.

Defining Data Section of Mapping PGN

After setting all the general properties of the mapping now the data part has to be defined. Some general rules apply:
- All variable and constant positions define the position of the least significant bit (for little and big endian)
- Receive mappings can have gaps - as long as the length can be determined.
- Receive mappings are recognized if all the following conditions are true:
```
if (  ( (PGN received) == (PGN configured) )
   && ( (ECU source address) == (ECU configured) )
   && ( (message length) == (message length configured) )
   && ( ((constants received >> shift) AND configured mask)
      == ((constants configured >> shift) AND configured mask) ) )
```
Note: It is allowed to leave gaps in the data section of receive mappings.

Note: The last condition allows covering multiplexed PGNs by defining a separate mapping for each value of the multiplexer
Note: Message length is only compared if less than 8 bytes are defined. For messages with flexible size or other reason to ignore length, make sure configure the length of the message to a value >8.

Assume we want to complete the definition of our "Engine Temperatures 1" PGN so that we can get the coolant temperature from it.
We only have to add the definition of the "engine coolant temperature" value. The length of the message is defined in the message length property.
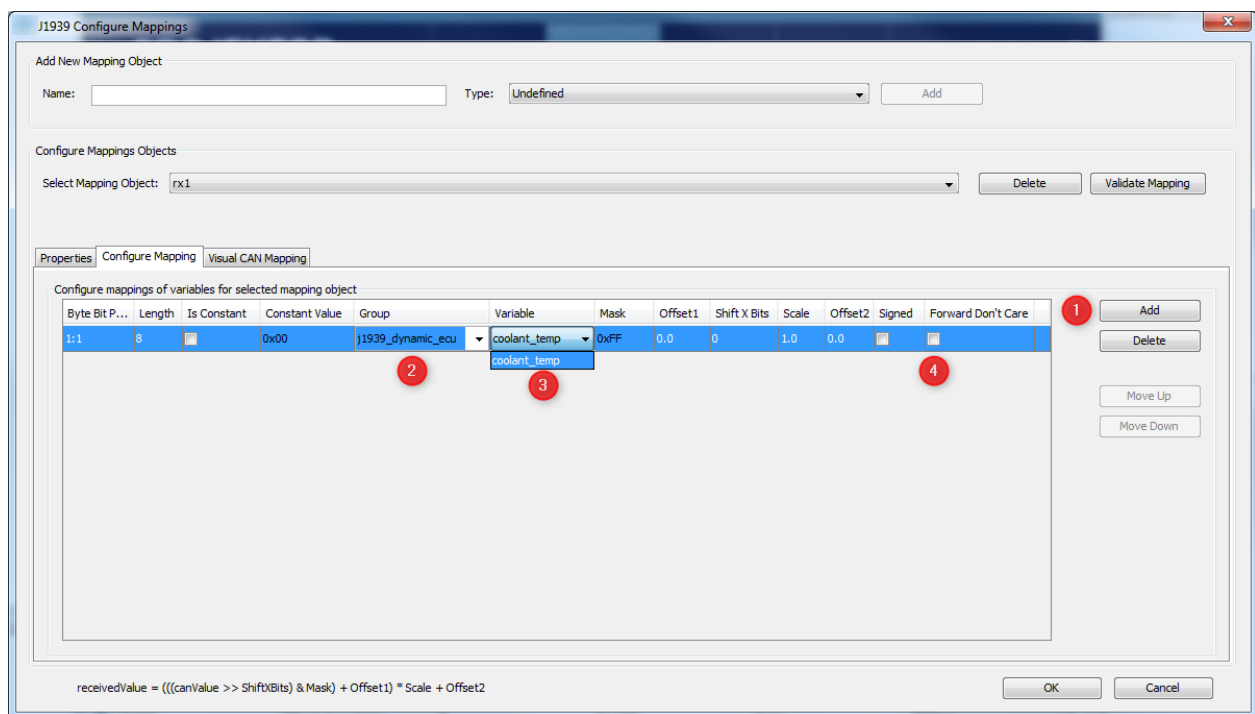
1. Press "Add" button to add a line to the mapping.
2. Select the group of the variable to map.
3. Select the variable itself.

After doing this the properties in the line adjust to the variable definitions made earlier e.g. the length is set to 8 bits and the mask is set to 0xff meaning all the bits will be taken.
The variable mapping can also be done in the Visual CAN Mapping[397] .
Please note, for J1939, if in a variable all bits are set to 1, this is defined as "don't care" in most cases. Therefore, if a received messages contains all 1s for a variable, this variable will not be written with that value. If the variable should still be set with that value of all 1s, activate the option "Forward Don't Care" (4).

J1939 PGN mapping variable properties

The following settings can be made for each variable added to the mapping:

Byte Bit Position

Position is shown as byte#:bit# where byte 1 is the byte closest to the CAN identifier and bit 1 is the least significant bit of a byte. I.e. full CAN message starts with byte 1 directly

behind the CAN ID and ends with byte 8. In a byte with value 1 bit 1 is set, in a byte with value 128 bit 8 is set.
These enumeration scheme matches the one used in J1939 standard documents so that you can directly copy the position. Engine coolant temperature is defined as the first byte - so set start position to 1:1.
Alternatively one can type the bit position directly: 1 refers to the bit closest to the CAN identifier and 64 is the bit at the end of the CAN message.
Warning: This input does not always behave as expected when typing position by hand. Finally the displayed "byte:bit" value is relevant.
The mapping is filled correctly if there are no gaps i.e. the whole message gets filled with values.
Please refer to the following image for byte and bit enumeration within a CAN message. This enumeration is valid for little and big endian variables since it only refers to the CAN message itself.

byte	CAN ID	1	2	3	4	5	6	7	8
bit of byte		8 7 6 5 4 3 2 1	8 7 6 5 4 3 2 1	8 7 6 5 4 3 2 1	8 7 6 5 4 3 2 1	8 7 6 5 4 3 2 1	8 7 6 5 4 3 2 1	8 7 6 5 4 3 2 1	8 7 6 5 4 3 2 1
bit of CAN message		1 2 3 4 5 6 7 8	9 10 11 12 13 14 15 16	17 18 19 20 21 22 23 24	25 26 27 28 29 30 31 32	33 34 35 36 37 38 39 40	41 42 43 44 45 46 47 48	49 50 51 52 53 54 55 56	57 58 59 60 61 62 63 64

Length

Defines the size of the mapped numeric variable or constant in bits. If a string is mapped this defines the length of the string in bytes.
Mapped length can be shorter than the length defined for the variable but not longer. By default the variable length is used.

Is Constant

Enable this checkbox to define a constant. Depending on the mask constants can be used to fill gaps or recognize multiplexers in receive mappings. In transmit mappings constants are required for every part of the message not filled with a variable.
Uncheck this property to map a variable.

Constant Value

Although appearance does not change this property is only valid and editable for constants i.e. lines where "Is Constant" property is set.
If valid this defines the numeric value of the constant. Please note that constants are always expected to be numeric - there is no option to define string constants.

Group

Variable group containing the variable to be mapped. Does not apply to constants.

Variable

The variable to be mapped.

Mask

Variable or constant are bitwise ANDed with mask before they are applied.
See receive condition[360] on how this can be used to receive multiplexed PGNs.
See section mapping calculations[369] for details about how this property applies to receive and transmit mappings.

Offset1

Offset 1 is a decimal value applied to the received value after mask and shift factor but before scale and offset 2.
See section mapping calculations[369] for details about how this property applies to receive and transmit mappings.

Values can be entered in decimal form or in scientific notation / floating-point representation.

Shift X Bits

Shift received value to the right by as many bits as defined here. Negative values shift to the left.
This value is applied after the mask but before offset and scale.
See section mapping calculations[369] for details about how this property applies to receive and transmit mappings.

Scale

Decimal value to scale the received value before applying it to the mapped variable. Is applied after mask, shift and offset 1 but before offset 2.
Values can be entered in decimal form or in scientific notation / floating-point representation.

Offset2

Decimal value applied after all the other calculations are done.
See section mapping calculations[369] for details about how this property applies to receive and transmit mappings.
Values can be entered in decimal form or in scientific notation / floating-point representation.

Mapping calculations

The calculations applied to a mapping are defined mainly for receive mappings - in case of J1939 they can be directly derived from the resolution and offset defined in the standard.
For transmit mappings all calculations are inverted so that the same values can be applied to receive and transmit mappings.
The formula converting a mapped area of a CAN message into the variable value is as follows:

> **variable_value** = ((((**CAN_value >> shift**) & **mask**) + **offset1**) * **scale**) + **offset2**

Consequently the calculation applied to transmit mappings inverts the above:

> **CAN_value** = ((((**variable_value - offset2**) / **scale**) - **offset1**) & **mask**) << **shift**

Note: For constants only "*shift*" and "*mask*" are applied.
Note: Currently signed/unsigned calculation is performed based on variable definition. If variable is signed this means the CAN_value is considered negative if the most significant bit of the mapped area is set. This leads into some issues with J1939 values which are transmitted unsigned but should be displayed in signed format (as described in example below).

Example

Coming back to the example of "Engine Temperatures 1" PGN we would set the following values:

This will correctly provide temperatures from 0 ... 210 °C since we defined the variable as "unsigned". CAN values 0 ... 39 will be converted into high unsigned values by Offset1. Alternatively we could have defined the variable as "signed" so that temperatures of -40 ... 87 °C would be displayed correctly. Then higher values would be interpreted as negative and the offset of -40 would make them even more negative.

It is planned to solve this by defining the CAN format independent of the variable format. Until this works one could either live with the limitations described above or as a workaround use the signed variable type and map one more bit than transmitted and then reduce it with the mask to the correct size. This would require one unused bit next to the variable though.

5.3.4   J1939 DM1 and DM2

 Tutorial video about setting up J1939 DM1 / DM2
https://www.youtube.com/watch?v=IoaWtchS_sY

For receiving and displaying the contents of DM1 and DM2 messages various settings can be made.

Please note that only conversion method 4 is used for the processing of DM1 messages. To cite the standard:
"Effective 1996, Version 4 shall be used for all future OBD applications"
We will focus on products newer than 1996, which is why only version 4 is supported.

To open the settings dialog use the main menu tab **Communication** | **Protocols** | **J1939** | **DM1 / DM2 Settings**.

A description of all available settings follows here:

Common Settings for DM1 and DM2

Select CAN Port

Select the CAN port for which you want to configure the DM settings. Only CAN ports that have a J1939 protocol attached are available. When changing the CAN port you can choose to apply the settings you have made for the current CAN port.

Select Translation File

Using a translation file can make the viewing of the trouble codes easier. In the translation file the following translations can be made:

- ECU Address to ECU Name
- SPN to SPN Description
- FMI to FMI Description (general or SPN specific)

Example content of a translation file (for one language):

```
#ECU address to ECU Name translation
# syntax: ECU;ECUAddress;Translation
ECU;0x00;EMS
ECU;0x03;TECU
ECU;0x0b;EBS
ECU;0x33;TPMS
```

```
#SPN translation
# syntax: SPN;Number;Translation
SPN;110;Coolant Temp.
SPN;1119;Exh. Oxygen
SPN;161;Shaft Speed
SPN;177;Oil Temp.
SPN;521;Pedal Pos.
SPN;1091;Pressure
SPN;258856;Tire F_L
SPN;258857;Tire F_R
SPN;258866;Tire R_F_L
SPN;258876;Tire R_R_L
SPN;258867;Tire R_F_R
SPN;258877;Tire R_R_R

#General FMI codes. Used if no SPN specific FMI was declared
#Syntax: FMI;Number;Translation
FMI;0;Above Normal
FMI;1;Below Normal
FMI;2;Data Erratic
FMI;3;Voltage Above Normal
FMI;4;Voltage Below Normal
FMI;5;Current Above Normal
FMI;6;Current Below Normal
FMI;7;Mech. Sys. Not Responding
FMI;8;Abnormal Frequency
FMI;9;Abnormal Update Rate
FMI;10;Abnormal Rate of Change
FMI;11;Root Cause Unknown
FMI;12;Bad Device
FMI;13;Calibration
FMI;14;Special Instructions
FMI;15;Above Normal (Least)
FMI;16;Above Normal (Moderate)
FMI;17;Below Normal (Least)
FMI;18;Below Normal (Moderate)
FMI;19;Recv. Network Data in Error
FMI;20;Data Drifted High
FMI;21;Data Drifted Low
FMI;31;Condition Exists

#SPN specific FMI codes.
#Syntax: SPNFMI;ECUAddress;SPN;FMI;FMITranslation
SPNFMI;0x0b;521;0;Pedal position too high
```

The Translation File is valid for both DM1 and DM2 messages.

If the project is a multi-language project, other language translations have to be added. Here are some examples for two languages:

```
#SPN translation
```

```
# syntax: SPN;Number;TranslationLanguage0;TranslationLanguage1
SPN;110;Coolant Temp.;Kuehltemperatur
SPN;177;Oil Temp.;Oel Temperatur
SPN;1091;Pressure;Druck

#General FMI codes. Used if no SPN specific FMI was declared
#Syntax: FMI;Number; TranslationLanguage0;TranslationLanguage1
FMI;0;Above Normal;Ueber Normal
FMI;1;Below Normal;Unter Normal

#SPN specific FMI codes.
#Syntax:
SPNFMI;ECUAddress;SPN;FMI;FMITranslationLang0;FMITranslationLan
g1
SPNFMI;0x0b;521;0;Pedal position too high; Pedal Position zu
hoch
```

Please note that the sequence of the languages corresponds to the configured language indices (see Project Properties => Languages[□98])
Please also note that there is a maximum line length the translation file can have. This limit is 32767 characters. If one or more lines exceed this limit, the translation file may not be parsed correctly.
You can use a text file encoded in UTF-8 format to include special characters, chinese, kyrillic characters and so on.

Validation

Upon loading the translation file, or by pressing the Validate button, the translation file will be validated. The result and possible errors will be written to the output window[□39].
Note that you need a translation text for every language you have in your project.

DM1 Settings

Enable DM1 Support

With this checkbox the DM1 support can be enabled or completely disabled.

Log DM1 Messages To File

It is possible to log all DM1 DTCs to a file. Just enable the checkbox and enter a log file name. If you choose .csv as suffix, it is easy to open that log file in Excel or in another spreadsheet application.
Every DTC will be logged in a separate line. A date- and timestamp will be placed before each DTC. If you have selected a translation file, the DTC components will be translated. Which DTC components will be logged, you can decide with the "ExtractXXX" settings (see below)
The log file will be stored on the device in the directory /opt/logfiles/. The variables @CurrentDirectory, @CurrentFile, @DestinationDirectory and @CopyFileOrFolder can be used to copy or move the log file to an USB stick.
Note: There is currently no restriction in the size of the log file. Please consider deleting or moving the log file to a USB stick at a regular basis (or if it becomes too big).

Open Alarm On Change

If the "Open Alarm On Change" checkbox is enabled, the Alarm Number of an existing Alarm can be entered in the text field.

Now whenever a new or changed DTC arrives in a DM message this Alarm will pop up and the currently active DTCs can be presented to the user.

Advise: place a table object inside the alarm to show all active DTCs (see below for table object settings).

Screenshot of an Alarm displaying DM1 Trouble codes in a table (with a valid translation file configured):

Active Trouble Codes:

ECU	Component	Description
EMS	Exh. Oxygen	Root Cause Unknown
EMS	Coolant Temp	Root Cause Unknown
TECU	Oil Temp.	Below Normal
EBS	Pressure	Below Normal
EBS	Pedal Pos.	Root Cause Unknown
TPMS	Tire R_F_R	Below Normal
TPMS	Tire R_R_L	Below Normal
TPMS	Tire R_F_L	Above Normal
TPMS	Tire F_L	Above Normal
TPMS	Tire F_R	Above Normal

Update Table with extracted DM1 Data

When the checkbox "Update Table with extracted DM 1 Data" is enabled, the ID of an existing Table Object can be entered in the filed "Enter ID of the Table Object".

The DTCs of a DM1 message will be extracted and passed as data to the Table Object. You can select which components of the DTC shall be passed to the Table Object by enabling or disabling the "Extract ECU Name", "Extract SPN", "Extract OC" and "Extract FMI" checkboxes.

If the Table Object shall only show ECU, SPN and FMI, select those three and deselect "Extract OC" (recommended for DM1 since OC is always 1 in DM1 messages). The Table Object should then be configured to have three columns.

Configure ECUs to track

In this section you can chose if you want to track the DM1 messages of all ECUs on the selected CAN port or if you want to only handle a selection of ECUs.

If you want to track all ECUs and do no filtering you can just select "Track All ECUs" in the bottom right.

If you want that not all ECU's DM1 messages are handled, you have to disable "Track All ECUs". After that you can click the "Add ECU" button to add a new ECU to the list of handled ECUs. Once the ECU was added, the address of the ECU can be changed by clicking in the cell in the table.

To remove an ECU from the white-list, you can just select one and hit "Remove selected ECU".

Each ECU can then be configured if all SPNs shall be handled or only the configured ones. Furthermore it can be selected if the DM1 message of one ECU shall be passed to a specific table object. To do the settings for each ECU, click the "…" button in the "Settings" column to open the ECU settings dialog:

In this dialog it can be configured if all SPNs of one ECU shall be extracted or if only the configured ones shall be handled. Furthermore it can be specified if an alarm shall be triggered if a certain SPN is received in a DM1 message.

If all SPNs shall be extracted, the "Extract Errors of all SPNs" checkbox has to be enabled.

If only some SPNs shall be handled, disable the "Extract Errors of all SPNs" checkbox. Then just write the SPNs in the "Add SPN" text field and click Add or hit <Enter> on the keyboard. The new SPN will appear in the table on the left.

If you want to remove one or more SPNs, just select it/them and click the "Remove Selected SPN" button.

After a SPN was added to the table it can be configured if a dedicated alarm shall be engaged when the SPN is included in the DM1 message. For that, the alarm number needs to be entered in the in the "Alarm" cell. The default value for that cell is -1 which means: engage no alarm.

Additionally it can be configured if the alarm shall only be engaged for a certain FMI number. Just write the FMI into the cell. The default value for the FMI cell is -1 which means: engage the alarm for any FMI.

To trigger different alarms for different FMIs, the same SPN can be added several times to the table and different FMIs and alarms can be configured.

For example take the screenshot above. It has the following SPN FMI Alarm settings:

110	-1	-1
116	0	16
116	1	17
177	-1	24
521	-1	-1
1119	-1	-1

That means:

- Track SPNs 110, 116, 177, 521 and 1119
- Don't engage an alarm for SPN 110 (FMI and Alarm are -1)

- Engage alarm 16 for SPN 116 if FMI is 0 (FMI is 0, Alarm is 16)
- Engage alarm 17 for SPN 116 if FMI is 1 (FMI is 1, Alarm is 17)
- Engage alarm 24 for SPN 177 and don't care for FMI (FMI is -1, Alarm is 24)
- Don't engage alarm for SPNs 521 and 1119 (FMI and Alarm is -1)

In the lower part of the dialog, it can be selected if the DM1 message entries of the selected ECU shall be passed to a specific table object. The settings in this dialog correspond to those in the parent dialog.

By clicking the "OK" button, all settings will be saved. When clicking "Cancel" all settings will be discarded.

DM2 Settings

The only settings that can be made for DM2 messages is to enable or disable it and "Update Table with extracted DM 2 Data" setting. This setting is the same as for DM1 messages. You can enter a different Table Object ID in the "Enter ID of the Table Object" field than for the DM1 Table so that you can have different tables for both message types.

DM1 and DM2 How To

How to get DM2 messages

DM2 Messages are only sent on request. They can be either requested for single ECUs or a global request can be sent. The following example shows how to make a global request for all ECUs to send the DM2 messages.

First you have to create an internal variable (**Communication** | **Variable Manager...** | button **Create A New Variable**) to which we can write. In this example it is called j1939_dm2_trigger, the owner is PClient and the data type is UNSIGNED8. You are free to choose index and subindex and the variable group:

Next a J1939 transmit PGN needs to be configured. Go to **Communication** | **Protocols** | **J1939** | **Configure Mappings...**. Enter "dm2_global_request" in the "Name", as type select Transmit field and click the "Add" button.

The PGN Id for this request is 0xea00, the priority is 6. Destination is Any ECU and for "Send Value on" select "Any Variable Change":

ID	5
Name	dm2_global_request
Type	Transmit
Description	Message to request all ECUs to send DM2 messages
Mapping Variable Type	Length
Delimiter	*
Transmit To	CAN1:Any ECU
Little Endian	☑
PGN	0xEA00
PGN Priority	6
⊞ Request Settings	
⊞ Receive Settings	
⊟ Transmit Settings	
Use as Write Request	☑
Send Value On	Any Variable Change
Select Variable(Tx)	None
Transmission Period (in ms)	0

Now add two entries in the **Configure mappings of variables for selected mapping object** section of the dialog by pressing the **Add** button twice. Make the same settings as in the screenshot below:

Byte Bit Pos...	Length	Is Const...	Constant Value	Group	Variable	Mask	Offset1	Shift X Bits	Scale	Offset2
1:1	16	☑	0xFECB			0xFFFF	0.0	0	1.0	0.0
3:1	8	☐	0x00	Common	j1939_dm2_trigger	0x00	0.0	0	1.0	0.0

Important is that the first 16 bit entry is constant with Mask 0xffff and the Constant Value of 0xfecb. The following byte is our j1939_dm2_trigger variable. Here the Mask has to be 0x0. Click OK on both dialogs.

How to trigger sending of this request:

Select a button or softkey in your project and attach an action e.g. to "OnRelease". As action select "Execute Script". Create a new script with the following content:

```
//Get the old variable value, increment it and set it again
//to trigger the sending of the DM2 global request.

var value = getVariableValue("j1939_dm2_trigger");

//overflow prevention
if (value >= 0x7fff)
{
value = 0;
}
value++;
setVariableValue("j1939_dm2_trigger", value);
```

Now every time the button/softkey is pressed and released, the DM2 request will be sent. You can also attach this action e.g. to the "Page Init" event of the page where the table is displayed.

Screenshots from an example DM2 Table Object running on adevice:
First without a translation file:

Stored Trouble Codes:

Where	What	Count	Reason		
0x0	110	3	11	Get all Errors	
0x0	1119	4	11		
0x3	161	2	1		
0x3	177	1	1	Clear Table	
0xb	1091	2	1		
0xb	521	1	11		
0x33	258867	2	17		
0x33	258867	2	1		
0x33	258867	1	15		
0x33	258867	1	0		
0x33	258877	3	15		
0x33	258877	2	0		
0x33	258877	2	1		
0x33	258877	3	17		
0x33	258876	2	15		

And with a valid translation file:

Stored Trouble Codes:

Where	What	Count	Reason		
EMS	Coolant Temp.	3	Root Cause Unknown	Get all Errors	
EMS	Exh. Oxygen	4	Root Cause Unknown		
TECU	Shaft Speed	2	Below Normal		
TECU	Oil Temp.	1	Below Normal	Clear Table	
EBS	Pressure	2	Below Normal		
EBS	Pedal Pos.	1	Root Cause Unknown		
TPMS	Tire R_F_R	2	Below Normal (Least)		
TPMS	Tire R_F_R	2	Below Normal		
TPMS	Tire R_F_R	1	Above Normal (Least)		
TPMS	Tire R_F_R	1	Above Normal		
TPMS	Tire R_R_R	3	Above Normal (Least)		
TPMS	Tire R_R_R	2	Above Normal		
TPMS	Tire R_R_R	2	Below Normal		
TPMS	Tire R_R_R	3	Below Normal (Least)		
TPMS	Tire R_R_L	2	Above Normal (Least)		

How to clear the data of a table object

Execute a script with the following content:

```
var emptyArray = new Array(0);
//the first parameter of setProperty function is the Object ID,
//make sure that you enter the correct ID (probably not 291)
var tableID = 291;
setProperty(tableID, "Table Data", emptyArray);
```

How to display logged DM1 messages on the device

Preconditions: Enable DM1 support. Enable "Log DM 1 Messages To File" and enter a file name. Disable "ExtractOC".
First create a Table Object. This table should have 5 columns.
The Table Heading should be: "Date;Time;ECU;SPN;FMI".
For updating the table with the logged DTCs, a Script has to be executed. Assign an "Execute Script" action to either a softkey's/button's "OnRelease" event or to the page's "Page Init" event.
The Script has to look like this:

```
//Create an Array to store each line of the log file:
var logFile = new Array();

//create a string with the name of the log file (the same that
was configured in DM1 Settings)
var logFileName = "j1939_DM1_log_file.csv";

//now a variable with the default log path:
var path = "/opt/logfiles/";

//now read the file
logFile = readFile(path + logFileName);

//after reading, the data has to be assigned to the table
//(change tableID to the tableID your table has):
var tableID = 332;
setProperty(tableID, "Table Data", logFile);
```

Here is a screenshot from the device displaying the log file contents:

Logged DM1 DTCs:				Refresh
Date	Time	ECU	SPN	FMI
2012.01.20	08:55:10	EBS	Pedal Pos.	Root Cause Unknown
2012.01.20	08:55:10	EMS	Exh. Oxygen	Root Cause Unknown
2012.01.20	08:55:10	EMS	Coolant Temp.	Root Cause Unknown
2012.01.20	08:55:11	TPMS	Tire R_F_R	Below Normal
2012.01.20	08:55:11	TPMS	Tire R_R_L	Below Normal
2012.01.20	08:55:11	TPMS	Tire R_F_L	Above Normal
2012.01.20	08:55:11	TPMS	Tire F_L	Above Normal
2012.01.20	08:55:11	TPMS	Tire F_R	Above Normal
2012.01.20	08:56:35	TECU	Oil Temp.	Below Normal
2012.01.20	08:56:35	EMS	Exh. Oxygen	Root Cause Unknown
2012.01.20	08:56:35	EMS	Coolant Temp.	Root Cause Unknown
2012.01.20	08:56:36	TPMS	Tire R_F_R	Below Normal
2012.01.20	08:56:36	TPMS	Tire R_R_L	Below Normal
2012.01.20	08:56:36	TPMS	Tire R_F_L	Above Normal
2012.01.20	08:56:36	TPMS	Tire F_L	Above Normal

Note: You can scroll through the table if "Set As Input" property of the table is enabled.
Note 2: The JavaScript readFile function is currently limited to only read the first 2MB of a file.
If the log file is larger than that, the rest of the content is not shown. Consider deleting the log file if the size of it gets too big.

DM1 Lamps

The standard SAE J1939-73 defines four lamps:
- Malfunction Indicator Lamp
- Red Stop Lamp
- Amber Warning Lamp
- Protect Lamp
The status of these lamps is transmitted in the DM1 messages of each ECU in bytes 1 and 2.
Each lamp has a status value (byte 1):

bits 8-7	Malfunction Indicator Lamp Status
bits 6-5	Red Stop Lamp Status

bits 8-7	Malfunction Indicator Lamp Status
bits 4-3	Amber Warning Lamp Status
bits 2-1	Protect Lamp Status

and a flashing indicator (byte 2):

bits 8-7	Flash Malfunction Indicator
bits 6-5	Flash Red Stop Lamp
bits 4-3	Flash Amber Warning Lamp
bits 2-1	Flash Protect Lamp

There are 8 predefined internal variables available that represent these values (read-only).
The variables are available in the group J1939 and are named

- @MalfunctionIndicatorLampStatus
- @RedStopLampStatus
- @AmberWarningLampStatus
- @ProtectLampStatus
- @MalfunctionIndicatorLampFlashing
- @RedStopLampFlashing
- @AmberWarningLampFlashing
- @ProtectLampFlashing

The @XXXLampStatus contains the combined information of the status and flashing bits in the DM1 message with 16 possible combinations. The @XXXLampFlashing variable contains the direct "is it on or not?" information, when the lamp is flashing, it will change value between 0 and 1, if the lamp should glow, it will stay 1. So normally only the @XXXLampFlashing variable is needed to display the lamps by creating a list object with two images (an on and an off lamp image) and connecting it with the @XXXLampFlashing variable.

The following table shows the possible values of the DM1 message bytes and our variables. The calculation is the same for all the four variables:

Value for Status read from DM1 messages (Byte 1)	Value for Flashing read from DM1 messages (Byte 2)	value of @XXXLampStatus	value of @XXXLampFlashing
0	0	0	0: Lamp Off
0	1	1	0: Lamp Off
0	2	2	0: Lamp Off
0	3	3	0: Lamp Off
1	0	4	0/1: **Flashing 1Hz**
1	1	5	0/1: **Flashing 2Hz**

Value for Status read from DM1 messages (Byte 1)	Value for Flashing read from DM1 messages (Byte 2)	value of @XXXLampStatus	value of @XXXLampFlashing
1	2	6	0: Lamp Off
1	3	7	1: **Lamp On**
2	0	8	0: Lamp Off
2	1	9	0: Lamp Off
2	2	10	0: Lamp Off
2	3	11	0: Lamp Off
3	0	12	0: Lamp Off
3	1	13	0: Lamp Off
3	2	14	0: Lamp Off
3	3	15	0: Lamp Off

As can be seen in the table above the status 2 and 3 for the lamp status and value 2 for flashing will always switch off the flashing variable. If the used ECUs support these states, a custom handling of these values needs to be developed manually by using the value of the @XXXLampStatus variables.

If multiple ECUs send DM1 messages, the lamp stati and flashing information will be combined from all messages. The priority (highest to lowest) is: Fast Flashing, Slow Flashing, On, Off.

To enable the calculations for the lamp variables DM1 support needs to be enabled.

5.4   Import DBC File

 Tutorial video of the DBC import
https://www.youtube.com/watch?v=4JgRJyLcjRE

Please note that this feature has to be bought first. Contact your vendor to obtain a license. Without a license, it is not possible to import the content of a DBC file.

Disclaimer!

Please always validate your DBC file in Vector before importing it. This will ensure that the import works and the functionality is as expected.

If you have problems importing a DBC file and you don't think it's a problem in the file, please send the file to your vendor for analysis.
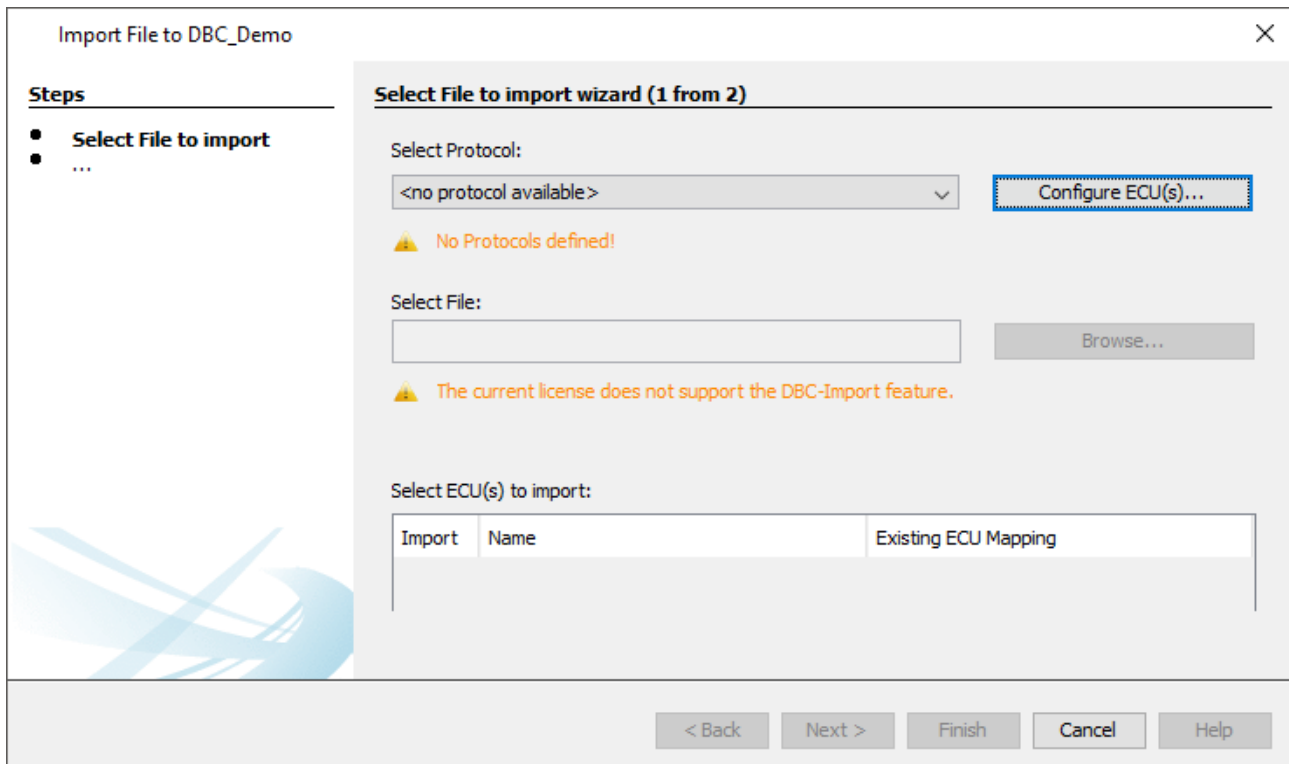
To start the import, open the dialog under the menu Communication -> Import-Export -> Import DBC File.
Note that the dialog can only be opened with an opened project.

The import process is done in two steps.

In the **first** step, a CAN protocol needs to be chosen and a DBC file is selected.
In the **second** step, the CAN messages and variables to be imported are selected.

1. Select File for Import
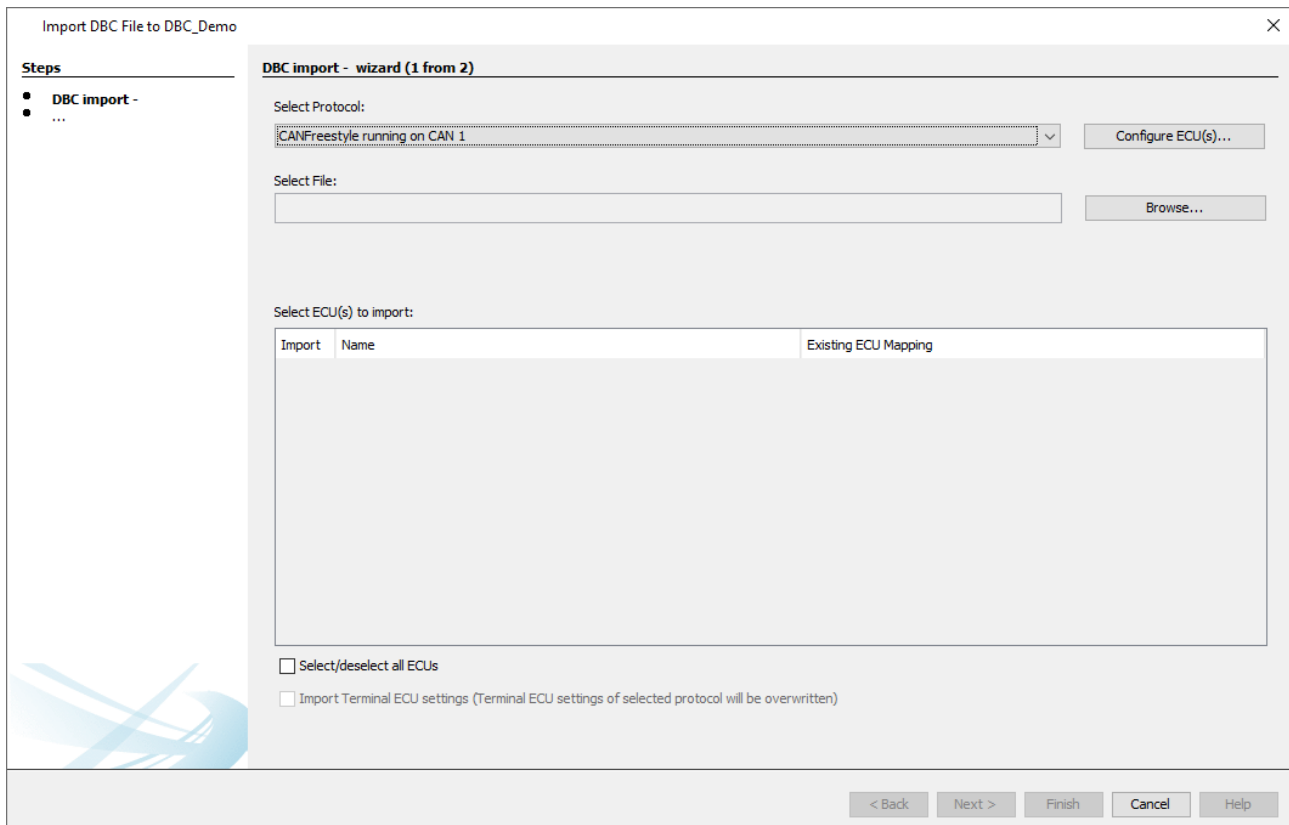
If no CAN protocol has been added to the project, an import of a DBC file isn't possible, so first at least one CAN protocol has to be added.
The Configure ECU(s) dialog can be reached directly from this dialog with the button "Configure ECU(s)...".
If there are several protocols in the project, one has to be chosen for the DBC import, i.e. in which protocol should the CAN mappings be added.

When a protocol has been added and / or chosen, a DBC file can be loaded.

The program will check the integrity of the DBC file. If there is a problem parsing the file, an error message will be shown, indicating the problem with the file.

In this case the parsing failed with an error in the Message_3 of Signal_9. The hint [NumberFormatExceoption] shows the root cause for this issue. In the example the Factor 1g could not be parsed.
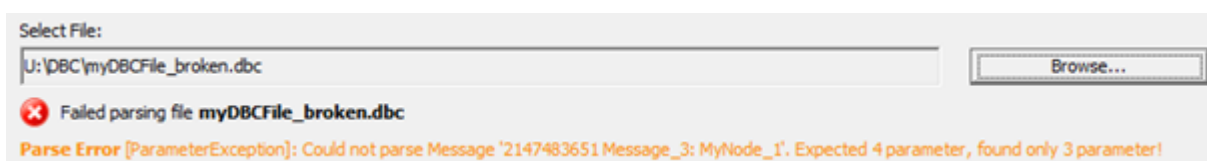
In this case the parsing failed with an error in the Message_3. The hint [ParameterExceoption] shows the root cause for this issue. In the example a parameter Message-Length is missing.

NOTE: Please validate the DBC-File with Help of Vector CANdb++ Editor and solve all warnings before importing the DBC-File.

When the DBC file is loaded successfully, at first the ECU(s) used in the DBC file can be mapped to (an) existing ECU(s).
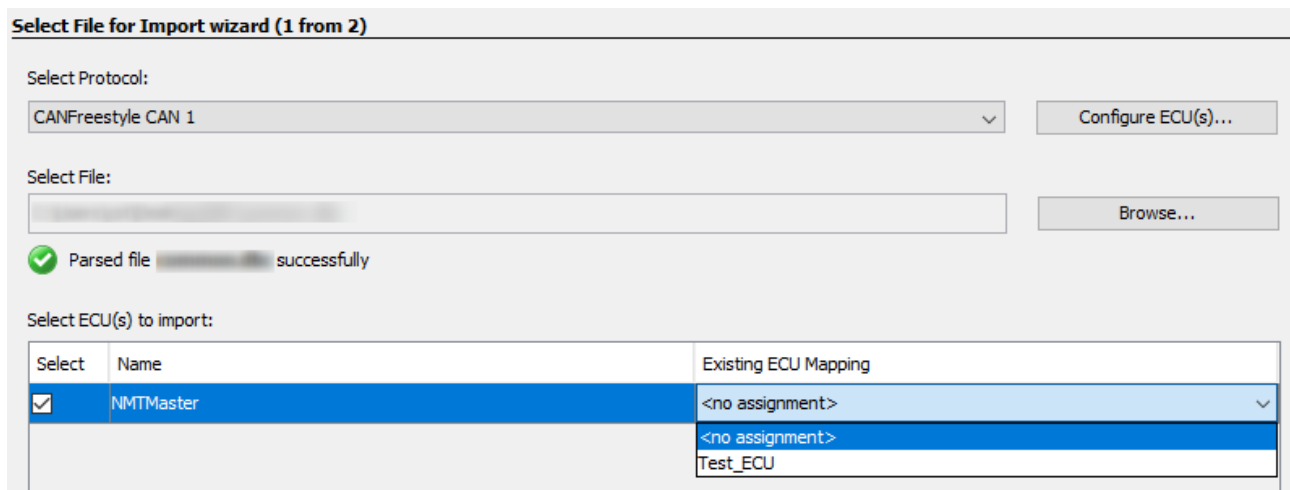
If no ECU is chosen (<no assignment>), the ECU will be created for the selected protocol. If an ECU with the same name already exists, the ECU name is marked in red. Then the ECU needs to be mapped to an existing ECU, since there can't be two ECU's with the same name.
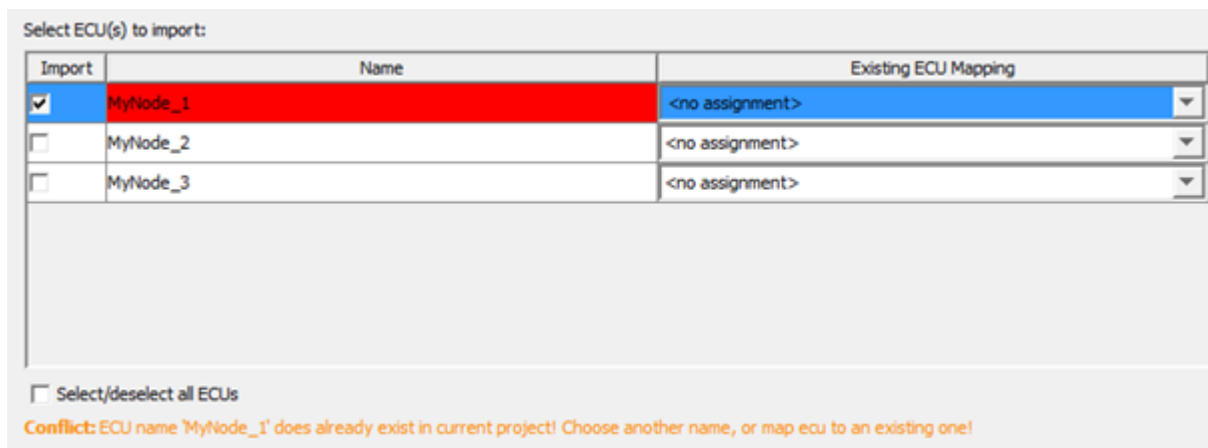
For the CAN-Protocol CANopen the COB-ID oft the ECU must be unique. In case that the COB-ID already exists the ECU can not be imported.
To solve the error either the COB-ID in the DBC-File must be updated or the COB-ID of the already existing ECU must be updated.

Below the ECU selection box there are two check boxes.

With the fist check box, all ECUs can be selected / deselected for import.
With the second check box, the terminal ECU settings (the CAN protocol settings for the OPUS device) can be imported from a DBC file.
Note: The check box is only available if such settings have been exported into the DBC file. Only DBC files from version 2020.4 onwards can have these settings included.
Note: The check box is only available for J1939 and CANopen imports, as CANFreestyle doesn't have any necessary Terminal ECU settings.
Note: The check box is only available if the selected protocol and the Terminal ECU settings in the DBC files are from the same protocol.

If no ECU is specified in the DBC file, no further settings can be made

Once all settings have been made, click Next to continue.
The import process can be canceled at any time by clicking Cancel.

2. Select CAN Messages and Variables
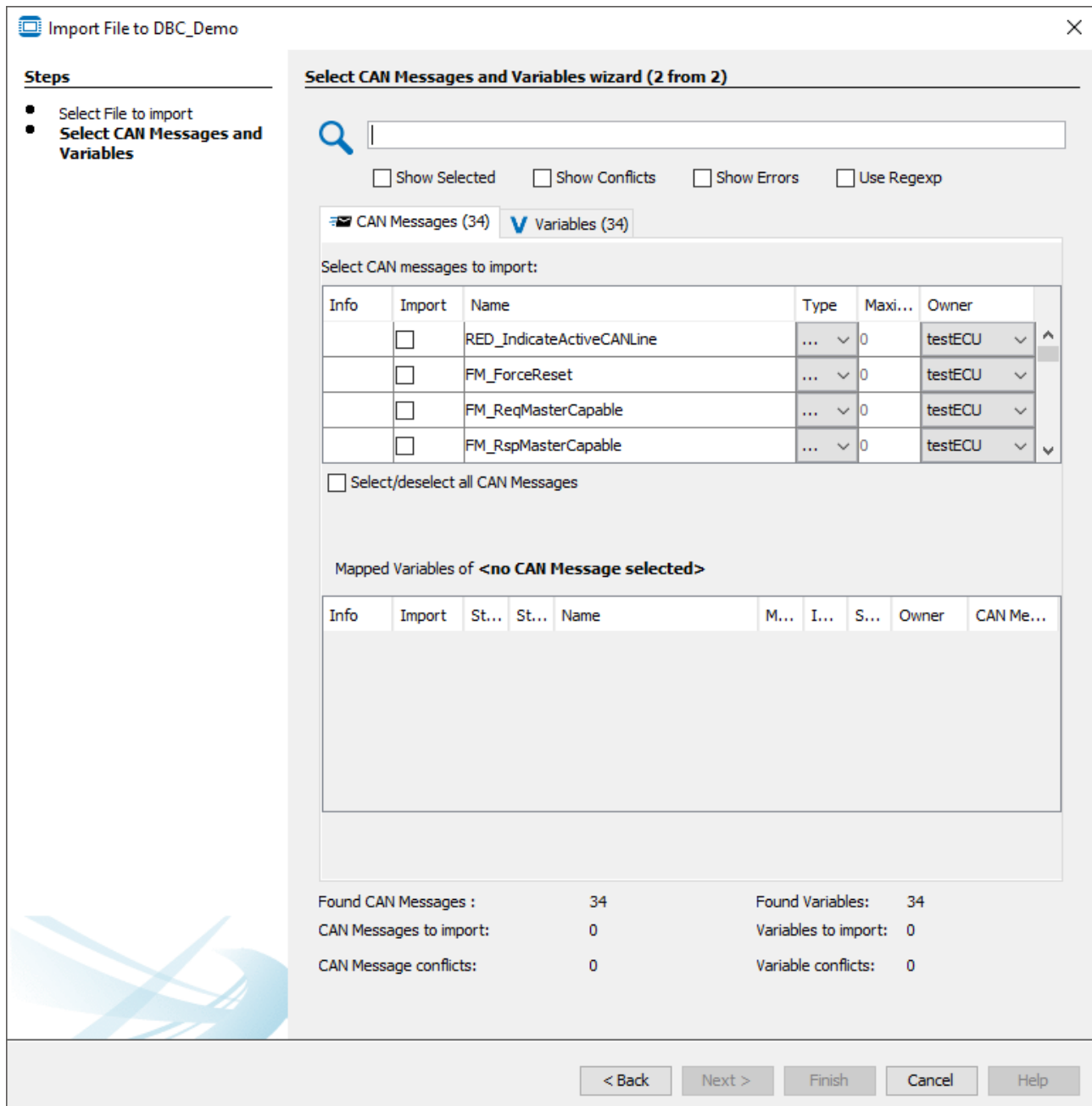
In the second step of the dialog the CAN messages and variables for import can be selected. To select a CAN message for import, activate the check box in the Select column in the upper table of the dialog. With Select/deselect all CAN Messages, all messages can be selected or deselected at once.

If a CAN message in the DBC file is not compatible with the selected CAN protocol (e.g. a J1939 message with > 8 bytes is defined in the DBC file, but the CAN Freestyle protocol has been selected to be used), the CAN message will be displayed with a red error icon and cannot be selected.
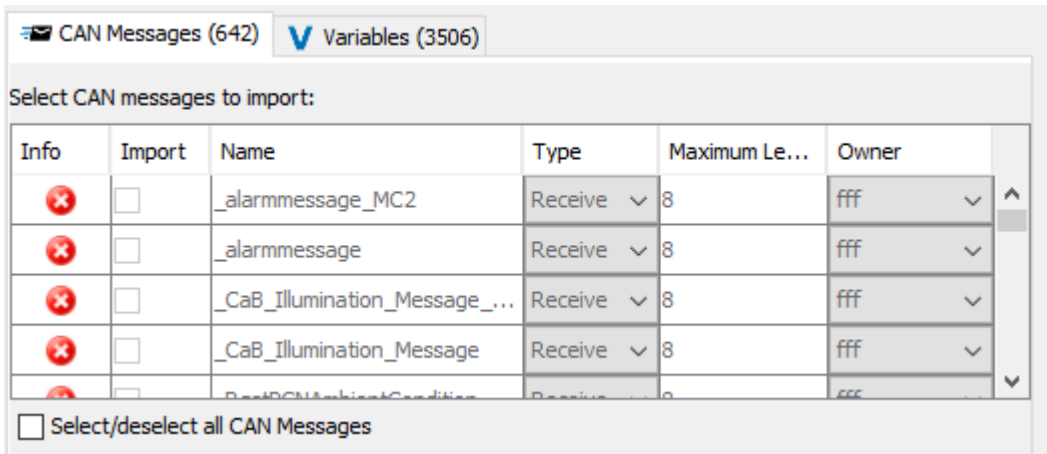
The direction type (Receive or Transmit) can be set in the column Type. The name of the CAN mapping can also be changed from what is given in the DBC file. Finally, the owner can be changed in the column Owner to specify from which ECU the message will be sent.

When a CAN message is selected for import, the according variables are automatically selected for import, as well.
The table below the CAN messages shows the mapped variables of the currently selected CAN message.
The variable name, index, sub index and owner can be changed in this table. When the name, index and sub index is changed, it is checked if there is a conflict with existing variables. If there is, the according variable will be marked in red and an import is not possible.
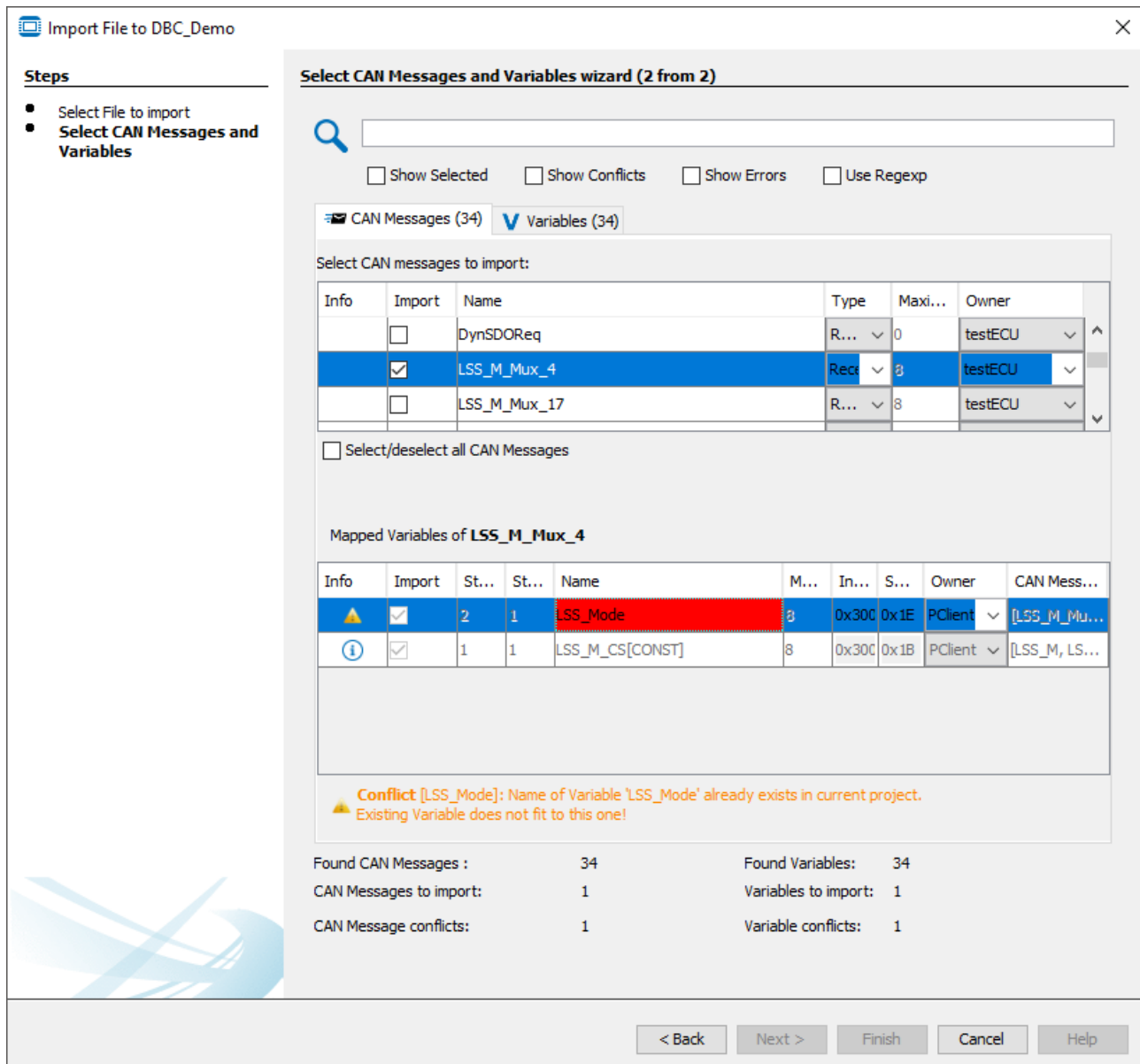
Only when there are no conflicts, the import process can be finished.

If a variable is already existent in the project and the data type matches the variable in the DBC file, the variable will be used. In the Mapped Variable table, such variables will be shown with a gray background. No properties of such variables can be changed (since the variable is already created in the project).
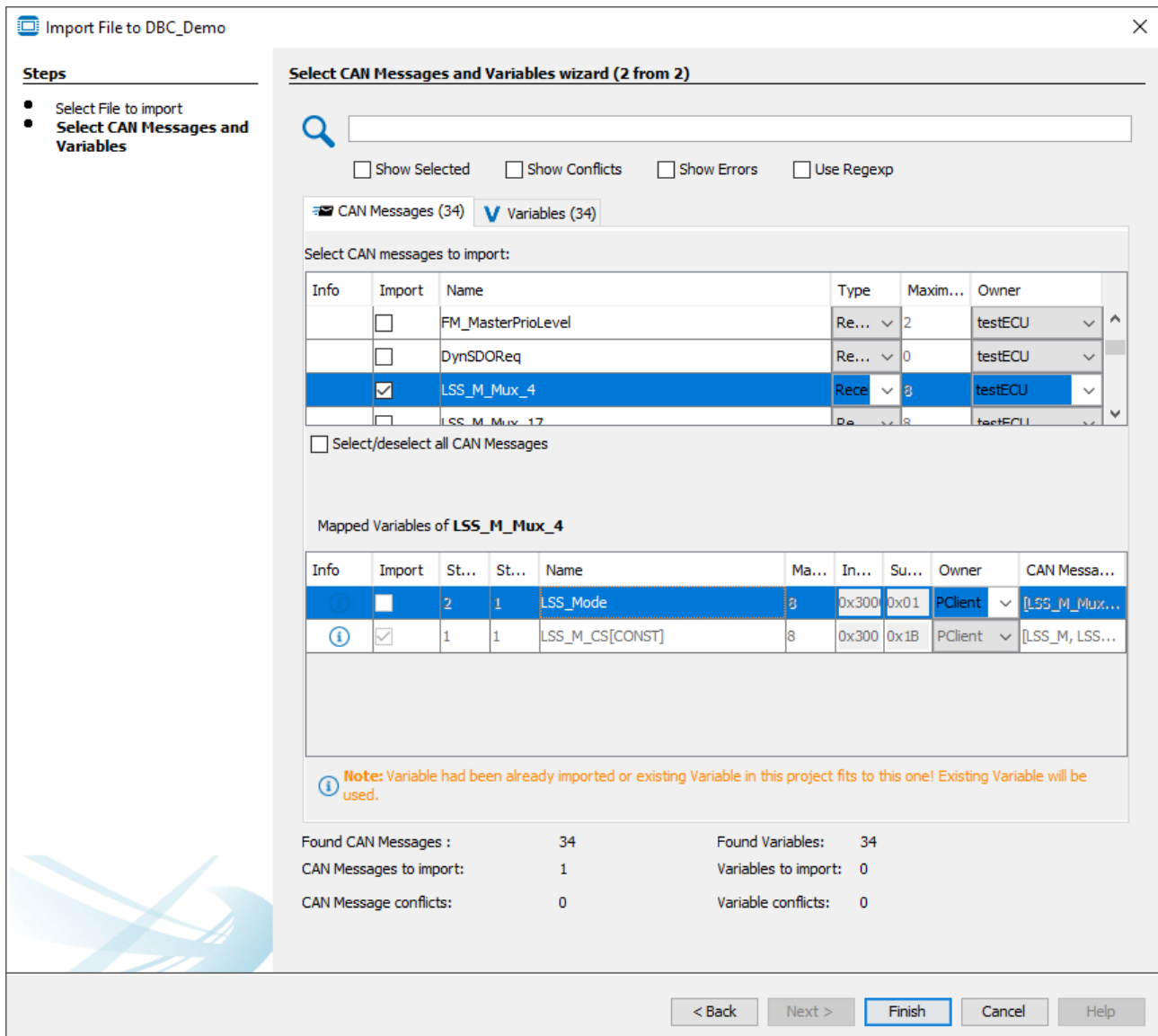
In the bottom of the dialog an overview of the CAN messages and variables in the DBC file and the ones selected for import are shown.

In the Variables tab all variables in the DBC file are shown. In this tab, variables can be selected and deselected for import.

Please note: If a variable is deselected in this tab, this will lead to the deselection of all CAN messages where this variable is mapped.
Variables can be imported without the CAN messages, but CAN messages can not be imported without the according variables.
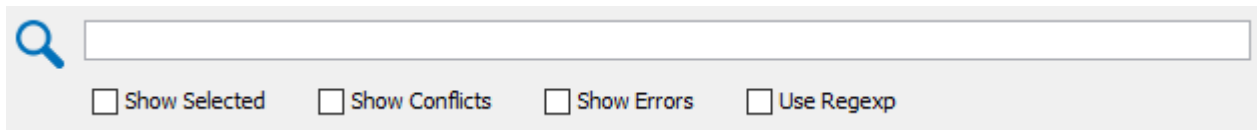
In the upper part of the dialog the search field can be used to filter the currently displayed variables / CAN messages.
Below the search field the current view can be filtered for

- only elements selected for import
- elements that have conflicts left to be resolved.
- elements with errors that can not be resolved

Additionally, regular expressions can be used to search in the DBC file data.

The search can also be used to specifically search in certain data fields:

CAN Messages tab:
- Column Import [boolean] => **#Import**=false/true => true if Object is selected for import
- Column Name [String] => **#Name**=<MappingName> => NOTE: Variable Mappings Name will also be considered!
- Column Type [String] => **#Type**=Receive/Transmit
- Column Length [Integer] => **#Length**=8 #Length<=4
- Column Owner [String] => **#Owner**=<OwnerName>

Other searchable properties are:
- Description [Sring] => **#Description**=<Text>
- CanImport [boolean] => **#CanImport**=false/true => false shows all Mappings which can not be imported!
  Mappings conflicts which can not be solved by user! E.g. Messages with Length > 8 Byte for CANFreestyle
- COB-ID (CANopen) [int] => **#COB-ID**=0x111
- PGN (J1939) [int] => **#PGN**=0x1200
- CAN-ID (CANFreestyle,) => **#CAN-ID**=0x16

Note: The term CAN-ID is used for all protocols, i.e. for CANopen it searches COB-IDs and for J1939 PGNs.

Variables tab:
- Column Import [boolean] => **#Import**=false/true => true if Object is selected for import
- Column GroupName [String] => **#GroupName**=<GroupName>
- Column Name [String] => **#Name**=<VariableName>
- Column Length [Integer] => **#Length**=8 or **#Length**<=4
- Column Index [Integer] => **#Index**=0x3000
- Column SubIndex [Integer] => **#SubIndex**=0x01
- Column Owner [String] => **#Owner**=<OwnerName>
- Column CANMessage [String] => **#CANMessages**=<MessageName>
  #Messages=<MessageName>
  Other searchable Properties are:
- Description [Sring] => **#Description**=<Text>
- CanImport [boolean] => **#CanImport**=false/true => false shows all Variables which can not be imported! e.g. Length of Variable is >32bit

Examples:

#CAN-ID=0x7e5 searches for all mappings with ID 0x7e5

#Name=engine searches for all mappings that contain the character sequence "engine" anywhere in the mapping name or variable name

Note: Search terms can be combined (AND) by typing them one after another:

#CAN-ID=0x7e5 #Name=engine searches for all mappings with ID 0x7e5 that have engine in the mapping name or variable name

Note: Invalid search terms will be ignored, e.g. the upper search term combination in the Variables tab will ignore the #CAN-ID search term.
Note: The search terms are case-sensitive.

Additional information

- The "producer" property of a message will be set as the "Receive From" value in a Receive mapping
- The "consumer" property of a message will be set as the "Transmit To" value in a Transmit mapping
- Scale and Offset of CAN messages will be imported as scale and offset 2 properties for CAN mappings. Offset1 is only populated if the DBC file was created by the program in version 2020.4 or higher

Finish

Only when there are no conflicts, the import process can be finished by clicking Finish.
Then a dialog will be shown with an overview of what has been imported.
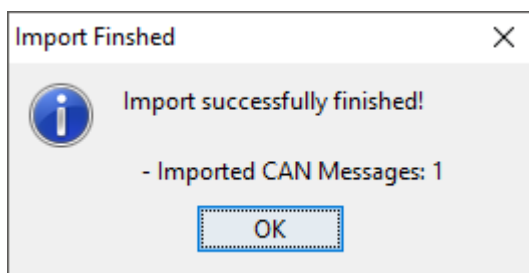
The import process can be canceled at any time by clicking Cancel.

5.5 Export DBC File

Please note that this feature has to be bought first. Contact your vendor to obtain a license. Without a license, it is not possible to export into a DBC file.

Disclaimer!
We try to adhere to the DBC file format as well as possible. If you find that an exported DBC file from our program, please contact the technical support from your vendor with your project, the DBC file and a description what is missing or wrong.

To start the export, open the dialog under the menu Communication -> Import-Export -> Export DBC File.
Note that the dialog can only be opened with an opened project.
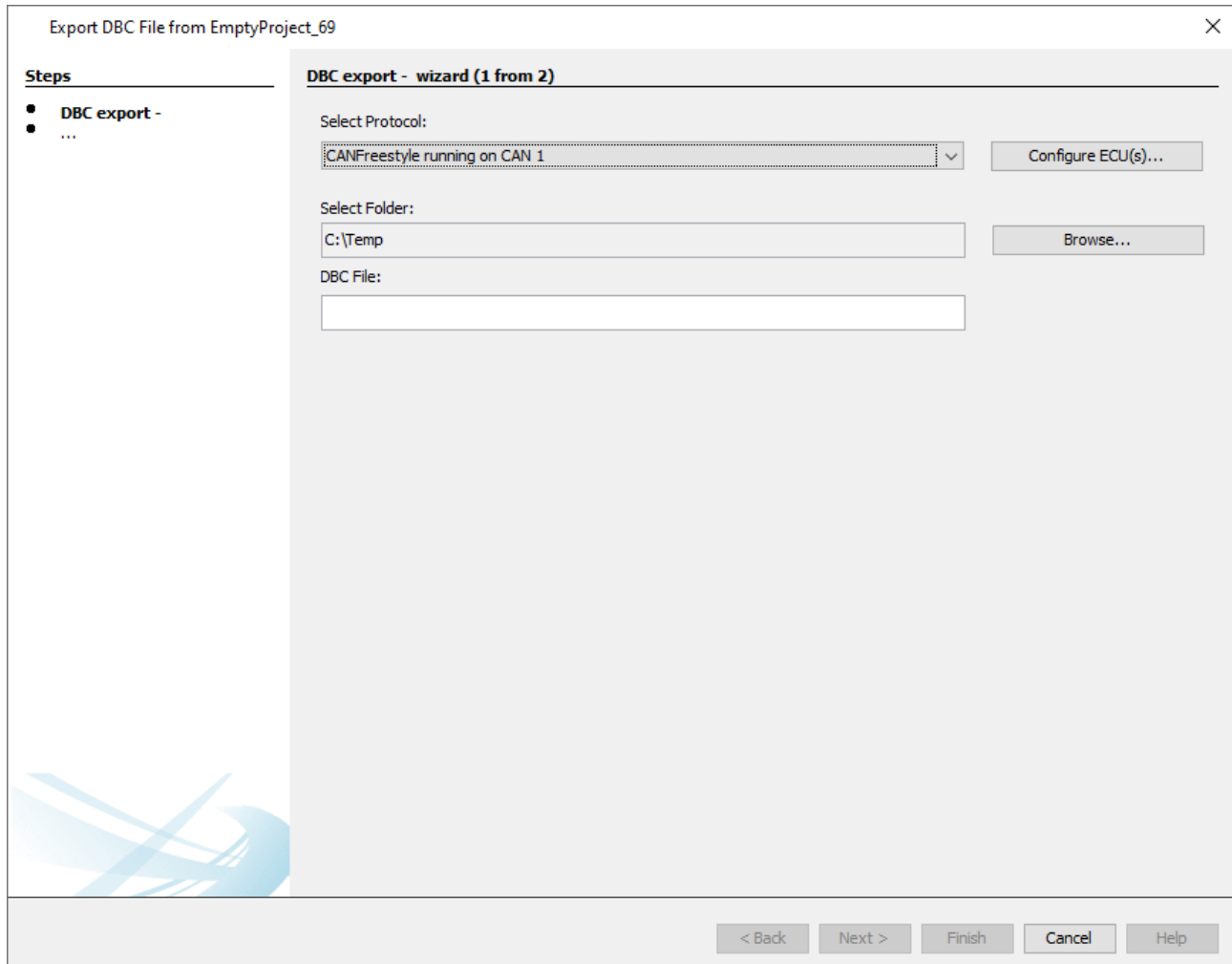
The export process is done in two steps.

In the **first** step, a CAN protocol needs to be chosen and a DBC file name is given.
In the **second** step, the CAN messages and variables to be exported are selected.

1. Select Protocol and File for Export

If no CAN protocol has been added to the project, an export of a DBC file isn't possible, so first at least one CAN protocol has to be added.
The Configure ECU(s) dialog can be reached with the button "Configure ECU(s)...".

If there are several protocols in the project, one has to be chosen for the DBC export, i.e. from which protocol the CAN mappings should be exported.

A folder to save the file in has to be selected. Then a valid file name has to be given. The file name needs to be complete, i.e with the extension, e.g. "test.dbc".

If a file with the same name already exists, a warning will be displayed that the existing file will be overwritten by the export.

Once all settings have been made, click Next to continue.
The export process can be canceled at any time by clicking Cancel.

2. Select CAN Messages and Variables
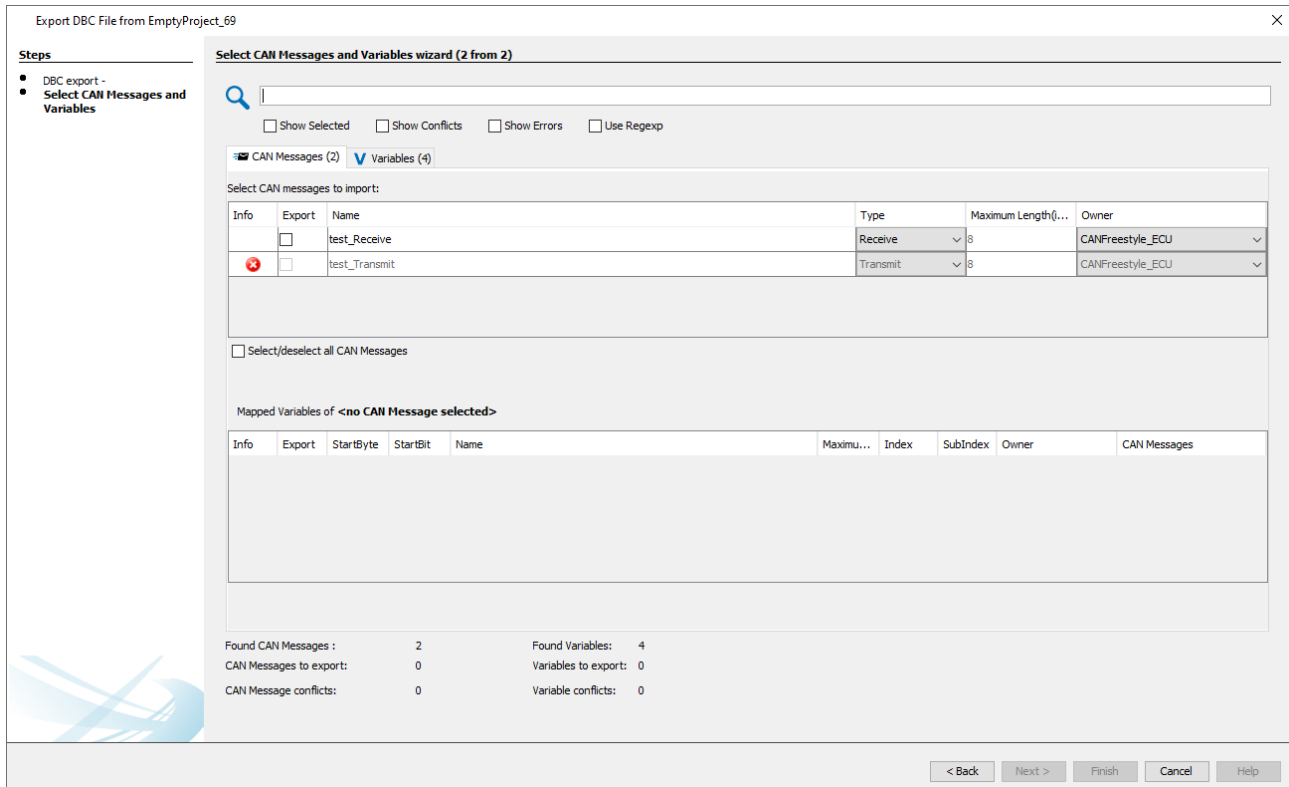
In the second step of the dialog the CAN messages and variables for import can be selected. To select a CAN message for import, activate the check box in the Select column in the upper table of the dialog. With Select/deselect all CAN Messages, all messages can be selected or deselected at once.
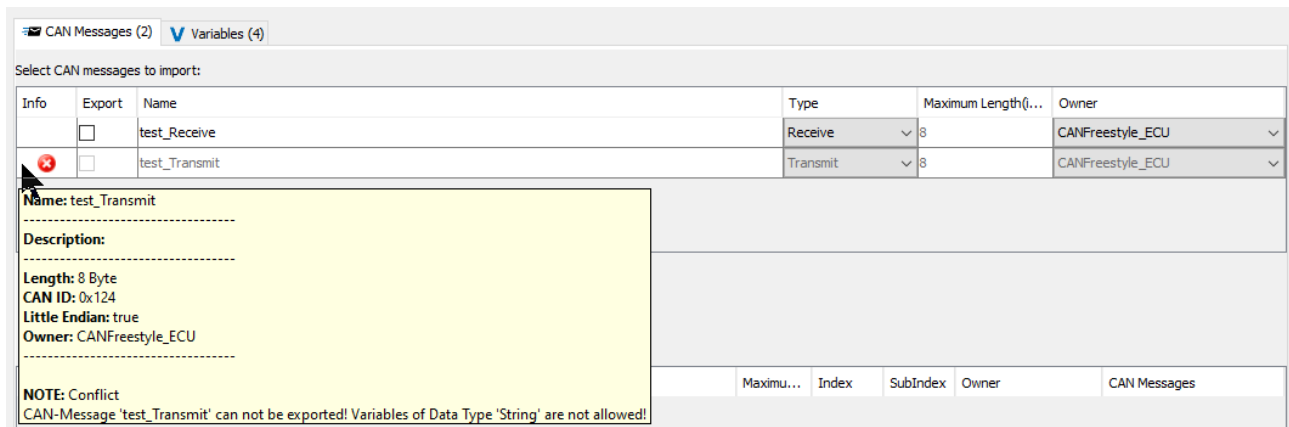
If a CAN message cannot be exported into a dbc file, the CAN message will be displayed with a red error icon and cannot be selected.
The restrictions for an export are:

- No String variables must be in a mapping. Strings are not supported by Vector
- No Delimiter-based mappings can be exported. Delimiter-based mappings are not supported by Vector
- Multiplexer-Messages are only available for CANFreestyle and J1939 protocols, but not for CANopen

- Messages with same CAN-ID but different Multiplexer-Positions can not be exported. This is not supported by the DBC format
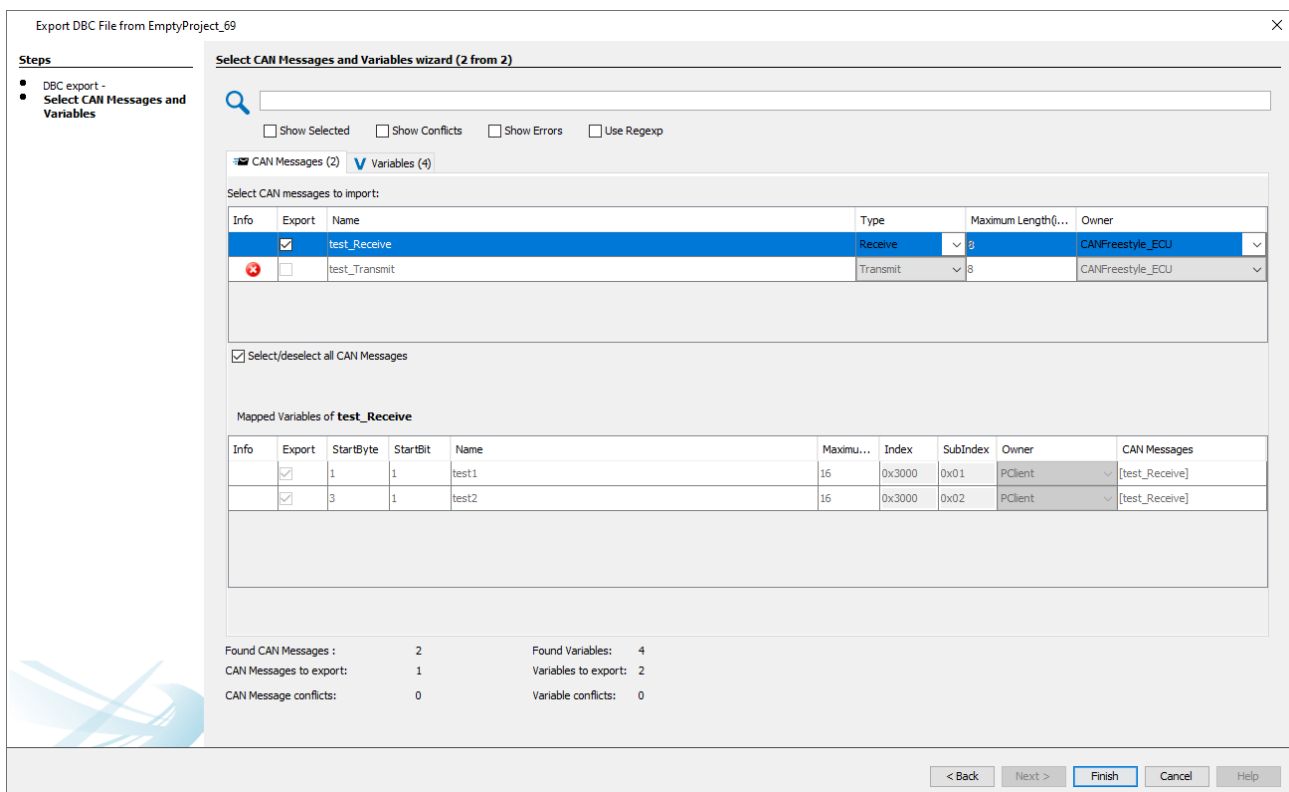
When a CAN message is selected for import, the according variables are automatically selected for import, as well.
The table below the CAN messages shows the mapped variables of the currently selected CAN message.

Note that the variables cannot be de-selected and no properties of the variables can be changed.

The export process can be finished by clicking Finish.
Then a dialog will be shown with an overview of what has been exported.
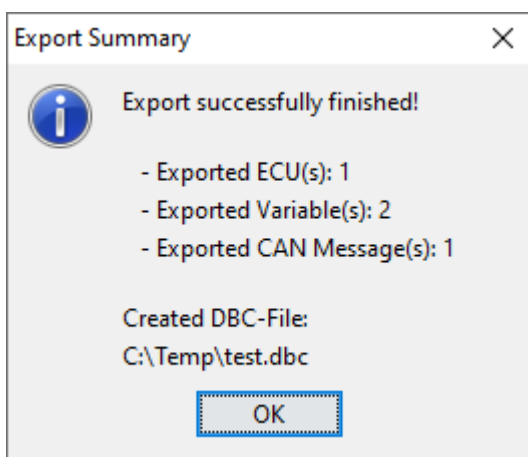
The export process can be canceled at any time by clicking Cancel.

The following properties of CAN mappings are exported:

All protocols:

- Type (RX/TX)
- Producer of the message (for Receive the "Receive From" value will be set, for Transmit the device will be set)
- Consumer of the message (for Receive the device will be set, for Transmit the "Transmit To" value will be set)

CANFreestyle:

- Rx/Tx: The property CAN ID(0x) is used for CAN-ID
- Rx: The property Receive Timeout is saved in attribute GenMsgCycleTime
- Rx: The property Enable Request is saved in attribute GenMsgRequestable
- Tx: Transmission Period

CANopen:

- Rx/Tx: The property COB-ID is used for CAN-ID
- Rx/Tx: The property Event Timer is saved in attribute GenMsgCycleTime.
- Tx: Event Timer
- The CANopen address of the device (Node ID)
- The Heartbeat Producer Time
- The Heartbeat Time
- SDO COB-IDs
- The SDO Timeout

J1939:

- Rx/Tx: The properties PGN Priority, PGN and the source address is used for CAN-ID (as described in J1939 standard)
- Rx: The property Receive Timeout is saved in attribute GenMsgCycleTime
- Rx: The property Enable Request is saved in attribute GenMsgRequestable => It is only set to true if also the property Use Standard Request is selected.
- Tx: The property Transmission Period (in ms) is saved in attribute GenMsgCycleTime
- The Preferred Source Address
- The Terminal Name and address claim properties

The following properties of variables are exported:

- The name of the variable is used for signal name. NOTE: EMPTY_SPACE will be converted to _ because it is not allowed in DBC-Format
- For constants, the prefix CONST_ is used with the variable mapping size, e.g CONST_8
- For constants, the suffix __n, with n starting at 1, is used if more than one constant is used in the source project

- The name of predefined variables e.g. @AlarmShow is not valid in DBC-Format. (the @ sign is a delimiter) so the prefix __AT__ is used instead (e.g. __AT__AlarmShow)
- The Scale and Offset2 of variable mappings is used for setting Scale and Offset. (Offset1 will be exported as a user-defined attribute that won't be used in Vector)
- The property values Min and Max of the mapped variable will be set as Min and Max. NOTE: If the variable mapping does not use the whole length of the variable, the Min and Max value is set appropriate to the mapped length. For constants the Const value is used for Min and Max
- Unit is currently not supported and will be always set to EMPTY_STRING
- The owner will be always set to Vector__XXX (default owner)

FAQ

What about multiplexer messages?
The DBC format supports multiplexer messages, and our program supports them, as well. For multiplexer messages to be recognized as that, messages that use the same CAN ID (which is otherwise forbidden in Vector) need to have a identifiable multiplexer byte, i.e. a constant in the same position in the mapping. Otherwise messages with the same CAN ID cannot be exported.
If there are several constants in different positions, it cannot be detected which should be used for the multiplexer byte, so the export will not work, as well.

5.6 Visual CAN Mapping

Visual CAN Mapping *(VCM)* is a graphical view for mapping objects. With Visual CAN Mapping there is an easy and comfortable way to create or adjust mapped variables.
Via drag and drop it is easy to add/move or remove variables in the table. So it is possible to update the mapping object very fast.

Overview

The Visual CAN Mapping view is divided in 3 views.
1. **VCM Table:** Here you can see all mapped variable and it is possible to add / move or remove them by drag and drop.
2. **VCM Toolbar:** In the toolbar you can *"Add"*, *"Move Up"*, *"Move Down"* and *"Delete"* variables. Within there is the possibility of a full search after variables and which will be shown at the *Variable Tree*. In addition you can drag variables from *Variable Tree* to the *VCM Table*.
3. **VCM Configure Mapping Table:** If variable is selected at the *VCM Table*, the properties of the mapping will be shown here. The table is similar to the table at the tab **"Configure Mapping"**
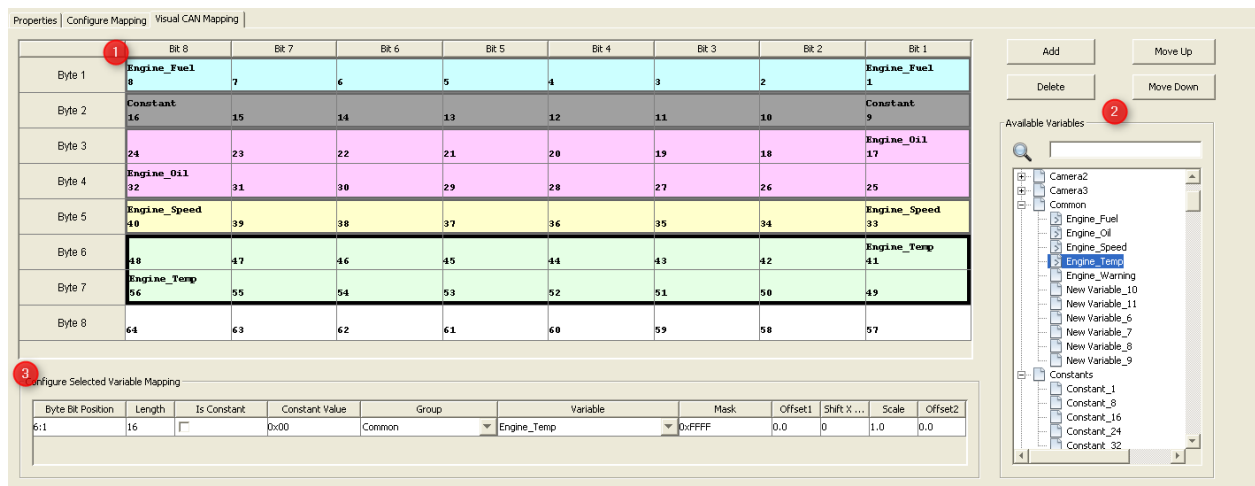
Open Visual CAN Mapping view

Visual CAN Mapping is available for all protocols (CANopen, J1939 and CANFreestyle)
It can be opened at the dialog **"Configure Mappings..."**.
To open Visual CAN Mapping please navigate to:
    Communication - Protocols -
        (1) CANFreestyle   or
        (2) CANopen   or
        (3) J1939
           - **(4)** Configure Mappings...
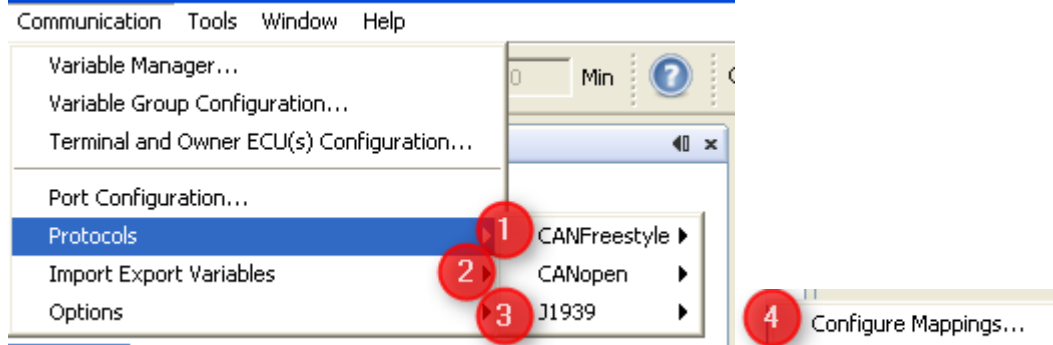
A new dialog "Configure Mappings Dialog" will be opened.
Afterwards you can open the VCM by selecting a **(1)** "Mapping Object" and clicking on **(2)**"Visual CAN Mapping" on the tab.
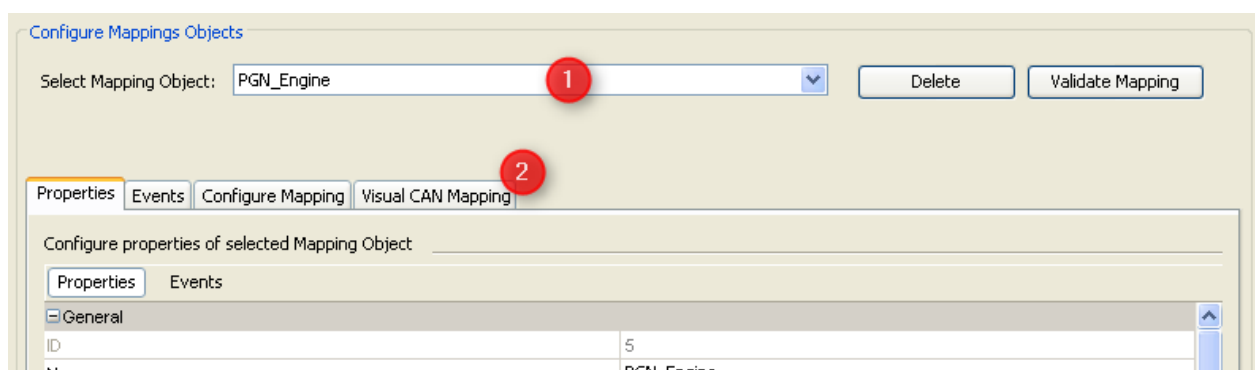
Alternatively, you can use the three shortcuts in the tool bar.

VCM Toolpanel
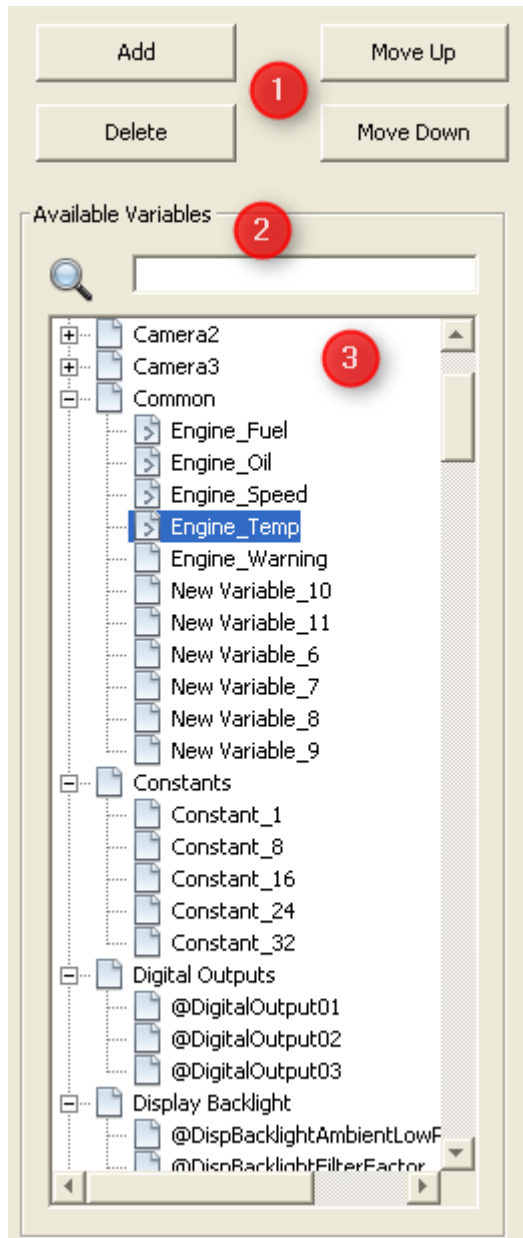
(1) Button panel for add, delete, move variable mappings. The buttons will be disabled/enabled depending on the state of the VCM table.
(2) Search field to search after variables
(3) Variables tree with all variables which can be mapped to current mapping object. The variable tree will be update automatically when the search field changes.

Adding Variables

There are 2 possibilities to add variables/constants

1. Add variable/constant with "Add" button.
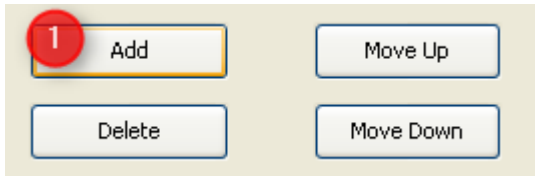
1a Add variable

By clicking on "Add" **(1)** a new dialog will be opened. In the dialog you can select a variable which is allowed to map to the current mapping object.
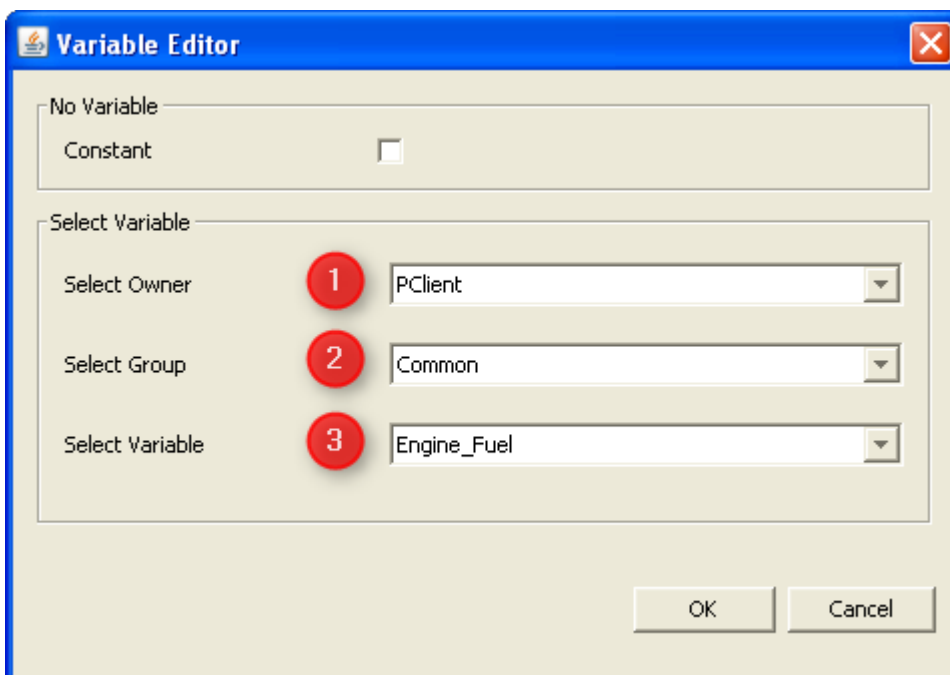
You can filter the variable in 2 categories:
- filter by **"Owner" (1)**
- filter by **"Group" (2)**

After changing the filter types the possible variables are shown in the last combo box **"Variables" (3)**.

By clicking **"OK"** the selected variable will be added to the current mapping object.

1b. Add constant

By clicking on **"Add"** a new dialog will be opened.
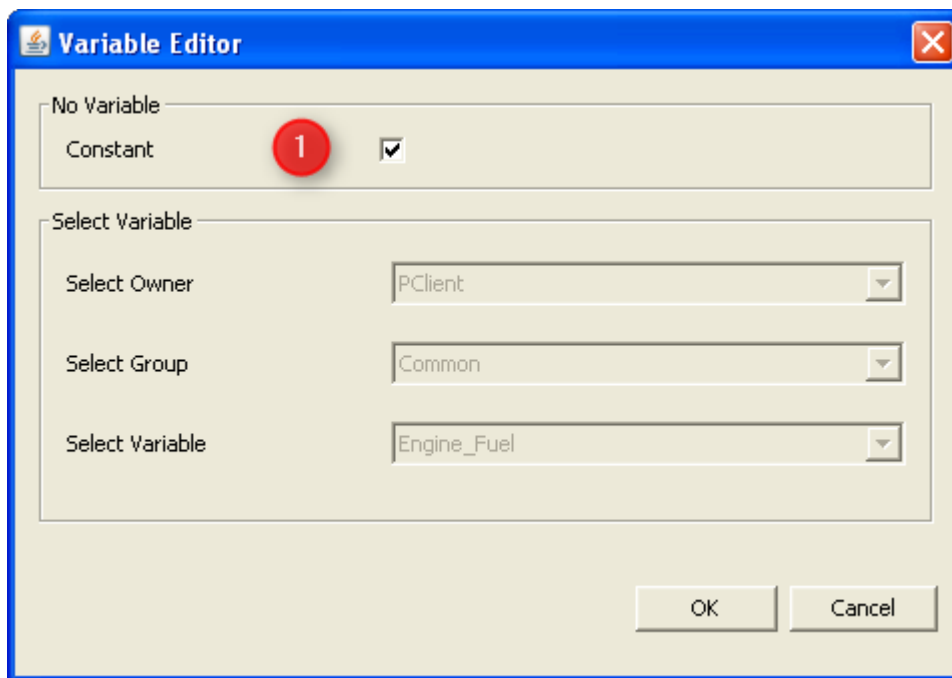
If you want to add a constant you have to select the check box **"Constant" (1)**. All other fields will become disabled.
By clicking **"Add"** a constant will be added to the current mapping object.

Note 1:
If a variable is selected the new variable/constant will be added at the start bit of the selected variable. All other variables will be moved as much as the length of the added variable.
If no variable is selected the new variable will be added after the last mapped variable.

If there is not enough space for the original length of the variable the mapped length will be reduced.
If the mapping object is full mapped the add button will be disabled. When it is tried to drag and drop a variable from variable tree the variable will not be added to the mapping object.

2. Add variable from variable tree

2a Add variables via drag and drop from variable tree
A comfortable way for adding variables is to drag and drop them from the variable tree. All available variables will be shown in the tree sorted by groups.
To add a variable to the mapping object you can drag one and drop them at a cell of the table.

2b. Add constants via drag and drop from variable tree.
If you want to add a constant from variable tree you can select a constant form group "Constants" (only for CANFreestyle and J1939). For CANopen the group "Constants" is replaced with the group "Dummy Variables".
It works as same as for variables (see above **2a.**).

Note 2:
Constants are **not available** for **CANopen** mappings and **J1939 delimiter** mappings.
The length of the constant can be changed by clicking on the constant at the **VCM table**.
Afterwards you can update the length at the **VCM Configure Mapping table**.
For more information about the **VCM Configure Mapping table** look [here](#)^{397}.

Deleting variables

Like with adding a variable there are also several ways to delete a variable:
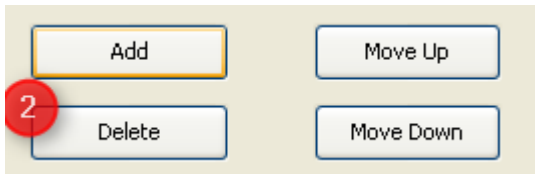
1a. Clicking "Delete" button
The delete button is only enabled if at least one variable is selected at the **VCM table**.
All selected variables will be deleted when **"Delete"** is pressed. The position of other
mapped variables will not change (**except for CANopen**).

1b. Delete variables via drag and drop
A second way to delete variables is to drag variables and drop them outside of the table.

1c. Delete variables with press of "Del" key
Another way to delete variables is to press the **"Del"** key of the keyboard. Only the
selected variables will be removed

1d. Delete variables via context menu
A variable can be removed by clicking with right mouse button on the variable and then
choosing **"Delete"**.

Moving variables in the table
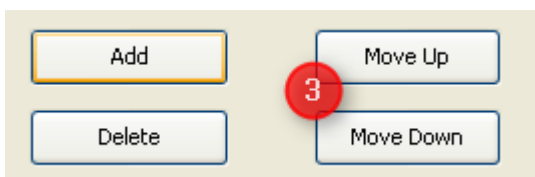
1. Move variable up/down with buttons
Selected variables can be moved in the **VCM table** via the buttons **"Move Up"** or
"Move Down".

If a variable is moved down and behind the current variable is another variable, the
variables switch there positions.
If there is no other variable behind the current selected on the variables moved by 1 bit.

For moving up it works the same way.

2. Trigger move up/down via mouse wheel

The move function can also be triggered with the mouse wheel. If you want to to move up or down selected variables, you have to press the **left "Ctrl"** key of the keyboard together with **mouse wheel up** or **down**.

Note:

If the current variable starts at bit 1. The button **"Move Up"** is disabled.

If the mapping object is **full mapped** and the last variable is selected the button **"Move Down"** is disabled.

If the **last bit** of mapping is mapped (e.g. bit 64 for CANFreestyle mapping) the mapping object is **full mapped.**

Variable tree
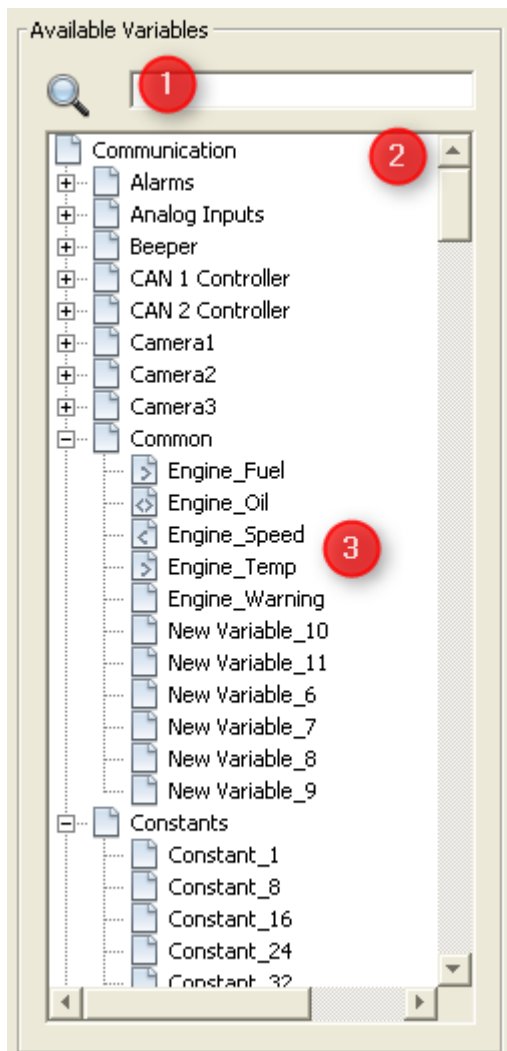
1. Structure

The variable tree contains all variables which can be mapped for the current mapping object.

At the searchfield **(1)** you can search after variables by their
- names
- groups
- owners
- index
- subindex

The variable tree **(2)** will be updated in real time after the first input at the search field.
Only variables which hit the search text will be shown.
To add a variable to the VCM table drag a varaible **(3)** and move it to the VCM table.
After dropping the variable, it will be visible at the table.

2. Variable icons
With the help of the variable icons it can be seen if the variable is already mapped in one of the mappings of the current protocol.
There are 4 icons:

1.Variable is not mapped in any mapping

 Engine_Warning
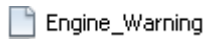
2. Variables is mapped in at least one "receive" mapping

 Engine_Fuel
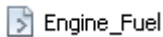
2. Variables is mapped in at least one "transmit" mapping

 Engine_Speed
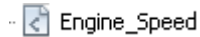
2. Variables is mapped in at least one "receive" and one "transmit" mapping

 Engine_Oil
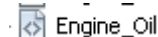
3. Variable tool tip
If the mouse is moved to a variable at the variable tree a tool tip will be shown with the useful information

3a. Information for all variables
- Variable Name
- Group
- Index
- SubIndex
- Access Type
- Data Type
- Length
- Owner

3b. Information for mapped variables
If a variable is mapped, the mapping objects will be shown also at the tooltip.

In this example it is displayed that the variable "Engine Oil" is mapped at one
- Receive Mapping   =>   "**Rx:**   Engine_Rx"
and it is mapped at one

- Transmit Mapping   =>   **"Tx: Engine_Tx"**

4 Variable tree context menu
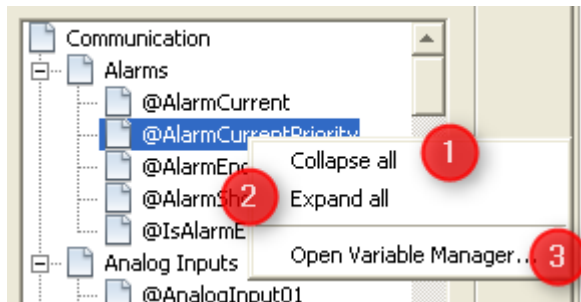
There are 3 options at the context menu of the variable tree

(1) Collapse all
Only the groups of the variable will be shown

(2) Expand all
All groups with all variables will be shown

(3) Open Variable Manager...
The variable manager will be opened and the selected variable of the **variable tree** will be shown. At this dialog you can edit the variable.
If you want to learn more about the **"Variable Manager"** dialog please have a look here.

Note:
If a variable is opened at the variable manager which is already mapped, the variable is not complete editable. Properties like "Name", "Index", "SubIndex", "Data Type" and "Length" are **not editable**.
Once the variable is not mapped anymore, the variable will complete **editable again**.

VCM Table

Overview

 - The column titles shows the bit number of the current byte and the row titles shows the appropriate byte.
 - The table starts in the upper right corner with bit index 1, which represents the first bit of a mapping.
 - At the end and start bit the name of mapped variable / constant is shown.
 - Each mapped variable has its own color. Altogether there are 16 colors for variables and 4 grey colors for constants.
 - Each variable (when selected) can be moved or removed at the table.
 - Selected variables are shown with a bold black border and the color is lightly highlighted.
 - It is possible to select several mapped variables by selecting the first one and then press "Shift" and selecting afterwards another variable.All variables between are selected too.
 - Selected variables are a variable block which can only be moved together.
 - As long as you stay at the view "Visual CAN Mapping" all add, remove, move steps and property changes can be reverted. If you want to undo the last step you can press **"Ctrl + z"** or **"Ctrl + y"** to redo the last step.

Moving variables

To move a variable via drag and drop, you can drag the first bit of the of a variable (**"Bit 1" "Engine_Speed"**) and then move the mouse to **"Bit 9"**)
If you drop the variable on **"Bit 9"** the variable **"Engine Speed"** and the **"Constant"** switch their positions.

After dropping the variable **"Engine_Speed"** at **"Bit 9"** they switch the position

	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	
Byte 1	Constant 8	7	6	5	4	3	2	Constant 1	2
Byte 2	Engine_Speed 16	15	14	13	12	11	10	Engine_Speed 9	1
Byte 3	24	23	22	21	20	19	18	Engine_Temp 17	
Byte 4	Engine_Temp 32	31	30	29	28	27	26	25	
								Engine_Oil	

Now you can move the variable **"Engine_Speed"** to **"Bit 17"**. If you try this you will get following picture **(1)**:

	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	
Byte 1	Constant 8	7	6	5	4	3	2	Constant 1	
Byte 2	16	15	14	13	12	11	10	9	1
Byte 3	Engine_Speed 24	23	22	21	20	19	18	Engine_Speed 17	
Byte 4	Engine_Temp 32	31	30	29	28	27	26	25	
Byte 5								Engine_Oil	

The variables **"Engine_Speed"** and **"Engine_Temp"** can not switch their positions, because variable **"Engine_Temp"** has a length of 16 bits but the gap (Bit 9 - Bit 16) is only 8 bits.
For that reason the moved variable **"Engine_Speed"** is turned to red color, which means that the current position of the variable is not valid. If you try to drop the variable at **"Bit 17"**,
the variable will go back to its last valid position at **"Bit 9"**.

If you drop the variable **"Engine Speed"** at **"Bit 25"**, **"Engine Speed"** and **"Engine_Temp"** will switch their positions **(1)**.

	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	
Byte 1	Constant 8	7	6	5	4	3	2	Constant 1	
Byte 2	16	15	14	13	12	11	10	Engine_Temp 9	
Byte 3	Engine_Temp 24	23	22	21	20	19	18	17	
Byte 4	Engine_Speed 32	31	30	29	28	27	26	Engine_Speed 25	1
Byte 5								Engine_Oil	

Note:
Every time if the moved variable(s) can not be dropped on the current mouse position the variables will be turned to red color to demonstrate that the current position is invalid.
The variable(s) can only be moved if the gap is big enough to switch the variables.
If you want to move the variables e.g. to add a gap you have to select all variables and move them together.

Deleting variables

If you want to delete some variables from the **VCM table,** you can select **(1)** them and then drop them outside of the table.

The variables will turn red if they are dragged out of the table. If you drop them outside they will be deleted.

The position of the other variables will not be changed after the deleting. (Except for CANopen (**CANopen does not allow gaps**): For CANopen the other variables will be moved to the beginning of the table.

VCM table context menu
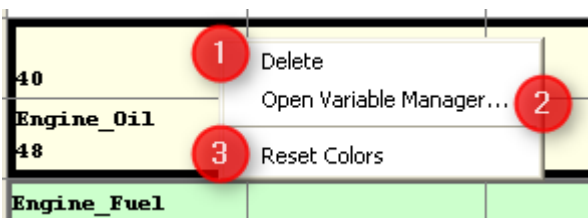
There are 3 options at the context menu of the VCM table

(1) Delete
The selected variable(s) will be deleted.

(2) Open Variable Manager...
The variable manager will be opened and the selected variable of the variable tree will be shown. At this dialog you can edit the variable.
If you want to learn more about the "Variable Manager" dialog please have a look here[165] .

(3) Reset Colors
Sometimes it can happen that variables have the same color. So it is difficult to differ them.
In that case the colors can be recalculated and all variables get different colors again.

Note:
The colors of mapped variables will **not** be **stored**.
If the tool is closed and started again, it could be that colors of mapped variables changes.

VCM table tool tip

If the mouse hovers over a variable at the variable tree, a tool tip will be shown with the useful information

```
Variable Name:   Engine_Oil
---------------------------------
Group:           Common
Length:          16
Mask:            0xffff
Offset:          0.0
Shift X Bits:    0
Scale:           1.0
Offset2:         0.0
---------------------------------
Bit Index:       40
```

Information for all mapped variables
  - Variable Name
  - Group
  - Length (Mapped Length)
  - Mask
  - Offset
  - Shift X Bits
  - Scale
  - Offset 2

Attention symbol of mapped variables

If a variable is mapped to a mapping object and afterwards the length is reduced, the mapped variable gets an attention symbol (**(1,2)** ⚠).
This should help to identify mapped variables with a wrong length. If everything is fine, the attention icon can be ignored.

At the tool tip **(3)** you can see the **original length (16 bit)** and the **mapped length (8 bit)**.

VCM Configure Mapping Table

To change a property of a variable the "Configure Selected Variable Mapping" (**(2) VCM Configure Mapping Table**) table can be used.
The **VCM Configure Mapping Table** is only filled if only **one variable** is selected **(1)**. If no variable or several variables are selected the **VCM Configure Mapping Table** keeps **empty**.

You can easily change the values at the **VCM Configure Mapping Table** like at the view **"Configure Mapping"**.
If the **"Byte Bit Position" (2)** is changed the **VCM table** view will updated in real time. The same happens if the **"Length"** is updated.
It is also possible to make a current variable to a constant or change the variable.

For more information about the fields **"Mask"**, **"Offset1"**, ... please have a look at the appropriate protocols³⁰⁷ .

Object Reference

6   Object Reference

The design tool provides various objects that can be used to design a project.
We can divide the objects in two main categories: General (or tree) objects and palette objects.

General Objects

The general objects are the "root" objects and can be found in the project tree. The following general objects are available:

- Page
- Alarm[443]
- Virtual Keyboard
- Softkey[449]
- Softkey Frame[448]

Palette Objects

Palette objects are "child" objects that can be dragged from a palette into root objects or other palette objects to create object hierarchy.
The palette objects contain again two categories: Containers and the objects that are to be placed inside a container:

Container Objects

Frame[467]

Container[466]

Common Controls
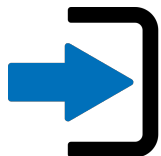
Button[463]

String Field[504]

Numeric Field[495]

Picture Graphic[496]
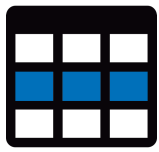
[Table](505)

[List](477)

Measurement Controls

[Meter](481)

[Gauge](504)

[Gauge 270](467)

[Gauge 180](467)

[Gauge 90](467)

[Arched Bargraph](461)
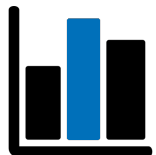
[Linear Bargraph](475)

[Graph](451)

Lamps and Switches

[Lamp](#)[473]

[Power Switch](#)[499]

[Push Switch](#)[501]

[Rocker Switch Horizontal](#)[503]

[Rocker Switch](#)[503]

Special Controls

[Multimedia Player](#)[484]

The basics - Create a page and add objects to it

After creating a project, one page will be automatically created. To create a new page, do a right click on the **Pages** node in the project tree and select **New Page**.

To open a page double click on the page in the project tree. The page will then open in the designer area.

To add objects to a page, a frame has to be added to the page. To do that drag a frame from the Palette's Container section to the page in the designer area.

Note: Only frames can be added to pages directly. All other objects can only be dragged in a frame.

After that, any other object from the palette can be dragged to the frame.

Note: Objects can also be dragged and dropped to the objects in the project tree.

Properties: Change behavior and Look & Feel

After objects have been added, they can be customized. Customization happens by changing the [object's properties](#)[415] .

There are a lot of properties that are shared by a lot of object types and there are object specific properties.

A list of common properties can be found [here](#)[415].

To change behavior of an object, it is possible to configure actions for certain events. See [here](#)[430] for a list of events and actions.

For object specific objects visit the links to each object above.

6.1   Common Object Properties

Here you can find a list of all properties. A lot of the properties are shared between objects.
For object specific properties please refer to the [linked object descriptions](#)[415].
Some of the default values may differ between the different object themes.

General properties for all objects

These properties are common for all objects (see exceptions):

ID

Every object in a project has a unique ID. The ID is automatically generated and can not be changed.
The ID is only important if a property of an object shall be changed or read via
[JavaScript](#)[514].

Type ID

The Type ID is just a numeric representation of the object type. It can not be changed.

Object Type	Type ID
Page	1
Alarm	2
Container	3
Softkey Frame	4
Softkey	5
Button	6
String Field	8
Numeric Field	9
List	10
Meter	17
Linear Bargraph	18
Arched Bargraph	19
Picture Graphic	20
Frame	256
Table	258
Virtual Keyboard	400
2D Graph	259
Multimedia Player	261
Gauge	262

Sub Type ID

The Type ID is a numeric representation of the object sub type. It can not be changed. The sub type ID is only existent for objects belonging to one parent Type ID.

Object Type	Type ID	Sub Type ID
Button		-1
Power Switch	6	0
Push Switch		1

Object Type	Type ID	Sub Type ID
Rocker Switch		3
Lamp		5
Gauge	262	-1
Gauge 270		0
Gauge 180		1
Gauge 90		2

Name

A string containing the name of an object. It is automatically created but it is strongly recommended that the name should be changed to match the purpose of the object. This makes the content of the project more clear and helps to find objects better. The object name is by default not used on the device (although it can be read by JavaScript).
Default value: Automatically generated name for the respective object type followed by a number
Possible values: Any string
Objects can be renamed in the project tree by selecting them and pressing F2, or clicking on them again after selecting them.

X Location

The X location of the object relative to it's parent object. The position of an object can be changed with this property.
Not applicable to the page, alarm and virtual keyboard objects since their position is always at X = 0 and Y = 0
Default value: Position of the object where it was dropped on parent object in the designer area or 0 when dropped to parent object in object tree.
Possible values:
- Minimum:
  - - object width
  - exceptions:
    - objects that contain other objects (Frame, Container, Button, List): They can't be moved out of it's parent.
    - Softkey Frame top or bottom. X is always 0.
    - Softkey. Fixed position inside a softkey frame
- Maximum:
  - parent width
  - exceptions:
    - objects that contain other objects (Frame, Container, Button, List): They can't be moved out of it's parent.
    - Softkey Frame top or bottom. X is always 0.
    - Softkey. Fixed position inside a softkey frame

Y Location

The Y location of the object relative to it's parent object. The position of an object can be changed with this property.
Not applicable to the page, alarm and virtual keyboard objects since their position is always at X = 0 and Y = 0

Default value: Position of the object where it was dropped on parent object in the designer area or 0 when dropped to parent object in object tree.
Possible values:
- Minimum:
  - - object height
  - exceptions:
    - objects that contain other objects (Frame, Container, Button, List): They can't be moved out of it's parent.
    - Softkey Frame left or right. Y is always 0.
    - Softkey. Fixed position inside a softkey frame
- Maximum:
  - parent height
  - exceptions:
    - objects that contain other objects (Frame, Container, Button, List): They can't be moved out of it's parent.
    - Softkey Frame left or right. Y is always 0.
    - Softkey. Fixed position inside a softkey frame

Width

The width of an object. The width can be changed for any object except for pages, alarms, virtual keyboards, softkeys (always same as softkey frame width in a left or right softkey frame) and top or bottom softkey frames.
The width of a meter is always the same as it's height (always square).
Default value: object type dependent
Possible values:
- 0 ... 2560 pixel
- 0 ... half page width for left or right softkey frames

Height

The height of an object. The height can be changed for any object except for pages, alarms, virtual keyboards, softkeys (always same as softkey frame height in a top or bottom softkey frame) and left or right softkey frames.
The height of a meter is always the same as it's width (always square).
Default value: object type dependent
Possible values:
- 0 ... 2560 pixel
- The height of a List object can only be set with it's property **Item Height**
- 0 ... half page height for a top or bottom softkey frame

Dependency

An object can either be language or unit dependent or not at all.
The following objects do NOT support dependency: Alarm, Button, Container, Frame, List, Page, Table, Virtual Keyboard.
For more about language dependency see [here](#)[98].
For more about unit dependency see [here](#)[98].

Unit Group

If an object's dependency (see above) is set to Unit, the drop down box for Unit Group is enabled. The type of unit that the object represents can then be selected.
For more about units see [here](#)[98].

Default Value: Force units
Possible Values:
- Force units
- Distance units
- Mass
- Volume
- Area
- Date format
- Temperature
- Pressure
- Units system
- Time format

Variable References

For most of the objects one or more references to a variable can be created. The object will then show the value of the variable at runtime (or shows a invalid state if the referenced variable is invalid) or the max or min value of a DDO can be adjusted with variables. See below for a description of the possible Variable Reference properties.

Variable Reference

The Variable Reference property is used to reference to a variable that shall deliver the value of the selected object.
A integer variable can be assigned as variable reference to the following objects:
- Button (value 0 for button release, any other value for button press)
- Softkey (value 0 for softkey release, any other value for softkey press)
- Arched Bargraph (fill the bargraph or move the value line)
- Linear Bargraph (fill the bargraph or move the value line)
- Meter (move the needle)
- Numeric Field (show the value as a number, see object specific properties for how it can be formatted)
- Table (scroll the table)
- List (select index to be shown)

A string variable can only be assigned to a String Field object.
All other objects do not support variable references.
Default value: None
Possible Values: Either any integer or string variable depending on object type

Min Value Variable Reference

When selecting a variable for this property, the value of the variable controls the Min Value of that object.
This property is supported by the following objects:
- Arched Bargraph
- Linear Bargraph
- Meter
- Numeric Field

Default value: None
Possible Values: Any integer variable

The Absolute Min Value can also be changed through the [JavaScript function setProperty.](#)[538]

Max Value Variable Reference

When selecting a variable for this property, the value of the variable controls the Max Value of that object.
This property is supported by the following objects:

- Arched Bargraph
- Linear Bargraph
- Meter
- Numeric Field

Default value: None
Possible Values: Any integer variable
The Absolute Max Value can also be changed through the [JavaScript function setProperty.](#)[538]

Target Value Variable Reference

When selecting a variable for this property, the value of the variable controls the Target Value of that object.
This property is supported by the following objects:

- Arched Bargraph
- Linear Bargraph

Default value: None
Possible Values: Any integer variable

Bit Mask

With this property a Bit Mask can be applied to the incoming variable value. This makes it possible to only pass certain bits from the variable's value to the object.
For example: A 16Bit variable is referenced to the object but only the value 0 or 1 from the last bit is of interest for the object. In this case a mask of 0x1 can be set, if the last 4Bits shall be used, the mask 0xf can be set.
When changing the property by clicking on the ... button besides the bit mask value, a dialog will appear that makes it easy to set the mask. The number of bits and their position can be chosen in that dialog.

Default value: 0xffffffff (no bit mask)

Possible values: 0x1 ... 0xffffffff

Skin Properties

The Skin Properties influence the look of the objects. Most of these properties are shared for almost all objects. The one exception is the Container object. It is not visible and therefore has no Skin Properties.

Background Color (Theme dependent)

As the name of this property implies, the background color of the selected object can be changed.
Any 24Bit RGB color can be chosen in the color picker dialog that pops up when clicking the ... button besides the color values. It is also possible to enter the three values for ref, green and blue directly without using the color picker (e.g. 255,0,0 for full red).
Default value: Object type dependent
Possible values: any 24Bit RGB color
Note: If you choose a (partly) transparent background image (see below), the background color will be visible in the transparent parts of the background image. If you don't want to see the background color, enable property Transparent (see below).

Background Image (Theme dependent)

A background image can be set for an object suing this property.
Any supported image type can be selected in the image chooser dialog that pops up when clicking the ... button of the property.
A chosen image will be automatically scaled to the size of the object and converted into the lossless png format (except for SVG image types, they will not be changed).
Default value: no image
Possible values: any image in the supported format (BMP, JPEG, PNG, SVG)
Note: If you choose a (partly) transparent background image, the background color (see above) will be visible in the transparent parts of the background image. If you don't want to see the background color, enable property Transparent (see below).

Draw Border (Theme dependent)

With this property the drawing of the border of an object can be enabled or disabled. The look of the border can be changed with the other border properties below.
Note: If the object is part of the navigation (e.g. an input), the drawing of the border should be enabled so that the user can distinguish between being selected or open for input.
Default value: enabled
Possible values: enabled or disabled

Border Width (Theme dependent)

The width of the border of the object can be set with this property. See also properties Border Image and Border State properties for how to define the look of the border.
Note: If you don't want any border, disable it with property Draw Border
Default value: 3
Possible values: 1 ... 15

Border Image (Theme dependent)

It is possible to display a border image for an object. However, the border profile will be painted over the border image. Be sure to check the property Border Width and the border profile (see above) so that the border has the desired size and is still visible with the border profile.

Default value: no image
Possible values: any image in the supported format (BMP, JPEG, PNG, SVG)

Border State

Using this property a preview of the different Border Profiles can be activated. This property has no influence at runtime since the different border profiles are chosen according to object's navigation state at runtime.
Default value: Enabled Without Focus
Possible values:

- Open For Input
- Enabled With Focus
- Enabled Without Focus
- Disabled

Border Profile

When changing this property the border profile editor will pop up. In this dialog it is possible to define a border profile for each navigation state of the object. Clicking on the ... button besides each state, the editor for the border profile itself pops up.
A border profile is a sequence of lines that are drawn the object. Each line can have a different color. There can be up to 15 lines (see property Border Width above) in each profile.
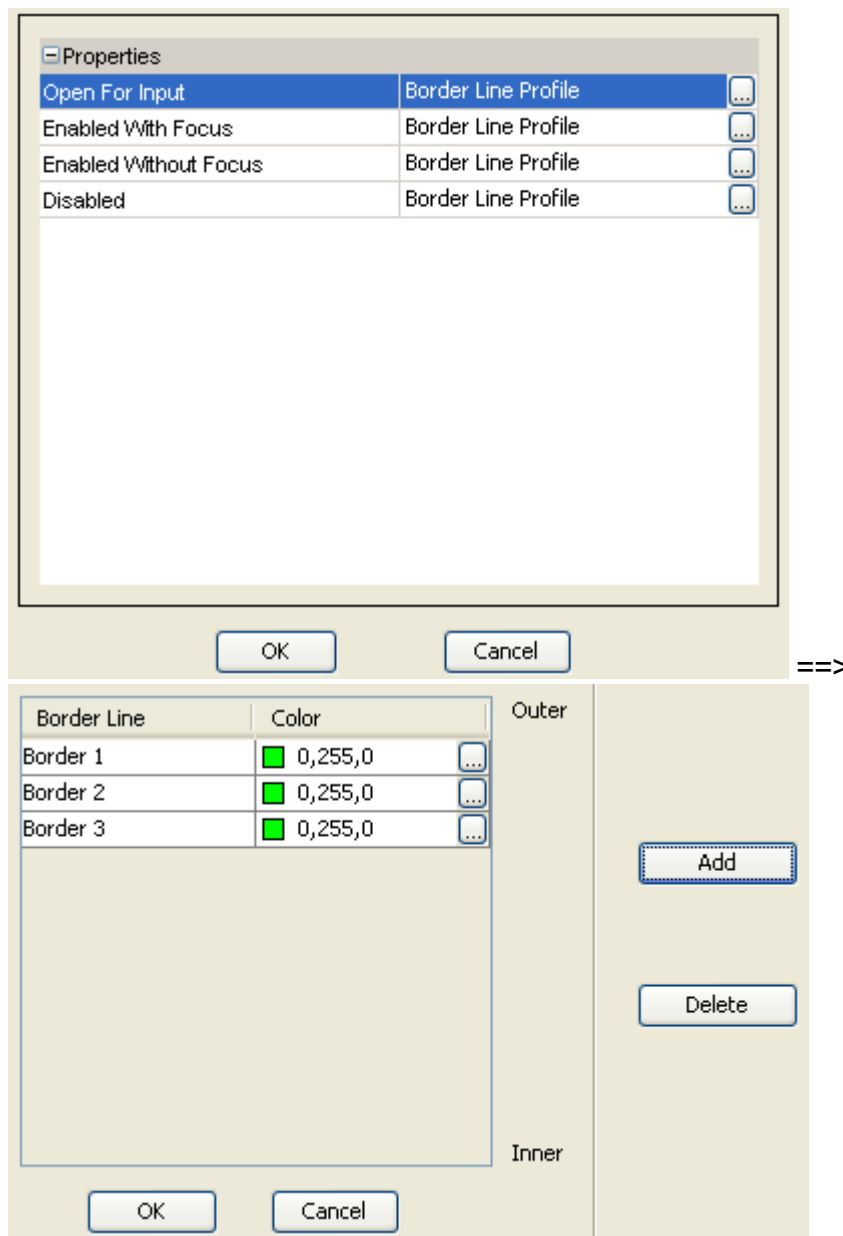
Transparent Border Color

With the Transparent Border Color it is possible to make a border profile of a certain state or single lines of a border profile invisible.

A relatively common use case is: Make the color that is used in the state Enabled Without Focus the same as the transparent border color. Like this the object will only have a visible border if it has focus or is open for input.

Default value: Depending on the selected theme when creating a project. It's blue for the modern themes, black for classic theme. Change the default transparent border color for all objects of a project in the Project Properties[84].

Possible values: any 24Bit RGB color

Transparent (Theme dependent)

The Transparent property can be used to make the background of an object invisible. If the property is disabled, the Background Color will be drawn as the background of the object. If the property is enabled, no background color will be drawn.
Note: If a Background Image is assigned, it will be shown also if the transparent property is enabled. This property only affects the background color.
Known Issue: Currently this property has no effect on Gauge objects if the property Draw Boundary Bar is activated.
Default value: disabled
Possible values: enabled or disabled

Corner Radius (Theme dependent)

The corner radius property can be used to give Numeric Fields and String Fields a rounded corner. With a value of 0 the objects will be rectangular. With a value of 1000 the maximum roundness is set.
Default value: 50
Possible values: 0 - 1000
Please note that even though the objects don't have a rectangular shape anymore, the objects themselves are still rectangular, and so are the borders for the objects.

Visibility

The visibility of an object can be configured with the visibility properties. Objects can be made visible or invisible during runtime.
For more information about visibility see here[588] .

Visibility

Enable or disable visibility for the selected object.
Default value: Disabled
Possible values: Enabled or disabled

Visibility Bit

With this property it can be defined to which bit of which visibility variable this object listens or, alternatively, which variable should be used as a visibility switch.
For more information see here[588] .
Default value: Value not set
Possible values: A configured visibility bit of a visibility variable or a variable name.

Input Configuration

Set As Input

Setting the Set As Input attribute to enabled makes an object editable by the user. How the object reacts to inputs can then be configured in the other input properties below.
This property is only available for objects that can reference a variable. It is not available for Page, Frame, Container, Virtual Keyboard, Softkey Frame, Softkey.
Default value:
- Enabled for button objects
- Disabled for all other objects

Possible values: Enabled or disabled

Connect To Encoder

An object can directly connected to the encoder. This means that the user does not need to select the object and make it open for input.
The consequence of enabling this property is, that this object is the only input object on the current page.
The following dialog will pop up when enabling this property:
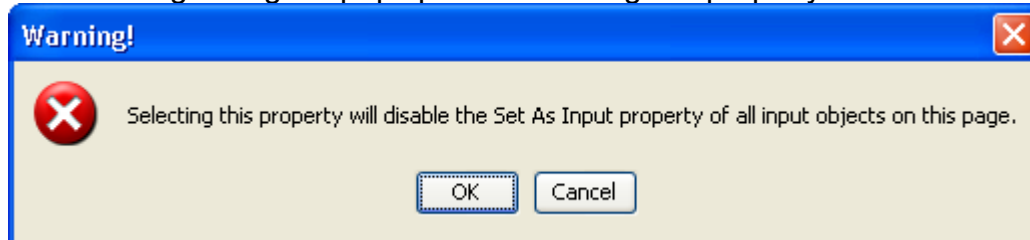
When clicking OK, the Set As Input property of all other objects on the current page will be disabled. Clicking cancel will keep the Connect To Encoder Property disabled.
Default value: Disabled
Possible values: Enabled or disabled
Note that the Send Value Directly property (see below) should also be enabled when enabling the Connect To Encoder property.
The property is only available if Set As Input for this object is enabled.

Send Value Directly

If Send Value Directly is enabled, the value that the user enters via encoder is transmitted to the variable (and possibly CAN bus) with every value change the user does.
If this property is disabled, the value will be set only after the user has finished editing (e.g. pressing encoder).
Send Value Directly could also be called *Live Editing*. Enable this if you want to have every value change by the user be reflected in an action (e.g. setting a value that's result can directly be seen, e.g. moving something physically at the vehicle, changing display backlight intensity, etc.). Disable the property if you are only interested in the final value that the user sets.
Default value: Disabled
Possible values: Enabled or disabled.
This property is only available for objects that work with integer values.
The property is only available if Set As Input for this object is enabled.

Encoder Movement

There are two options available:
- Linear:
  Every encoder click results in a value change of 1 (or -1 depending on direction)
- Dynamic
  Based on the speed of the encoder turns, the value increment or decrement is dynamically accelerated.

Use dynamic encoder movement if a very high value range is set for the object.
The amount of dynamic encoder acceleration can be configured with the **Value Change Factor** property (see below).
Default value: Linear
Possible values: Linear or Dynamic
The property is only available if Set As Input for this object is enabled.

Value Change Factor

This property only has influence if the Encoder Movement property (see above) is set to Dynamic. The Value Change Factor controls the amount of acceleration for dynamic encoder movement.
The higher the value, the more the value change by encoder is accelerated.
Rule of thumb: The higher the value range of an object is, the higher the Value Change Factor should be set.
Note: When simulating the project with the PClient simulation on the development PC, you can test the effect of this property using the mouse wheel.
Default value: 1000
Possible values: 1 ... 32767
The property is only available if Set As Input for this object is enabled.

Input Character Attribute

This property is only available for the string field object. With the Input Character Attribute it can be defined what characters are available when the user edits the value of the string field.
It is possible to restrict the user input to certain characters defined in the selected Input Character Attribute.
How an Input Character Attribute can be defined can be seen here[76] .
If no Character Input Attribute is selected, the default ASCII character set will be available as input characters for the user.
Default value: None
Possible values: Any Input Character Attribute defined in the project properties[76] .
The property is only available if Set As Input for this object is enabled.

Inplace Editor

The Inplace Editor property is only available for String Field and Numeric Field objects. With this property it can be chosen if the value of the object shall be changeable via encoder or by using a virtual keyboard[446] .
Default value: Encoder
Possible values: Encoder or Virtual Keyboard
The property is only available if Set As Input for this object is enabled.

Value Related

Except for **Preview Value**, all value related properties are only available for objects working with numeric values.

Absolute Max Value

As the name implies, the Absolute Max Value is the highest value that a user can set to an object (Max Value is not, see below).
This property is available only for Arched Bargraph, Linear Bargraph, Meter and Numeric Field objects.
Following rule applies: Absolute Max Value needs to be greater or equal to Max Value.
Default value: depends on object:
- Arched Bargraph and Meter: 360
- Linear Bargraph and Numeric Field: 100

Possible values: Anything greater or equal to Max Value but not more than 2147483647
(Absolute Min Value <= Min Value <= Max Value <= Absolute Max Value)

Color Above Maximum

This property defines the color of the object if the value of the object is greater than Max Value (see below).
Color Above Max Value is available for the following objects with the following appearance effect:

- Arched Bargraph: Fill color of the bargraph or value line color
- Linear Bargraph: Fill color of the bargraph or value line color
- Meter: Needle color
- Numeric Field: Color of the displayed number

Default value: object specific
Possible values: Any 24Bit RGB color

Max Value

The Max Value property is a threshold for doing the color change to **Color Above Maximum** when the value of the object is greater than the value of Max Value.
This property is available only for Arched Bargraph, Linear Bargraph, Meter and Numeric Field objects.
Default value: depends on object:

- Arched Bargraph and Meter: 300
- Linear Bargraph: 80
- Numeric Field: 90

Possible values: Any value greater than or equal to Min Value and less than or equal to Absolute Max Value (Absolute Min Value <= Min Value <= Max Value <= Absolute Max Value)

Font Attribute Above Max Value (Theme dependent)

This property defines the Font Attribute that is used when the value of the object is above Max Value.
This property is only available for the Numeric Field object.
To learn more about Font Attribute Objects see [here][76].
Default value: Default_DejaVu Sans
Possible values: Any Font Attribute Object defined in [project properties][76].

Min Value

The Min Value property is a threshold for doing the color change to **Color Below Minimum** when the value of the object is less than the value of Min Value.
This property is available only for Arched Bargraph, Linear Bargraph, Meter and Numeric Field objects.
Default value: depends on object:

- Arched Bargraph, Linear Bargraph and Meter: 20
- Numeric Field: 10

Possible values: Any value greater than or equal to Absolute Min Value and less than or equal to Max Value (Absolute Min Value <= Min Value <= Max Value <= Absolute Max Value)

Font Attribute Below Min Value (Theme dependent)

This property defines the Font Attribute that is used when the value of the object is below Min Value.
This property is only available for the Numeric Field object.
To learn more about Font Attribute Objects see [here][76].
Default value: Default_DejaVu Sans

Possible values: Any Font Attribute Object defined in [project properties](#)□76 .

Absolute Min Value

As the name implies, the Absolute Min Value is the lowest value that a user can set to an object (Min Value is not, see above).
This property is available only for Arched Bargraph, Linear Bargraph, Meter and Numeric Field objects.
Following rule applies: Absolute Max Value needs to be greater or equal to Max Value.
Default value: 0
Possible values: Anything less or equal to Min Value but not less than -2147483648
(Absolute Min Value <= Min Value <= Max Value <= Absolute Max Value)

Color Below Minimum

This property defines the color of the object if the value of the object is less than Min Value (see below).
Color Below Min Value is available for the following objects with the following appearance effect:

- Arched Bargraph: Fill color of the bargraph or value line color
- Linear Bargraph: Fill color of the bargraph or value line color
- Meter: Needle color
- Numeric Field: Color of the displayed number

Default value: object specific
Possible values: Any 24Bit RGB color

Preview Value

With the Preview Value property, the value of the object in the designer area can be previewed. If no variable is referenced to the object, the Preview Value will also be the value that is displayed at the object at runtime.
The Preview Value property is available for all objects except for the button object (use Pressed State button property for preview).
Default Value: object specific
Possible values:

- String field: Any string that is shorter than Max String Length (characters)
- Integer objects: A value greater than or equal to Absolute Min Value and less than or equal to Absolute Max Value (Absolute Min Value <= Preview Value <= Absolute Max Value)

Target Value

The Target Value property is only available for Arsched Bargraph and Linear Bargraph objects. If the Draw Target Line property for these objects is enabled, a line will be drawn at the Target Value position.
Default value: 80
Possible values: A value greater than or equal to Absolute Min Value and less than or equal to Absolute Max Value (Absolute Min Value <= Target Value <= Absolute Max Value)

Activate Low Pass Filter

With this property a low pass filter for the selected object can be activated. A low pass filter makes the displayed value changes more smooth and can reduce flickering of values.
To adjust the strength of the filter use the property Filter Factor (see below).
The formula for the low pass filter is:

CurrentValueToDisplay = ((oldValue * filterFactor) + (meassuredValue * (1000 - filterFactor))) / 1000
This calculation is made in a relatively high frequency until currentValueToDisplay is equal to meassuredValue.
Default value: Disabled
Possible values Enabled or disabled
This property is available only for Arched Bargraph, Linear Bargraph, Meter and Numeric Field objects.

Filter Factor

The Filter Factor property is responsible for controlling the strength of the low pass filter (see Activate Low Pass Filter above).
The higher the value is, the stronger the value changes will be filtered. Best way to find the best a good value for an object is to try and test some different values.
Default value: 500
Possible values: 0 (almost no filter) ... 1000 (very strong filter)
This property is available only for Arched Bargraph, Linear Bargraph, Meter and Numeric Field objects if the Activate Low Pass Filter property is enabled.

Step Width

The Step Width property can be used for rounding the value of a Numeric Field. It is considered both for input by the user and for displaying variable values.
If e.g. a step with of 10 is set, the user can only set values that are multiple of 10 (0, 10, 20, 30, etc.). Also the value of a referenced variable will be rounded (e.g. variable value: 7 => displayed value: 10, variable value: 42, displayed value: 40).
Default value: 1
Possible values: 1 ... 2147483647

Formatting

Font Attribute (Theme dependent)

A Font Attribute provides the size color and style (standard, bold, italic) of how a font shall be displayed. Font attributes can be created in the project properties[76].
With the Font Attribute property one of the created Font Attribute Objects can be assigned to the selected object.
If the font of the object shall be changed use property Font (see below).
Default value: Default_DejaVu Serif
Possible values: Any Font Attribute object defined in project properties[76].
This property is only available for Numeric Field, String Field and table objects.

Font

The font that shall be displayed in a Numeric Field, String Field or a Table can be selected using the Font property.
Note that the size and style of the font is controlled via the Font Attribute property (see above).
Default value:
- Numeric Field: DejaVu Sans
- String Field: DejaVu Serif
Possible values: Any pre defined or imported font[76].

Horizontal Alignment

With the Horizontal Alignment property it can be controlled if the text is aligned left, right or centered.
Default value: Center
Possible values: Left, Center, Right
This property is only available for Numeric Field, String Field and Table object.

Vertical Alignment

With the Vertical Alignment property it can be controlled if the text is aligned on top, bottom or centered.
Default value: Center
Possible values: Top, Center, Bottom
This property is only available for Numeric Field, String Field and Table object.

Font Outlined

This property is deprecated and currently has no effect. It will be removed in a future release.

Object specific properties

Click the links below to view the description and properties of the object

2D-Graph 451

Alarm 443

Arched Bargraph 461

Button 463

Container 466

Frame 467

Linear Bargraph 475

List 477

Meter 481

Numeric Field 495

Page 443

Picture Graphic 496

Softkey Frame 448

Softkey 449

String Field 504

Table 505
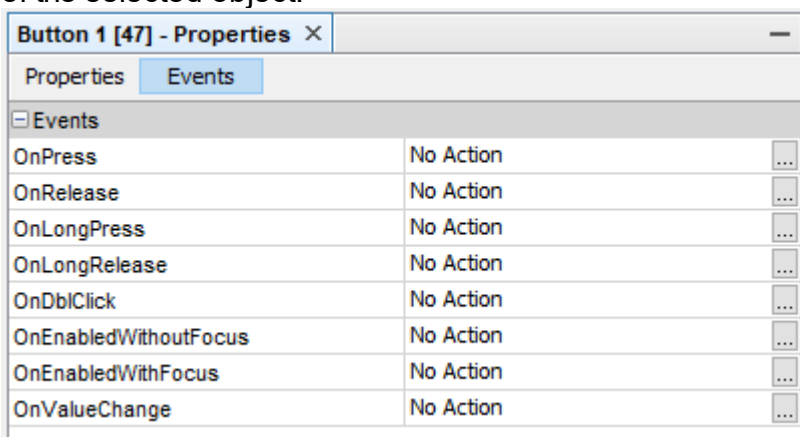
Virtual Keybard[446]

6.2   Events and Actions

With events and the corresponding actions, dynamic behavior can be added to a project.
Almost all objects support a certain set of events. For these events certain actions can be configured.
This chapter lists a description of all available events and actions. What event and what action is supported by an object, can be found in the documentation for that object.
To assign an action to the event of an object, select the *Events* tab on top of the property table of the selected object.

Button 1 [47] - Properties ✕		—
Properties	Events	
⊟ Events		
OnPress	No Action	...
OnRelease	No Action	...
OnLongPress	No Action	...
OnLongRelease	No Action	...
OnDblClick	No Action	...
OnEnabledWithoutFocus	No Action	...
OnEnabledWithFocus	No Action	...
OnValueChange	No Action	...

Then press the ... button of the event that shall be configured.
Note: Events that are specific to CAN protocols are explained in the protocol's manual chapter.

Event Options

For some events a repeat period can be configured (e.g. OnProjektRepeat or OnPageRepeat).
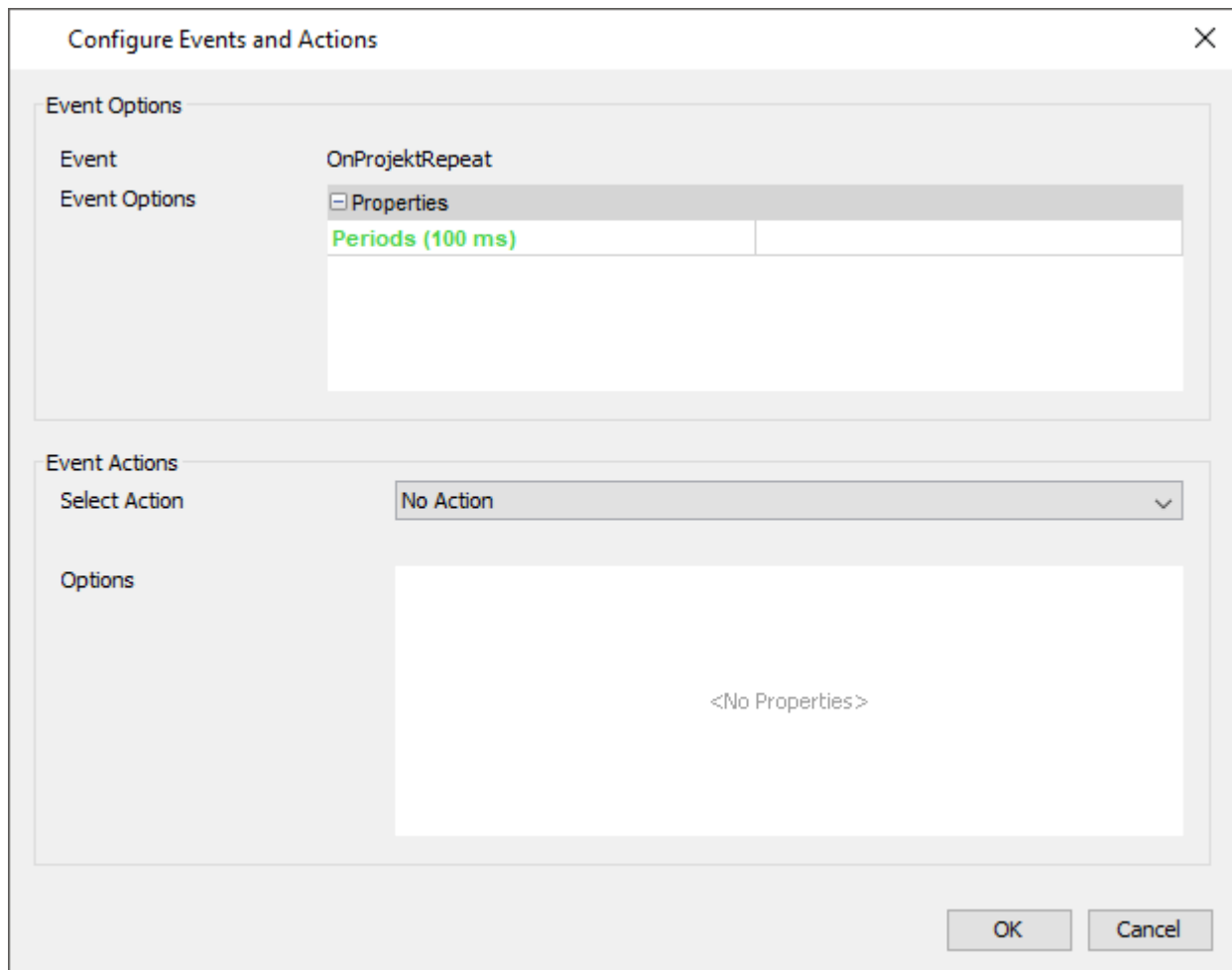
This period is always a multiple of 100 milliseconds (which means that the fastest period is 100ms). It is not guaranteed that the actions can be triggered exactly in the specified period, a slight deviation is possible (based on CPU load and other circumstances at runtime). A value has to be entered in this field for the periodic execution to work.
If a device is equipped with an internal sound card (speaker and/or line-out), an audio file can be played back when certain events happen.
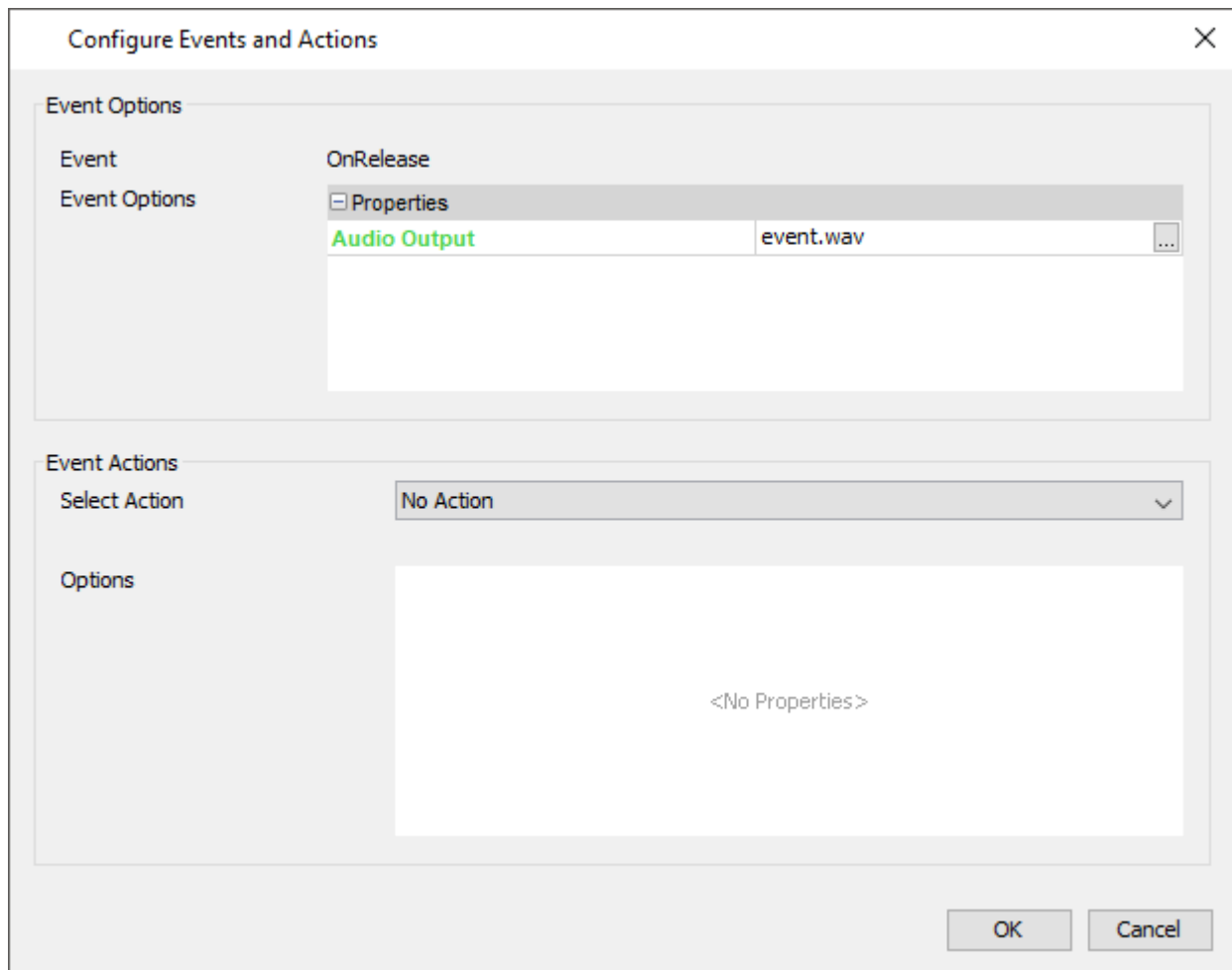
Audio Output
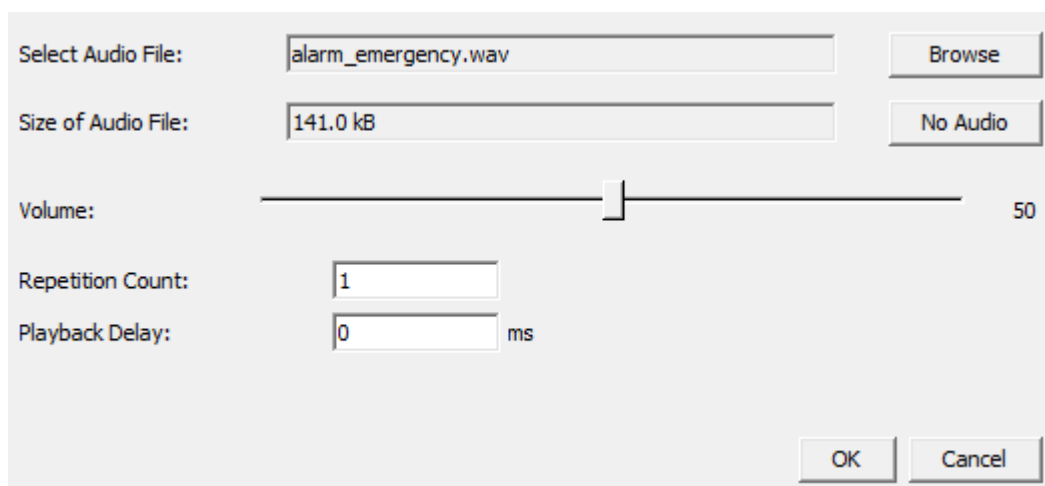
The Audio Output dialog allows to assign an audio file that will be played back when the event occurs.

The following elements of the dialog can be used for configuration:

- *Select Audio File*: Use the *Browse* button to select an audio file from the file system (There are different default audio files pre-configured for an event or alarms that can be directly used)
- *No Audio button*: Press this button if no audio shall be played for the event.
- *Size of Audio File*: This shows the size of the selected audio file

(very large files might add a delay before they are played and increase the size of the project package that needs to be transfered to the device)

- *Volume*: Select the desired volume with the slider. Note that the final volume is dependent on the values of the [beeper volume variables](#)[174] .
- *Repetition Count*: How often the audio file shall be repeated can be configured here (1 means that the audio file is played once). Please note that the file is not saved if the repetition count is set to 0.
- *Playback Delay*: If a repetition count greater than 1 is defined (see above), a playback gap can be configured. Enter a time in milliseconds that defines the gap (pause) between each repetition.

The following audio codecs have been verified on the device:

- MPEG AAC Audio (mp4a) (usually the file extension is .m4a or .aac, but .mp4 also works, if there is no video stream)
- MPEG Audio layer 1/2/3 (mpga) (usually the file extension is .mp3)
- Vorbis Audio (vorb) (usually the file extension is .oga or .ogg)
- Raw wave files in different format (tested: PCM S16 LE (araw), PCM U8 (araw)) (usually the file extension is .wav)

Windows Media Audio (.wma) is currently **NOT** supported.

Media files that include a video stream do **NOT** work for these events. Please use a [Multimedia Player](#)[484] for this.

Please note that the Audio Output settings are not saved if no event is configured for the according action.

Events

OnProjektInit

This event is only triggered once when the project is loaded before any other event.
Allowed actions:

- Execute Script

OnProjektInitFinished

This event is only triggered once when the loading of the project has finished.
Allowed actions:

- Execute Script

OnProjektExit

This event is only triggered once when the PClient is exited on the device. This will normally not happen during normal operation, only when updating the project on the device.
Allowed actions:

- Execute Script

OnProjektRepeat

This event is triggered repeatedly with a configurable period. This is the only repeat event that is triggered regardless of the active page (see OnPageRepeat).
Allowed Actions:

- Execute Script

OnPowerOn

This event is triggered as soon as the device enters the On (normal operation) state but only if it was in low power or sleep mode before (switch ignition on). If the device just did a cold boot, this event is not triggered.
Allowed actions:

- Execute Script
- Set Value

OnLowPower

This event is triggered as soon as the ignition is switched off and the device enters low power mode.
Allowed actions:

- Execute Script
- Set Value

OnSleep

This event is triggered as soon as the device awakes from sleep mode (not before going into sleep).
Allowed actions:

- Execute Script
- Set Value

OnPageInit

This event is triggered as soon as a new page is loaded.
Allowed actions:

- Execute Script

OnPageInitFinished

This event is triggered as soon as the loading of the new page is finished.
Allowed actions:

- Execute Script

OnPageExit

This event is triggered as soon as a page is left.
Allowed actions:

- Execute Script

OnPageRepeat

This event is periodically triggered when the page is loaded.
Allowed actions:

- Execute Script

OnPress

This event is triggered either if an object on screen is pressed via touchscreen or encoder or if a softkey is pressed.
Allowed actions:

- Execute Script
- Jump To Page
- Set Value (Not for Frame object)
- Next Softkey Level (Softkey object only)
- Previous Softkey Level (Softkey object only)

- Navigate Up (Softkey and function key object only)
- Navigate Down (Softkey and function key object only)
- Navigate Confirm (Softkey and function key object only)
- Open PDF Reader (Not for Frame object)
- Set Media Player Command (only for devices that support the playback of audio / video)

OnRelease

This event is triggered either if an object on screen was pressed via touchscreen or if a softkey was pressed via physical keys and has now been released.
Allowed actions:

- Execute Script
- Jump To Page
- Set Value (Not for Frame object)
- Next Softkey Level (Softkey object only)
- Previous Softkey Level (Softkey object only)
- Navigate Up (Softkey and function key object only)
- Navigate Down (Softkey and function key object only)
- Navigate Confirm (Softkey and function key object only)
- Open For Input (Softkey and Button object only)
- Open PDF Reader (Not for Frame object)
- Set Media Player Command (only for devices that support the playback of audio / video)

OnCancel

This event is triggered if an object on screen was pressed via touchscreen (including softkey regions) and then the finger swiped away from it and then releases.
Allowed actions:

- Execute Script
- Jump To Page
- Set Value (Not for Frame object)
- Next Softkey Level (Softkey object only)
- Previous Softkey Level (Softkey object only)
- Navigate Up (Softkey and function key object only)
- Navigate Down (Softkey and function key object only)
- Navigate Confirm (Softkey and function key object only)
- Open For Input (Softkey and Button object only)
- Open PDF Reader (Not for Frame object)
- Set Media Player Command (only for devices that support the playback of audio / video)

Please note that these events behave differently depending on the state of the Latching[465] property of the according object.
If latching is activated, this event will not be executed on a release after a press, but instead on the second press after a first press has activated the pressed state of the object. In this case, the release of the object and the cancelling will never execute an event.

Please also note that from this version on, devices with capacitive touch will behave differently when sliding away from a button:
When sliding away from a button, in earlier versions of the program, the button would stay pressed. Then, when the finger is released, the OnRelease event would be executed. Now, as soon as the finger leaves the button, the button becomes de-pressed WITHOUT the release event being executed! Additionally, the OnCancel event is executed in that case from version 2022.4.1 onwards.

OnLongPress

This event is triggered either if an object on screen is pressed via touchscreen or if a softkey is pressed via physical keys for longer than the configured Long Press Time (see project properties[76]).
Allowed actions:
- Execute Script
- Jump To Page
- Set Value (Not for Frame object)
- Next Softkey Level (Softkey object only)
- Previous Softkey Level (Softkey object only)
- Navigate Up (Softkey and function key object only)
- Navigate Down (Softkey and function key object only)
- Navigate Confirm (Softkey and function key object only)
- Open PDF Reader (Not for Frame object)
- Set Media Player Command (only for devices that support the playback of audio / video)

OnLongRelease

This event is triggered either if an object on screen is pressed via touchscreen or if a softkey is pressed via physical keys for longer than the configured Long Press Time (see project properties[76]) and was released again.
Allowed actions:
- Execute Script
- Jump To Page
- Set Value (Not for Frame object)
- Next Softkey Level (Softkey object only)
- Previous Softkey Level (Softkey object only)
- Navigate Up (Softkey and function key object only)
- Navigate Down (Softkey and function key object only)
- Navigate Confirm (Softkey and function key object only)
- Open PDF Reader (Not for Frame object)
- Set Media Player Command (only for devices that support the playback of audio / video)

Please note that this event behaves differently depending on the state of the Latching[465] property of the according object.
If latching is activated, this event will be executed at a release after a long press if the pressed state was not activated yet. If the pressed state was activated, the event will not be executed at a release after a long press.

OnDblClick

This event is triggered either if an object on screen is pressed via touchscreen or if a softkey is pressed via physical keys twice within the double click time (300ms).
Allowed actions:

- Execute Script
- Jump To Page
- Set Value (Not for Frame object)
- Next Softkey Level (Softkey object only)
- Previous Softkey Level (Softkey object only)
- Navigate Up (Softkey and function key object only)
- Navigate Down (Softkey and function key object only)
- Navigate Confirm (Softkey and function key object only)
- Open PDF Reader (Not for Frame object)
- Set Media Player Command (only for devices that support the playback of audio / video)

Please note that this event behaves differently depending on the state of the Latching[465] property of the according object.
If latching is activated, this event will never be executed.

Here are the possible scenarios for press/release events when using a button or softkey with latching deactivated:
- Short press, then release -> OnPress and OnRelease are executed
- Long press, then release -> OnPress, then OnLongPress, then OnLongRelease AND OnRelease are executed
- Double click -> OnPress, then OnRelease, then OnDblClick, then again Release are executed

Here are the scenarios for press/release events when using a button or softkey with latching activated:
- Short press, then release -> OnPress is executed on the first press, OnRelease is executed the second press. No actions for the release of the button / softkey.
- Long press, then release -> OnPress, then OnLongPress on the first press. OnLongRelease on the first release. OnRelease on the second press. No action for the second release.
- Double click -> Double click behaves like a short press and release. No double click event will be executed with latching activated.

OnEnabledWithoutFocus

This event is triggered when the object has lost focus (it was either enabled with focus or open for input before).
Allowed actions:

- Execute Script
- Jump To Page
- Set Value

- Set Media Player Command (only for devices that support the playback of audio / video) (only for buttons and softkeys)

OnEnabledWithFocus

This event is triggered when the user navigates to this object so that it gets into the EnabledWithFocus navigation border state. It only has focus but is not open for input.
Allowed actions:
- Execute Script
- Jump To Page
- Set Value
- Set Media Player Command (only for devices that support the playback of audio / video) (only for buttons and softkeys)

OnOpenForInput

This event is triggered as soon as an object is made for open for input to edit it's value.
Allowed actions:
- Execute Script
- Jump To Page
- Set Value

OnValueChange

This event is triggered as soon as the value that the object displays is changed (either by user or by a referenced variable).
Allowed actions:
- Execute Script
- Jump To Page
- Set Value

OnEnterNewValue

This event is triggered after the user has entered a new value for the object (not the same value).
Note: This event is only fired for objects that have the property "Send Value Directly" **disabled**. If "Send Value Directly" is enabled, the event will **not** occur and the assigned action will **not** be executed.
Allowed actions:
- Execute Script
- Jump To Page
- Set Value

OnEnterValue

This event is triggered after the user has finished editing the value of the object regardless if the value has changed or not.
Allowed actions:
- Execute Script
- Jump To Page
- Set Value

OnEscape

The OnEscape event is triggered if an object is open for input and the user pressed the Escape key to cancel the input.
Allowed actions:
- Execute Script
- Jump To Page
- Set Value

OnGetValue

This event is currently not supported.

OnValueChangeByOwner

This is a variable event. It is triggered as soon as the owner of the variable writes a **new** value to the variable. This event can be set in the <u>Variable Manager</u>[165]
Allowed actions:
- Set Value
- Scale Value
- Execute Script

OnValueChangeRequest

This is a variable event. It is triggered as soon as anyone that is not the owner of the variable tries to set a **new** value.
Allowed actions:
- Set Value
- Scale Value
- Execute Script

OnValidStateChange

This is a variable event. It is triggered as soon as the validity status of the variable changes, i.e. the variable goes invalid from valid or vice versa.
Allowed actions:
- Set Value
- Execute Script

Actions

Various actions can be assigned to an event. See above for what action can be configured for which event.

No Action

This is the default setting for any event. To select an action, open the drop-down menu and select one of the available actions. To disable the current action, choose No Action. Please note that when No Action is selected, all settings of the previously selected action are removed.

Execute Script
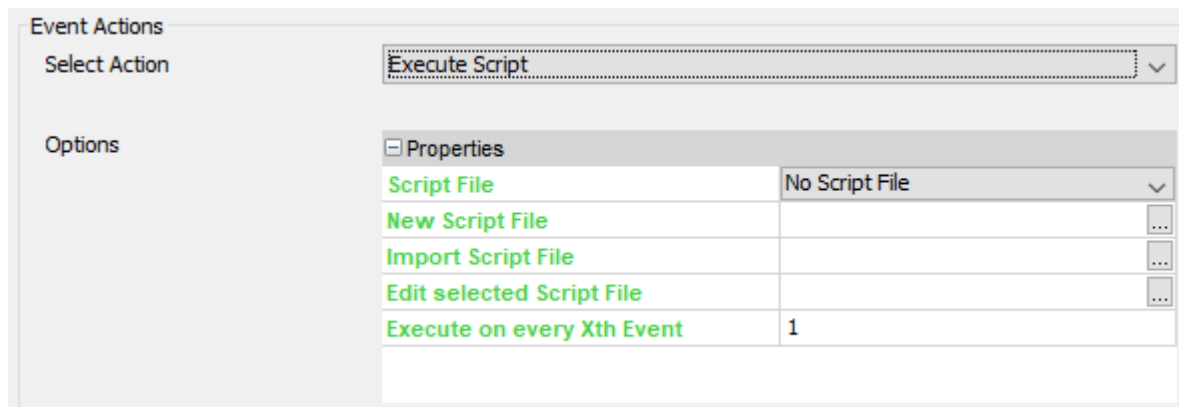
The execute script action can trigger the execution of a JavaScript file. See the JavaScript documentation[514] for more information.
Properties:
- Script File
  The JavaScript file that shall be executed
- New Script File
  Not really a property, rather a short-cut which will open the new script dialog.
- Import Script File
  Not really a property, rather a short-cut to import an existing script file from another location
- Edit selected Script File
  Not really a property, rather a short-cut to open the JavaScript editor with the currently selected script file.
- Execute on every Xth Event
  This property allows the execution of the JavaScript to not happen every time the event is triggered but only at e.g. every 5th event: This can act as kind of a low-pass filter for value change events.

Jump To Page

The jump to page action triggers a change to another page in the project.
Properties:
- Jump To
  Select the page from the drop down list to which the project shall change

Set Value

With the set value action a fixed value can be set to a variable.
Properties:
- Variable
  Select the variable that's value shall be set
- Value
  Either a number or a string depending on the variable type

Scale Value

With this action a value of a variable can be scaled. The following formula is used for that:
scaledValue = (((Value + Offset1) * Multiplier) / Divisor) + Offset2
Properties:
- Offset1
- Multiplier

- Divisor
- Offset2

Next Softkey Level

With this action the next softkey level in a softkey frame can be shown.
Properties: None

Previous Softkey Level

With this action the previous softkey level in a softkey frame can be shown.
Properties: None

Navigate Up

This action simulates one click of the encoder. It moves the focus frame one position up in the object hierarchy.
Properties: None

Navigate Down

This action simulates one click of the encoder. It moves the focus frame one position down in the object hierarchy.
Properties: None

Navigate Confirm

This action simulates an encoder press. It opens an object for input or confirms the value when the object is already open for input.
Properties: None

Set Media Player Command

Properties:
- Media Player ID
  The ID of the Media Player object that shall be controlled.
- Command
  The command that shall be executed. The following commands are available:
  - play
  - stop
  - pause
  - previous
  - mute
  - unmute
  - set volume
  - set position
  - shuffle

- loop
A description of the commands can be found in the chapter [Multimedia Player](#)[484] .
- Value
A value can be set for the set volume and set position command.

Open For Input

With the open for input action it is possible to open one object on the current page for input so that the user can directly edit it without the need to navigate to it.
Properties:

- Object ID
The ID of the object that shall be made open for input

Open PDF Reader

It is possible to show PDF documents in a separate application. With this action the external PDF reader can be opened.
Consequently the PDF reader has to be closed before one can see the project again. Even alarms will be hidden behind the PDF reader on i.MX 6 based devices.

Properties:

- PDF File To Open
A PDF file can be selected that will be copied to <project directory>/terminal_files/additional_files. The PDF reader on the device will then show this file when opening.
- Show file open button
The PDF reader can show a "File Open Button" that enables the user to open other PDF files from the file system or external media. If this button shall be shown, this property has to enabled. This property only has an effect if the PDF reader is opened with a file. If it is opened without a PDF file, the "File Open Button" is always shown.

Notes about viewing PDFs:
Handling of PDFs may need a lot of CPU power and may consume a lot of memory (especially when loading a new page). To improve the user experience the following points should be taken into consideration:

- All fonts used in the PDF should be embedded in the document so the PDF reader does not have to search for a matching system font.
- If the PDF document contains embedded graphics, they should be saved in the same resolution (DPI) as the display of the target device.
- The whole PDF document should already have the correct size to display at 100% on the target device so that the user does not need to zoom in or out.
- If your system needs very fast response times on the CAN bus, it should be tested if these response times can be kept while displaying a PDF or switching pages in the PDF reader.

Note that the PDF reader does not work in the PClient simulation.

Set Media Player Command

The available actions for the Media Player are described [here.](#)[545]

Toggle Video Display

This action is currently not supported and has no effect at runtime.

Temporarily Display Video

This action is currently not supported and has no effect at runtime.

6.3   General Objects

6.3.1   Page Object

The page object is the root object for designing a page.
Only Frame objects can be added to a page.
A page can also be used as video page. See properties below.
The page has a limited set of properties. For a description of the general properties see
here[415] .

Page Specific Properties

Connect All Softkey Frames

When changing Softkey levels it is possible to switch a level of each Softkey Frame
separately or to connect all Softkey Frames. When the Softkey Frames are connected,
changing the level of one Softkey Frame will also change the level of the other.
Default value: Enabled
Possible values: Enabled or Disabled

For **Video Properties** please see here[508] .

Events

The following events can be configured for the page:
- OnPageInit
- OnPageExit
- OnPageRepeat

See here [430] for the descriptions of all event and possible actions that can be assigned.

6.3.2   Alarms

For a description of the alarm settings in project properties and for how to create an alarm
see here[568] .
Alarms share some properties with other objects. For a description of the common properties
see here[415] .

Alarm Specific Properties

Alarm Number

This property shows the alarm number of the alarm as defined when creating the alarm.
This number can be changed but needs to stay unique.
Default value: Alarm number specified when creating the alarm
Possible values: 1 ... 32767 (but needs to be a unique alarm number)

Alarm Type Name

The type of the alarm is defined when creating it. It can not be changed after the alarm was
created.
Default value: Alarm type chosen when creating the alarm
Possible values: Not editable

Connect All Softkey Frames

When changing Softkey levels it is possible to switch a level of each Softkey Frame separately or to connect all Softkey Frames. When the Softkey Frames are connected, changing the level of one Softkey Frame will also change the level of the other.
Default value: Enabled
Possible values: Enabled or Disabled
Note: This property is only available for Page alarms. Frame alarms do not support softkeys.

Timeout

The timeout of the alarm as configured in project properties[568] for the alarm type.
This property can not be changed after the alarm was created.

Sound Pattern

When an alarm is created, the sound pattern as defined in the project properties is set for the alarm. This sound pattern can be changed for each alarm individually.
For a detailed description of the sound pattern, refer to Alarms[443]

Audio Output

When an alarm is created, the audio output options as defined in the project properties is set for the alarm. This audio output configuration can be changed for each alarm individually.
For a detailed description of the audio output configuration, refer to Events and Actions[432]

Min Display Time

The minimal display time is the time that the alarm that the alarm is at least visible on the screen. Only after this time the alarm can be acknowledged.
The value for the Min Display Time can only be set in the project properties[568] for each alarm type and can not be changed for each alarm.

Log this Alarm

When the alarm log is enabled in project properties[568] , this property can be used to choose if this particular alarm shall be logged or not.
Default value: Enabled
Possible values: Enabled or disabled.
If this property is enabled, it only has an effect if the alarm log is also enabled in project properties.

Alarm Log Delay

With this property it can be configured if this alarm shall be logged directly when the alarm is engaged or only after a certain amount of time. This can be helpful if an alarm shall only be logged if it was engaged for a longer time.
Default value: 0 seconds (log this alarm directly without delay)
Possible values: 0 seconds to 2147483647 seconds
The value of this property only has an effect if *Log this Alarm* is enabled and the alarm log is enabled in project properties[568] .

Alarm Log Text

For this property a text can be entered that is used in the alarm log. This text can not be automatically translated (it is not language dependent).
Default value: no value

Possible value: any string

LED Properties

The LED properties allow to configure the behavior of the multi color LED when entering the selected page.

LED Options

This property defines the behavior of the LED when entering the page.
Default value: LED off
Possible values:
- LED off (LED is off when entering page)
- Always On (LED is always on on that page)
- Blinking (LED is blinking)

Notes:
- The LED behavior can also be controlled via the variables in the LEDs variable group. This provides more customization possibilities.
- The settings made in this property will be overwritten by the values set to the variables in LEDs variable group.

Led Color

Define the color that the multi color LED emits when on or blinking.
Default value: Yellow (255, 255, 0)
Possible values: Any 24Bit RGB color

Blink Pattern

If the LED Options property is set to Blinking, a blink pattern can be configured.
A blink pattern consists of the number of of repetitions and a bit coded pattern of on/off states. In the dialog that pops up when changing this property, the pattern can be defined by clicking on the boxes.
Each box represents 100ms of LED on or off.
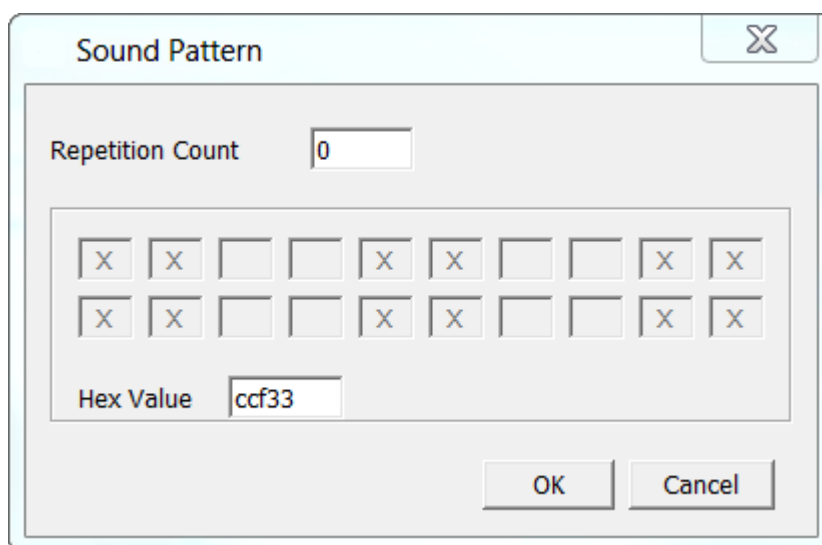
In the screenshot above the blink pattern 0xccf33 was defined. In binary this is
11001100111100110011
- 100ms ON

- 100ms ON
- 100ms OFF
- 100ms OFF
- 100ms ON
- 100ms ON
- 100ms OFF
- 100ms OFF
- 100ms ON
- 100ms ON
- 100ms OFF
- 100ms OFF
- 100ms ON
- 100ms ON
- 100ms OFF
- 100ms OFF
- 100ms ON
- 100ms ON
- 100ms OFF
- 100ms OFF

Internal Variables

If Connected to LED internal variable, this property is enabled and any variable can be selected that then controls the on/off state of the LED.
This property is not supported at runtime on all devices.
It is recommended to use the predefined internal variables in the LEDs variable group to control the LED via variables (e.g. in a PageInit script).

6.3.3   Virtual Keyboards

Virtual keyboards assist the user in entering text or numbers. Two types of virtual keyboards exist:
- Layered Keyboards for text input (portrait or landscape version)
- Numeric Keyboard for number input

To operate a virtual keyboard, the keyboards contain a special version of the button object. See below for special properties of the virtual keyboard buttons.
For virtual keyboard settings in the project properties see [here].
To create a virtual keyboard do a right click on *Virtual Keyboards* in the project tree and select the virtual keyboard type that shall be created.

Virtual Keyboard Layer Specific properties

Virtual keyboard properties are same for layered and numeric keyboards.

Select Layer

Here the layer of the virtual keyboard can be selected. A layer contains a set of keys. On a layered keyboard there are three layers available:
- Layer 1: Lower case characters
- Layer 2: Upper case characters

- Layer 3: Numbers and special characters

A numeric keyboard only has one layer with the numbers.
Default value: Layer 1
Possible values: Any available layer
This property has no effect on runtime, it only shows the different layers in the designer area.

Background Color

The background color of the virtual keyboard.
Default color: black (RGB 0, 0, 0)
Possible values: any 24Bit RGB color

Key Background Image

An image can be selected with this property that is then used as background image for all buttons on the virtual keyboard.
Default value: A default background image
Possible values: Any supported image or no image

Key Background Color

The background color for all buttons in the virtual keyboard can be selected with this property. The background color of the buttons is only visible if a Key Background Image (see above) with transparent areas is used or no background image is set.
Default value: grey (RGB 153, 153, 153)
Possible values: Any 24Bit RGB color

Draw Border

With this property the Draw Border property of all buttons in the virtual keyboard can be enabled or disabled. Enable it if the user shall be able to navigate the buttons via encoder. If only touchscreen input is used, this property can be disabled.
Default value: Enabled
Possible values: Enabled or Disabled

Font Attribute

The font attribute for all buttons in the virtual keyboard can be selected with this property.
Default value: Default_DejaVu Serif
Possible values: Any font attribute defined in project properties[76]

Font

The font that is used for all buttons in the virtual keyboard can be selected with this property.
Default value: DejaVu Serif
Possible values: Any available font

VKBButton Specific properties

Virtual keyboard buttons are basically normal button objects[463] that have some additional properties.
The virtual keyboard buttons are language dependent by default.

Button Type

A virtual keyboard button can have one of the following types:
- Character

Button for normal character input. When a Character button is pressed, the button's value is added to the edited value.
- Layer Switch
When pressing this button, the layer of the keyboard will be switched. The target layer can be selected with the *Switch To Layer* property (see below).
- OK
The OK button confirms the edited value and closes the virtual keyboard
- Cancel
The cancel button closes the virtual keyboard without confirming the new value (it will be reverted to the value before opening the virtual keyboard)
- Delete
The delete key deletes the last character or number that was entered
- Toggle Sign
The Toggle Sign button should only be used on numeric keyboards. It can be used to toggle the edited value from positive to negative and vise versa.

Display Text

This language dependent property defines the character displayed on the button. It can be different for each language.

Character Values

This language dependent property defines the value that is added to the edited value when the button is pressed. It can be different for each language.

Switch To Layer

This property is only available if *Button Type* (see above) was set to *Layer Switch*. Possible values: Any available layer of the virtual keyboard.

6.3.4   Softkey Frame

Softkey frames are (at default) invisible containers for softkey levels. Softkey levels contain the softkeys[449] .
A Softkey frame can have up to 4 softkey levels. Softkey levels can be used if there is not enough space on the screen to show all softkeys. At runtime it is possible to switch to another softkey level to show different softkeys.

Create Softkey Frames and Levels

At default two softkey frames frames are created for each page automatically each containing one softkey level.
To create a new softkey frame open the page from project tree and do a right click on the DataMask object of the page in the project tree and select Create Softkey Frame XXX (XXX is either Left or Right for landscape devices or Bottom or Top for portrait devices). If all softkey frames for the page already exist, this option will not be available on right click. After softkey frame contains one default softkey level. To create more levels do a right click on the softkey frame in the project tree and select Create Softkey Level. The new created level will then directly be shown.
To switch between the levels in the designer, the level that shall be shown can also be selected via right click on the softkey frame in the project tree.
To switch between softkey levels at runtime, the actions Next Softkey Level or Previous Softkey Level can be attached to a softkey.

Softkey Frame Specific Properties

For a list of other common properties of a softkey frame that are shared with other objects see here □ ⁴¹⁵ .

Softkey Width

As default softkeys have a calculated size so that no gaps exist between the softkeys. With this properties it is possible to make the softkey width smaller in a bottom or top softkey frame so that gaps between the softkeys are possible.
Default value: Calculated based on device
Possible values: Anything smaller than the calculated value
This property can only be changed for top or bottom softkey frames. For left or right softkey frames the width of the softkeys is directly connected to the width of the softkey frame.

Softkey Height

As default softkeys have a calculated size so that no gaps exist between the softkeys. With this properties it is possible to make the softkey height smaller in a left or right softkey frame so that gaps between the softkeys are possible.
Default value: Calculated based on device
Possible values: Anything smaller than the calculated value
This property can only be changed for left or right softkey frames. For top or bottom softkey frames the height of the softkeys is directly connected to the height of the softkey frame.

Current Softkey Level

The *Current Softkey Level* property is not visible in the designer tool and can only be accessed at runtime via JavaScript.
With this property the currently shown softkey level can be changed.
Note: it is also possible to change the softkey levels via the variable *@ChangeSoftkeyLevel* (see variable table for description).

Events

The following events can be configured for the softkey frame:
- OnPress
- OnRelease

Note: These events are only triggered if there are no input objects on the page. Otherwise navigation scheme uses the touch events for input objects only and the softkey frame is not considered an input object.
See here □ ⁴³⁰ for the descriptions of all event and possible actions that can be assigned.

6.3.5   Softkey

Softkeys are an easy and convenient way to add functionality to a hardware key but also can be used on devices without keys (via touchscreen).
Softkeys are very similar to buttons □ ⁴⁶³ (they only lack the Draw Shadow property.
Softkeys also share a lot of their properties with other objects. See here □ ⁴¹⁵ for a description of the common properties.

How to add pressed or released objects to the softkey

It is the same as for the button object. Click here □ ⁴⁶³ for a description.

Softkey Specific Properties

Key Number

This is a property that can not be changed. Each hardware key gets a specific number. The value of this property is based on the softkey frame in which the softkey is in and the position inside the softkey frame. This number is used in the variable @UserButtonPressed which indicates the last key that has been pressed and, accordingly, @UserButtonReleased.

Latching

Latching means: When a softkey is pressed and released, it stays in the pressed state until it is pressed again.
With this property this behavior can be enabled.
Default value: Disabled
Possible values: Enabled or Disabled

Pressed State

Switch between the release or pressed state of the softkey. This is important for having a preview of the object and also for adding pressed/released objects for the softkey (see here [463]).
This property has no influence to runtime behavior.
Default value: Disabled
Possible values:
- Enabled (softkey is in pressed state)
- Disabled (softkey is in released state)

Pressed Object ID

The ID of the object that represents the pressed object of the softkey.
Will be automatically set when adding a pressed state object to the softkey (see here [463]).
Can't be edited by hand.

Released Object ID

The ID of the object that represents the released object of the softkey.
Will be automatically set when adding a released state object to the softkey (see here [463]).
Can't be edited by hand.

Events

The following events can be configured for the softkey:
- OnPress
- OnRelease
- OnLongPress
- OnLongRelease
- OnDblClick

See here [430] for the descriptions of all event and possible actions that can be assigned.

6.4 Palette Objects

6.4.1 2D Graph

New and changed features since May 2016

- All data points are logged to the non-volatile memory of the device so that they are still available after reboot.
- Maximum value of Number of Samples to Record[456] reduced from 10,000,000 to 1,000,000
- Maximum number of value points visible for each variable when showing the 2D Graph set to 699050 (see graph description below for details)
- Introduced a maximum "Number of Samples to Record" for all graphs in a project (number is device dependent)
- Each variable can be logged either time based or at every value change (see Log Trigger[457])
- Possible to configure if data shall be logged in low power mode or not with new property *Log Data in Low-Power Mode*[454]
- Introduced new variable reference for showing the progress of copying the graph's history. See Copy Progress Variable[453] and Process 2D Graph Data on a PC[458]
- Introduced two new JavaScript functions to save (copy) or delete the graph's value history (save2DGraphHistory[547] , delete2DGraphHistory[548])

The graph object is designed to show the value history of one or more variables in a line graph.

For each variable that shall be displayed in the graph it can be configured if logging of a value shall happen for every value change or time based (see Log Trigger[457]).

Each logged data point consists of a time stamp and the value at that time stamp. All entries are saved to the non-volatile memory of the device in a ring buffer so that the value history is still available after shutting down and re-booting the device. Logging is active even if the 2D graph object is not visible on the screen.

WARNING: If power is completely removed from the device without properly shutting it down using the ignition, it is possible that the last 1% of data in the ring buffer gets lost! Always power down the device using the ignition and the integrated power management!

To save RAM on the device only the data points that are currently shown on the screen are loaded from the hard drive. This means that one can experience slight delays when changing the currently visible time span or the offset because the data needs to be loaded from disk.

To make sure that the non-volatile memory of the device does not get filled up with graph history data, the number of data points that are logged for all graphs in a project is limited (device dependent: on devices with i.MX6 CPU: 34952533 samples (about 400MB on the non-volatile memory), on devices with i.MX35 CPU: 8947848 (about 100MB on the non-volatile memory)).

Drawing the graph's value lines can be very CPU heavy. The value lines are a polygon that consists of many points. To improve drawing performance, this polygon will be simplyfied if it consists of too many points (current threshold: ~1000 samples). While this improves the drawing performance it may reduce the accuracy of the drawn polygon but may keep the display still usable if too many samples shall be drawn.

The logged history of each graph can be **copied** from the device and then be converted to a CSV file that in turn can be processed by a spreadsheet application. See here[458] for instructions.

The complete value history can be **reset (deleted)** with the JavaScript delete2DGraphHistory[548] function.

Displaying the logged data at run time needs some CPU time. To improve performance of the 2D Graph, the object is not redrawn at each variable change but automatically redraws itself with the latest values in a fixed interval (currently about every second). If there are still performance issues the following measures can improve it:

- Reduce the numbers of shown variables in the graph
- Reduce the number of samples for each variable in the graph (see Number of Samples to Record[456])
- Scale the X axis so that less data points are visible at once
- Consider changing the Log Trigger[457] for each variable to *Time Based* with an appropriate timer value.
- If all of the above is not feasable for you application, think about upgrading to a newer device generation with more CPU power

Variable References

Each variable can either be attached via the dialog that pops up when the ... button is pressed in the property table for property "Variables To Draw" or by dragging a variable from the Variable View onto the graph in the designer area.

X Axis Time Variable

With a variable attached to this property the time span of the X axis can be scaled. If no variable is attached, the default time span is 10 seconds.

At run time the scaling is done every time the variable value is changed. The value of the variable represents the time span of the X axis in seconds.

Variable values less than 1 are not considered.

Default value: None

Possible values: Any integer variable

X Axis Offset Variable

When a variable is attached to this property, it is possible to move in the history of the graph. For each increment of the variable, the graph will move one second into it's history and shows "old" values. Values less than 0 are not considered. If the value 0 is set, the graph shows "live updates" of the values.

Default value: None

Possible values: Any integer variable

Cursor Position Variable

It is possible to move a cursor line in the graph. With this cursor line it is possible show the values of each variable at the time where the cursor is. The cursor position represents the distance in pixel from the right side of the 2D Graph.

When a variable is attached to this property the cursor can be activated by setting a value greater or equal to 0 (0 shows the latest values).
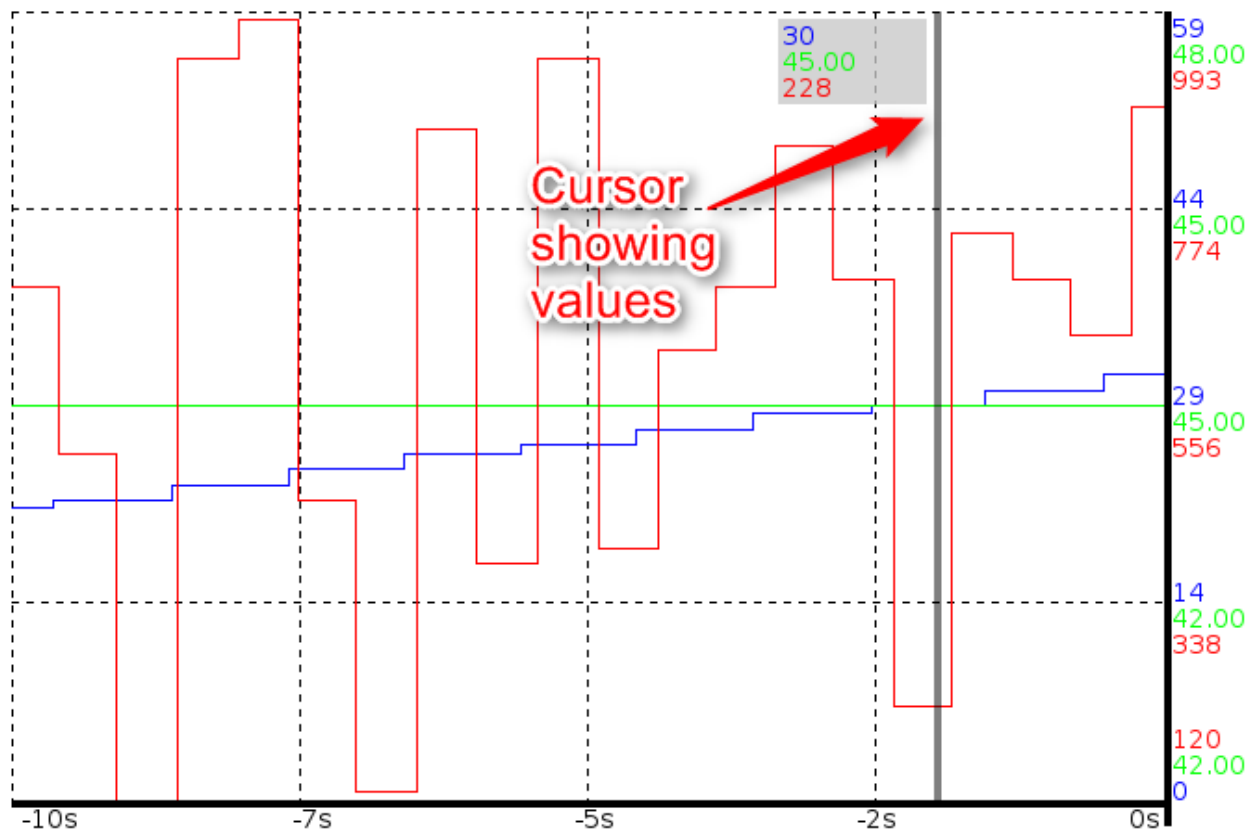
Default value: None
Possible values: Any integer variable

Copy Progress Variable

If an integer variable is atached, it will show the copy progress when copying the graph's history data. See [save2DGraphHistory](#)[547] and [here](#)[458] for details.
Default value: As configured for the varaible
Possible values: 0 .. 100 while copy is in progress or -1 on error.

2D Graph Specific Properties

Draw Grid Lines

Enable this property to either show one or three vertical and horizontal grid lines (see *Number of Grid Lines* property below). The look of the grid lines can be configured with the *Grid Line Attribute* property.
Disable this property to hide all grid lines.
Default value: Enabled
Possible values: Enabled or Disabled.

Grid Line Attribute (Theme dependent)

Select the line attribute object that shall be used for drawing the grid lines.
Default value: Solid Black
Possible values: Any Line Attribute Object defined in project properties.

Number of Grid Lines

If the property *Draw Grid Lines* is enabled, grid lines will be drawn in the graph. With this property the number of drawn grid lines can be configured.

Default value: 3
Possible values:
- 3 (three horizontal and three vertical grid lines will be drawn)
- 1 (one horizontal and one vertical grid line will be drawn)

Number of Grid Lines

With this property the existence and position of the Y axis can be configured. It either can be on the left or on the right or completely disabled.
The main purpose of the Y axis is to show the (current) minimum and maximum values of the variables shown in the graph (configure this for each variable with the property *Show Value Labels at Y Axis* in the *Variables To Draw* dialog, see below).
Default value: Right
Possible values:
- Right
- Left
- Don't draw at all

Axis Line Attribute (Theme dependent)

Select the line attribute object that shall be used for drawing the X and Y axis lines.
Default value: Solid Black
Possible values: Any Line Attribute Object defined in project properties.

X Axis Labels

The X axis can show either full time stamps (format hh:mm:ss, eg. 09:42:37) or relative timestamps (e.g. -15.0s, 0s, etc.). The X axis labels can also be completely disabled.
Default value: Full Time Stamps
Possible values:
- Full Time Stamps
- Relative Time Stamps
- None

Axis Font Attribute (Theme dependent)

The look of the labels of the X and Y axis can be configured with a font attribute. The color of the font attribute only has influence on the X axis labels. The values at Y axis will be shown in the color of the line attribute object of the configured variable (see *Line Attribute* property for each variable in the *Variables To Draw* dialog).
Default value: Default_DejaVu Serif
Possible values: Any existing font attribute object

Log Data in Low-Power Mode

Configure if data shall be logged if ignition is off and the device is in low power mode.
Default value: Enabled
Possible values Enabled or Disabled

Variables To Draw

The Variables To Draw are the variables that are recorded and shown by the graph. Variables can either be added by dragging them from the Variable View onto the graph in the designer area or by clicking the ... button on this property. When clicking the ... button the following dialog will pop up in which various settings for each variable can be made:
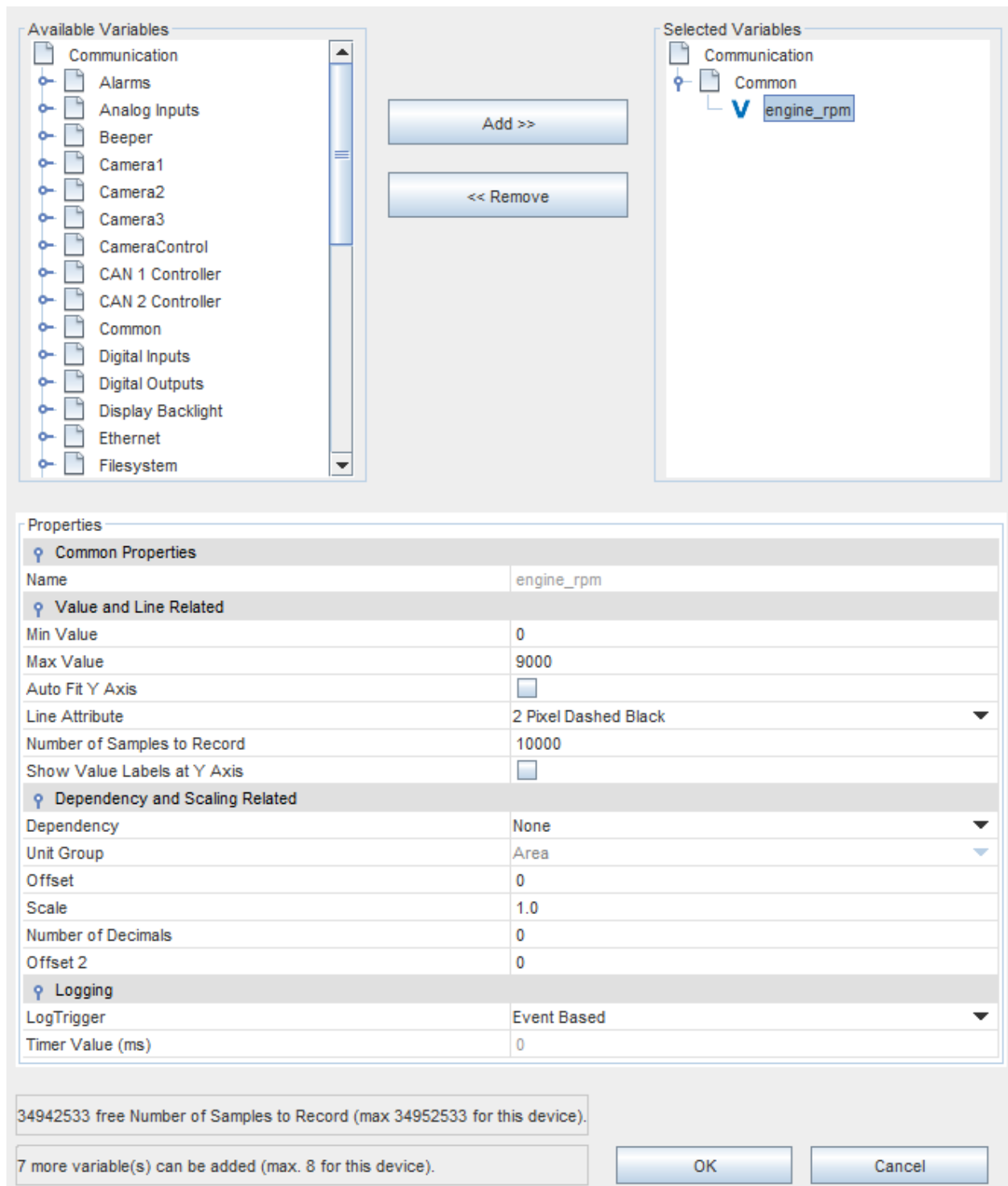
The top part of the dialog is designated for selecting the variables that the graph object shall draw.

On the left the tree with all variables available in the current project is displayed. On the right there is the tree showing the variables that are already assigned to the graph.

To add a variable to the graph, select one in the left tree and click the *Add >>* button.

To remove a variable from the graph, select one in the right graph and click the *<< Remove* button.

It is also possible to add or remove complete groups.

Note: There is a limit of variables that can be added to the graph. This limit is device specific. On the bottom of the dialog the limit is shown and how much variables can be added.
Note: Adding many variables might make the graph confusing.
After a variable was added, the following options can be set for each variable:

Name

The Name of the variable. This can't be edited here.

Min Value

The minimum value for the variable that shall be used for the graph. This defines the lowest value of the Y axis and together with Max Value the range of the Y axis.
If Show Value Labels at Y Axis is enabled, the value of Min Value will be shown at the bottom of the Y axis.
Min Value will not be considered if Auto Fit Y Axis is enabled.
Default value: Configured minimum value of the selected variable
Possible values: Any integer value that is less than Max Value

MaxValue

The maximum value for the variable that shall be used for the graph. This defines the highest value of the Y axis and together with Min Value the range of the Y axis.
If Show Value Labels at Y Axis is enabled, the value of Max Value will be shown at the top of the Y axis.
Max Value will not be considered if Auto Fit Y Axis is enabled.
Default value: Configured maximum value of the selected variable
Possible values: Any integer value that is greater than Min Value

Auto Fit Y Axis

If this property is disabled, the value range of the Y axis for this variable is fixed from Min Value to Max Value.
If this property is enabled, the graph will automatically calculate the range of the Y axis based on all values of the variable currently visible in the graph. Min and Max Value will not be considered.
Default value: Disabled
Possible values: Disabled or Enabled

Line Attribute

To change the look of the value line that is drawn for this variable, a line attribute object can be selected here. Line width, color and line pattern can be defined.
Each variable should get a different line attribute so that they can be distinguished in the graph.
Default value: Solid Black
Possible values: Any line attribute object defined in the project properties.

Number of Samples to Record

To improve performance, the number of value changes that are recorded by the graph can be limited. This should be done especially if no long history is required (e.g. only the latest 100 values are of interest).
As soon as the number of samples are reached during run time, the oldest values recorded will be deleted.
Default value: 10000

Possible values: 10 ... 1000000

Show Value Labels at Y Axis

If this property is disabled, no values for this variable will be shown at Y axis.
If this property is enabled, the Min and Max value are displayed at the Y axis. If grid lines are drawn in the graph, values between Min and Max value will be shown at the grid lines positions at Y axis.
Default value: Disabled
Possible values: Disabled or Enabled

Dependency

It is possible to mark a variable as unit dependent so that the values shown (Min Value, Max Value etc. at Y axis and the value in the cursor) can be calculated for the currently active unit.
Default value: None
Possible values: None or Unit

Unit Group

If the Dependency property is set to Unit, the unit group can be chosen to which this variable belongs.
Default value: Force units (only enabled if Dependency property is set to Unit)
Possible values: Any of the available unit groups.

Offset

The raw value of the variable can be modified before showing it. The formula that is used is:
shownValue = ((rawValue + Offset) * Scale) + Offset2
Default value: 0
Possible values: Any integer number

Scale

The raw value of the variable can be modified before showing it. The formula that is used is:
shownValue = ((rawValue + Offset) * Scale) + Offset2
Default value: 1.0
Possible values: Any float number

Number of Decimals

The number of decimals property allows to define how many digits will be displayed behind the decimal point.
Default value: 0
Possible values: 0 ... 10

Offset / Offset 2

The raw value of the variable can be modified before showing it. The formula that is used is:
shownValue = ((rawValue + Offset) * Scale) + Offset2
Default value: 0
Possible values: Any integer number

Log Trigger

With this property the trigger for logging a value can be configured. It can either be time based (every value change is logged) or time based (a value sample is recorded cyclacly. If *Time Based* is selected, the *Timer Value* can be used to configure the log period for the selected variable.
Default value: Event Based
Possible values: *Event Based* or *Time Based*

Timer Value

Configure the log period in milliseconds for the selected variable if *Log Trigger* is set to *Time Based*
Default value: 500ms
Possible values: 500ms ... 2147483647ms

Process 2D Graph Data on a PC

The recorded log data can be copied to a PC and then be processed by a tool to make it readable by e.g. a spreadsheet application.
Use the JavaScript function save2DGraphHistory[547] to store the data on a USB drive.
Example JavaScript:

```
//Get path of mounted USB drive:
var usbMountPath = getVariableValue("@USBMountPath");
//Save history to target path. The 2D Graph has object ID 42
var result = save2DGraphHistory(42, usbMountPath);
if (true == result)
{
    print("Copy 2D Graph history started");
}
else
{
    print("Starting copy 2D Graph history FAILED");
}
```

Note that the copy progress can be observed via the Copy Progress Variable[453] .
When saving the history, a new directory is created on the USB drive with the following name: graph_hsitory_*NAME_OF_2D_GRAPH*

On the PC

After transfer is done, the data can be viewed and processed with the Graph History Tool. Follow these basic steps to export the data into a CSV file that can be opened by a spreadsheet application:
- Press the browse button and select the graph history directory on the USB drive.
- After the directory was selected, the data of each variable can be viewed in a table.
- Press the Export button to select export options and then press OK to export the data into one or more CSV files.

A detailed manual for the Graph History Tool can be viewed here[459] .

Manual data processing

The data can also be processed by you own applications. The storage format is:
- 64Bit timestamp in milliseconds since 01-01-1970
- 32Bit integer value

All values in Little Endian format.

The ring buffer is implemented as 100 files per variable. Each file is named with the name of the variable and a sequence number.

6.4.1.1 Graph History Tool

The Graph History Tool is a PC tool for viewing and converting saved history data of a 2D Graph object.
This manual is split in two parts: The end user manual that explains how the tool can be used and the OEM documentation that explains how the tool can be deployed to the end user and how it can be customized.

End user manual
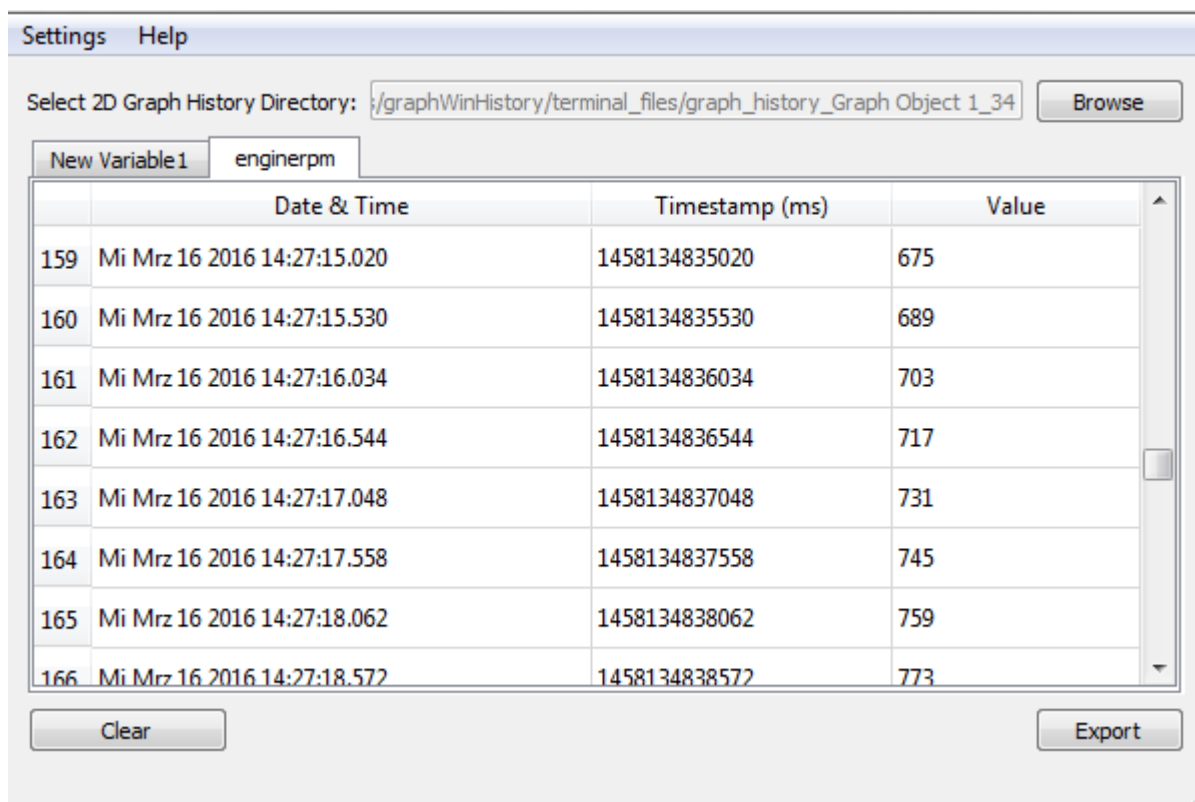
Import and show Graph History Data

Press the *Browse* button to select the directory that contains the graph history data. After the directory has been selected the data is parsed. For each variable that is configured for the 2D Graph, a tab with a table will be shown in the tool. Each table consists of 3 columns:
- Human readable date and time
- Timestamp as a 64Bit unsigned value (milliseconds from 1970-01-01 00:00)
- The value at that timestamp

To clear imported data simply press the *Clear* button.

Export Data to CSV

After data has been imported, press the *Export* button to show the export settings dialog.

The first step is to select what shall be exported:

- **Export Selected Tab Only**
  If this option is selected, only the selected variable (active tab) will be exported into one csv file
- **Export All Tabs in Single Files**
  If this option is selected, one csv file for each variable will be exported.
- **Export All Tabs in One Combined file** If this option is selected, one csv file will be exported that will contain all variable values with a column for each variable.

If a different separation character shall be used, this can be configured with **Separator character**. This can be handy because some language versions of spreadsheet applications treat the separation character differently (e.g. English Excel wants a *,* whereas German Excel wants a *;*).

If either Date and Time or the 64Bit timestamp is not needed in the CSV file, the checkmarks of **Export Date and Time** or **Export Timestamp** can be removed.

The last step before the export can begin is to select an output file (if **Export Selected Tab Only** or **Export All Tabs in One Combined file** is selected) or an output directory (if **Export All Tabs in Single Files**) is selected

When all export settings are correct, hit the **OK** button to start the export.

OEM documentation

Deploying the tool to the end user

The Graph History Tool is a Qt5 application and therefore has to be shipped with the matching Qt and runtime libraries. It should run on every 32Bit or 64Bit Windows version (starting with Windows 7).

The application and all the neccessary libraries can be found in the roaming files installation path of the designer application. The sub-directory is *\dev\extracted_components\pclient\simulation\graph_history_tool*. As long as the

graphHistoryTool.exe is inside this directory, the start.bat batch file can be used to start it.
To deploy the application for the end user, copy the following libraries from the shared_libraries directory into the graph_history_tool directory:

- Qt5Core.dll
- Qt5Gui.dll
- Qt5Widgets.dll
- libgcc_s_dw2-1.dll
- libstdc++-6.dll

After that the graph_history_tool directory can be zipped and sent to customers. The start.bat is then not needed anymore and the graphHistoryTool.exe can be started directly.

Adding new language translations

The Graph History Tool comes with two language translations (English and German). If other languages are needed, there is a way to add them:
The language translation is done with the Qt Linguist application (see here). If you want to create a translation on your own, you have to install Qt on your PC (download here, make sure to install the Linguist application).
After installation open the Linguist application and open **one** of the existing translation files (the files with *.ts* suffix, e.g. graphhistorytool_English_en.ts). Now go to File -> Save As and choose a new name for the translation file (to show the language in the Settings menu of the Graph History Tool the language file needs to be called *whatever_<NATIVE_LANGUGAGE_NAME>_<LANGUAGE_ACRONYM>.ts*, e.g. for spanish: *graphhistorytool_Spanish_es.ts*). After that go to Edit -> Settings. In this dialog the target language can be chosen. Chose the language and Country/Region then press OK and save the file.
Now you can open all translation files and start the translation process for the new language.
When the translation is finished, a binary translation file has to be created from the ts file. Sadly Qt Linguist does not provide a GUI tool for this operation so you have to do this on the command line: Open a command line and navigate to the directory where the .ts files are stored. Now type *lrelease *.ts*. This will create the binary translation files (suffix .qm). Place all .qm files in the application directory. When you open the Graph History Tool now, the new application should be available under Settings -> Language.
If lrelease.exe is not in your search path, you have to call it with the complete path, e.g. *C:\workspace\Qt5.4\5.4\mingw491_32\bin\lrelease.exe *.ts*

6.4.2   Arched Bargraph

The arched bargraph basically works like a linear bargraph[475] with the exception that it can be bent and the start and end angles can be adjusted.
The arched bargraph shares many properties with other objects. Find a description of them here[415].

Angles

The angles of the arched bargraph can be changed with the Start Angle and End Angle properties (see below). Some screenshots with examples follow here:

- Start Angle 0, End Angle 360:
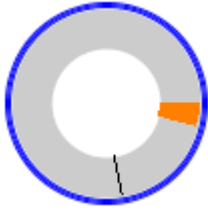
- Start Angle 0, End Angle 315:

- Start Angle 45, End Angle 0:

- Start Angle 135, End Angle 225:

- Start Angle 315, End Angle 225:

Arched Bargraph Specific Properties

Start Angle

The start angle of the arched bargraph can be defined with this property.
The start angle's 0° position is at 3 o clock.
Default value: 0°
Possible values: 0° ... 360°
See examples above

End Angle

The end angle of the arched bargraph can be defined with this property.
The start angle's 0° position is at 3 o clock.
Default value: 0°
Possible values: 0° ... 360°

See examples above

Draw Target Line

When enabling this property, a line will be drawn at the position of the [Target Value](#)[427] property value.
Default value: Enabled
Possible values Enabled or disabled.

Target Line Color

The color in which the target line (see above) will be drawn.
Default value:black (RGB 0, 0, 0)
Possible values: Any 24Bit RGB color

Deflection Direction

The deflection direction defines how the value grows in an arched bargraph. Either clockwise or anti clockwise.
Default value: ClockWise
Possible values: ClockWise or AntiClockWise

Bar Graph Width

Since the Width property only defines the width of the bounding box of the arched bargraph it is possible to define the width (or thickness) of the arched bargraph with the Bar Graph Width property.
Default value: 20 pixel
Possible values: 1 ... (height of the object divided by 2 or width of the object divided by 2, whatever is smaller)

Fill Mode

If Fill Mode is enabled, the value will be shown as a filled area in the bargraph. If Fill Mode is disabled only a line will be drawn at the value's position.
Default value: Enabled
Possible values: Enabled or disabled.

Events

The following events can be configured for the arched bargraph object:
- OnOpenForInput
- OnEnabledWithFocus
- OnEnabledWithoutFocus
- OnValueChange
- OnEnterValue
- OnEnterNewValue
- OnEscape

See [here](#)[430] for the descriptions of all event and possible actions that can be assigned.

6.4.3 Button

The button object can display two states: Pressed or released.
To change the look & feel of the button, it can be configured with its properties.
For common object properties refer to [this page](#)[415] . Here only the button specific properties are described.

Example project: [Download this project](#) to see the different button events in action.

How to add pressed or released objects to the button

Two other objects can be added to a button. One indicating the released state of the button and one the pressed state.
The button supports the following child object types:

- Numeric Field
- String Field
- Arched Bargraph
- Linear Bargraph
- Meter
- List
- Picture Graphic
- Container

It is common to use String Fields or Picture Graphics for pressed and released objects. Containers can be used to design more complex buttons while keeping the "one object for pressed and one object for released state" constraint.
Note: If only one child is assigned it will be shown permanently on device whereas toggling "*Pressed State*" on PC tool will show one state empty and the other with the child.
Attention: While it is technically possible, do not put a container inside of a button and then another button inside the container. Such combinations are not supported or tested and will most likely lead to unexpected behavior.

How to add the objects to a button:

- Drop a button on a frame
- Make sure that the "*Pressed State*" property of the button is disabled
- Drag any supported object from the palette to the button in the designer area (or an existing object from project tree to the button in project tree)
  - o The added object will now represent the release state of the button
- Enable the property "*Pressed State*"
- Drag any supported object from the palette to the button in the designer area (or an existing object from project tree to the button in project tree)
  - o The added object will now represent the press state of the button

To assign another object to a button, follow the same steps above. A dialog will ask if the pressed/released object shall be replaced.

Pressed / Released Image

Alternatively to putting one or two Picture Graphic objects into the button, these properties can be used to add 1 or 2 images to the button without adding more objects.

Button Specific Properties

Draw Shadow (Theme dependent)

A button shadow can be drawn around the button that changes according to pressed or released state of the button. This is a visual help for the user to determine if the button is just pressed or not.

Disable this property if the design of the button is completely custom and the press/release state is distinguishable using the pressed and released objects (see above).
If this property is enabled, a small region between button background (or object inside button) and border of the button is reserved for the shadow. If it is disabled, the background of the button (or object inside the button) goes directly to the border.
Default value: Enabled
Possible value: Enabled or Disabled

Shadow Color (Theme dependent)

If "*Draw Shadow*" is enabled one can define the shadow color used to draw the shadow.
The background color of the shadow is the button background color which includes that. the background of the shadow is transparent if the button is configured as transparent.
The button shadow consists of two areas where one is drawn in the opaque color selected and the other is drawn in a half-transparent version of the same color.
Both colors are switched when button is pressed to simulate a kind of 3D effect of the pressed/released button.
Default value: Black.
Possible value: Any color allowed by standard color chooser.

Latching

Latching means: When a button is pressed and released, it toggles state and stays in the toggled state until it is pressed again.
Like that a latching button resembles a light switch whereas a non-latching button resembles a door-bell switch.
With this property this behavior can be enabled.
Default value: Disabled
Possible values: Enabled or Disabled

Please note that this property affects the behavior of the press and release events of the objects.

Pressed State

Switch between the release or pressed state of the button. This is important for having a preview of the object and also for adding pressed/released objects for the button (see above).
Note: If only one child is assigned to the button the other state will be shown as empty on PC tool but on the device this one child will show permanently - button shadow should be used to show button state in this scenario.
This property has no influence to runtime behavior.
Default value: Disabled
Possible values:
- Enabled (button is in pressed state)
- Disabled (button is in released state)

Pressed Object ID

The ID of the object that represents the pressed object of the button.
Will be automatically set when adding a pressed state object to the button (see above).
Can't be edited by hand.

Released Object ID

The ID of the object that represents the released object of the button.

Will be automatically set when adding a released state object to the button (see above). Can't be edited by hand.

Pressed Image (Theme dependent)

Alternatively to putting one or two Picture Graphic objects into the button, this property can be used to add an image to the button. The image set in this property will be used as the pressed image. If no image is set for the Released Image property but for the Pressed Image property, the button will only show an image when it is pressed. Both properties should be set with an image.

Released Image (Theme dependent)

Alternatively to putting one or two Picture Graphic objects into the button, this property can be used to add an image to the button. The image set in this property will be used as the released image. If no image is set for the Pressed Image property but for the Released Image property, the button will only show an image when it is released. Both properties should be set with an image.

Events

The following events can be configured for the button object:
- OnPress
- OnRelease
- OnLongPress
- OnLongRelease
- OnDblClick
- OnOpenForInput
- OnEnabledWithFocus
- OnEnabledWithoutFocus
- OnValueChange
- OnCancel

See [here](#) [430] for the descriptions of all event and possible actions that can be assigned.

6.4.4   Container

 Tutorial video about the container object
https://www.youtube.com/watch?v=6ttjYv3flRc

The container object is only a "container" for other objects. It is not visible at runtime.
A container can be used to group other objects and add a deeper hierarchy in the project tree.
The [Visibility property](#) [588] can be used to make all objects inside the container visible or invisible with just one command and no need to assign visibility properties to each object in the container. Also, moving a set of objects is easier when they are in a container.

To put an object into a container, drag and drop the object into the container in the project tree.

When putting existing objects into a container, please note that the position of the object relative to the parent object will be used to position the object in the container.

I.e. if the object is in a frame that starts at position 0,0 then the container should also start at that position, otherwise the object will be moved to position + container position.

To put an existing layout of objects into a container and keep all positions, put the top left edge of the container at the top left edge of the page and re-size the container so that the objects you want to put in are within the container boundaries.

Attention!

Please do not try to override limitations of the objects by using containers, i.e. by putting a container inside of a button and then a button inside of the container in the button. Such combinations are not supported or tested and will most likely lead to unexpected behavior. Also, don't put containers inside of other containers.

6.4.5   Frame

The frame is the basic object that should be at least once on a page (maybe except a video page).

A frame needs to be placed on a page so that objects can be added to it. When creating a new page, a full screen frame will automatically be created.

Frames can be used to display a video from the device's analog video input (see property description below).

As many other objects, the frame has a lot of common properties. See [here](#) [415] for a description. Only frame specific properties will be described on this page.

Frame Specific Properties

Frame Type

This property has no effect and will be changed or removed in a future release.

For **Video Properties** please see [here](#) [508] .

Events

The following events can be configured for the frame object:
- OnPress
- OnRelease

See [here](#) [430] for the descriptions of all event and possible actions that can be assigned.

6.4.6   Gauges

The gauge object can be used to design gauges or other instruments requiring a needle (indicator).

The gauge shares many properties with other objects. Find a description of them [here](#) [415] .

Angle

The angle of a gauge can be changed with the Angle property.
For the full Gauge, Start and End Angle can be modified to change the range of the instrument as needed.
For the other Gauges, only the Start Angle can be modified. The End Angle will be calculated depending on the range of the gauge (270, 180 or 90 °).

Gauge Specific Properties

Start Angle

The start angle of the meter can be defined with this property.
The start angle's 0° position is at 3 o clock.
Default value: 0°
Possible values: 0° ... 360°
See examples above

End Angle

The end angle of the meter can be defined with this property.
The start angle's 0° position is at 3 o clock.
Default value: 0°
Possible values: 0° ... 360°
See examples above

Deflection Direction

The deflection direction defines how the value grows in the gauge. Either clockwise or anti clockwise.
Default value: ClockWise
Possible values: ClockWise or AntiClockWise

Needle Color (Theme dependent)

With this property the color of needle can be defined.
Note: This property has no effect if a *Needle Image* is used.
Note: See also common properties *Color Above Maximum* and *Color Below Minimum* which also have an influence on the needle color.
Default value: White (RGB 255,255, 255, 255)
Possible values: Any 24Bit ARGB color

Needle Image (Theme dependent)

Instead of the automatically drawn needle, a needle image can be assigned to the gauge to show a different kind of indicator.
The needle image should have the same size as the gauge and the needle in the image should always point to the 3 o'clock position.
Note that the image will be rotated around its center, so place the needle accordingly.
Example needle image:

Default value: No image
Possible values: Any image with supported format (bmp, jpeg, png, svg)
Note: To make the needle image look good when rotated, it is recommended to use scalable vector images (SVG). This prevents steps or anti aliasing artifacts.

Needle Thickness Factor

With this property the thickness of the needle can be modified. The reference value (which equals a factor of 1) is the radius of the gauge object.

Default value: 0.04
Possible values: 0.0 ... 1.0

Draw Arc (Theme dependent)

The arc is a small colored line around the gauge (see *Arc Color* below).
With this property the drawing of the arc can be enabled or disabled.
Default value: Disabled
Possible values: Enabled or disabled

Arc Color (Theme dependent)

The color of the arc of the gauge (see above).
Default value: Cyan (RGB 255,0, 255, 255)
Possible values: Any 24Bit ARGB color

Arc Width Factor (Theme dependent)

The width of the arc can be modified. The reference value (which equals a factor of 1) is the radius of the gauge object.
Default value: 0.03
Possible values: 0.0 ... 1.0

Draw Ticks (Theme dependent)

Ticks (small lines as value indicators) can be drawn around the gauge. With this property the drawing of the ticks can be enabled or disabled.
Default value: Enabled
Possible values: Enabled or disabled

Draw Minor Ticks (Theme dependent)

Ticks (small lines as value indicators) can be drawn between the "major" ticks around the gauge. With this property the drawing of the minor ticks can be enabled or disabled.
Default value: Enabled

Possible values: Enabled or disabled

Tick Color (Theme dependent)

The color in which the gauge's ticks are drawn.
Default value: Light grey (255,204,204,204)
Possible values: Any 32Bit ARGB color

Minor Tick Color (Theme dependent)

The color in which the gauge's minor ticks are drawn.
Default value: Light grey (255,204,204,204)
Possible values: Any 32Bit ARGB color

Number Of Ticks (Theme dependent)

The ticks around the gauge are evenly spread. The number of ticks can be defined with this property.
Default value (Gauge): 12
Default value (Gauge 270): 9
Default value (Gauge 180): 7
Default value (Gauge 90): 3
Possible value: 0 ... 360

Number Of Minor Ticks (Theme dependent)

The minor ticks between two "major" ticks are evenly spread. The number of minor ticks between to "major" ticks can be defined with this property.
Default value: 3
Possible value: 0 ... 360

Tick Height Factor (Theme dependent)

The height of the ticks can be modified. The reference value (which equals a factor of 1) is the radius of the gauge object.
Default value: 0.2
Possible values: 0.0 ... 1.0

Tick Width Factor (Theme dependent)

The width of the ticks can be modified. The reference value (which equals a factor of 1) is the radius of the gauge object.
Default value: 0.02
Possible values: 0.0 ... 1.0

Minor Tick Height Factor (Theme dependent)

The height of the minor ticks can be modified. The reference value (which equals a factor of 1) is the radius of the gauge object.
Default value: 0.08
Possible values: 0.0 ... 1.0

Minor Tick Width Factor (Theme dependent)

The width of the minor ticks can be modified. The reference value (which equals a factor of 1) is the radius of the gauge object.
Default value: 0.01
Possible values: 0.0 ... 1.0

Draw Tick Point (Theme dependent)

With this property the drawing of the tick points can be enabled or disabled.
Default value: Enabled
Possible values: Enabled or disabled

Tick Point Size Factor (Theme dependent)

With this property the size of the tick points can be modified. The reference value (which equals a factor of 1) is the radius of the gauge object.
Default value: 0.06
Possible values: 0.0 ... 1.0

Tick Point Color (Theme dependent)

With this property the color in which the gauge's tick points are drawn.
Default value: Light grey (255,204,204,204)
Possible values: Any 32Bit ARGB color

Tick Point Format (Theme dependent)

The format in which the gauge's tick points are drawn.
Default value: Oval
Possible values: Oval, Rect(-angle)

Draw Values (Theme dependent)

Values can be drawn at the end of the ticks around the gauge. With this property the drawing of the values can be enabled or disabled.
Default value: Enabled
Possible values: Enabled or disabled

Values Distance Factor (Theme dependent)

The distance of the values from the ticks can be modified. The reference value (which equals a factor of 1) is the radius of the gauge object.
Default value: 0.12
Possible values: 0.0 ... 1.0

Font Auto Scaling (Theme dependent)

With this property the automatic scaling of the values next to the ticks can be enabled or disabled. If enabled, the Font size will increase with the size of the object. Please note that the size defined in the Font Attribute[76] is disregarded for the font scaling if this property is enabled.
Default value: Enabled
Possible values: Enabled or disabled.

Font Scaling Factor (Theme dependent)

The scaling factor of the values can be modified. The reference value (which equals a factor of 1) is the radius of the gauge object. Please note that the size defined in the Font Attribute[76] is disregarded for the font scaling.
Default value: 0.09
Possible values: 0.0 ... 1.0

Offset, Scale, Offset 2

Two offsets and a scale value can be defined that will be used in the following way:
displayed value = (original value + Offset 1) * scale + Offset 2
offsets:

Default value: 0
Possible values: -2147483648.0 ... 2147483647.0

scale:
Default value: 1
Possible values: -2147483648.0 ... 2147483647.0

Draw Boundary Bar (Theme dependent)

A colored bar can be drawn around the gauge. The bar is divided in 3 regions, following the min and max value settings. With this property the drawing of the boundary bar can be enabled or disabled.
Default value: Enabled
Possible values: Enabled or disabled

Boundary Bar Width Factor (Theme dependent)

The width of the Boundary Bar can be modified. The reference value (which equals a factor of 1) is the radius of the gauge object.
Default value: 0.02
Possible values: 0.0 ... 1.0

Draw Digital Value (Theme dependent)

A digital values can be drawn inside the gauge. With this property the drawing of the digital value can be enabled or disabled.
Default value: Enabled
Possible values: Enabled or disabled

Digital Value Font Scaling Factor (Theme dependent)

The scaling factor of the digital value can be modified. The reference value (which equals a factor of 1) is the radius of the gauge object.
Default value: 0.25
Possible values: 0.0 ... 1.0

Digital Value Background X Offset Factor (Theme dependent)

The horizontal position of the digital value box inside the gauge can be modified. The reference value (which equals a factor of 1) is the radius of the gauge object.
Default value: 0.35
Possible values: -1.0 ... 1.0

Digital Value Background Y Offset Factor (Theme dependent)

The vertical position of the digital value box inside the gauge can be modified. The reference value (which equals a factor of 1) is the radius of the gauge object.
Default value: 0.35
Possible values: -1.0 ... 1.0

Digital Value Background Width Factor (Theme dependent)

The width of the digital value box inside the gauge can be modified. The reference value (which equals a factor of 1) is the radius of the gauge object.
Default value: 0.45
Possible values: 0.0 ... 2.0

Digital Value Background Height Factor (Theme dependent)

The width of the digital value box inside the gauge can be modified. The reference value (which equals a factor of 1) is the radius of the gauge object.
Default value: 0.3
Possible values: 0.0 ... 2.0

Events

The following events can be configured for the gauge object:
- OnOpenForInput
- OnEnabledWithFocus
- OnEnabledWithoutFocus
- OnValueChange
- OnEnterValue
- OnEnterNewValue
- OnEscape

See [here](#) [430] for the descriptions of all events and possible actions that can be assigned.

6.4.7   Lamp

The lamp object is based on the [button object](#)[463]. It can display two states: On or off / lit or not lit.
The lamp can be turned on or off by a variable reference. 0 or false equals off, 1 or true equals on.
In difference to the button object, the images for the lamp are not included as separate Picture Graphic objects.
Instead, there are two new parameters, Pressed Image and Released Image.
For common object properties of the object refer to [this page](#)[415] . Here only the lamp specific properties are described.
Please note that the "Set as Input" property for the lamp object is disabled by default, as it is rather an indicator than a button. But it can still be enabled to be used like a button.

Lamp Specific Properties

Draw Shadow

A button shadow can be drawn around the button that changes according to pressed or released state of the button. This is a visual help for the user to determine if the button is just pressed or not.
Disable this property if the design of the button is completely custom and the press/release state is distinguishable using the pressed and released objects (see above).
If this property is enabled, a small region between button background (or object inside button) and border of the button is reserved for the shadow. If it is disabled, the background of the button (or object inside the button) goes directly to the border.
Note: Even though the lamp is a round object visually, the button is still rectangle, and so is the shadow, if this property is enabled.
Default value: Disabled
Possible value: Enabled or Disabled

Shadow Color

If "*Draw Shadow*" is enabled one can define the shadow color used to draw the shadow.
The background color of the shadow is the button background color which includes that. the background of the shadow is transparent if the button is configured as transparent.
The button shadow consists of two areas where one is drawn in the opaque color selected and the other is drawn in a half-transparent version of the same color.
Both colors are switched when button is pressed to simulate a kind of 3D effect of the pressed/released button.
Default value: Black.
Possible value: Any color allowed by standard color chooser.

Latching

Latching means: When a button is pressed and released, it toggles state and stays in the toggled state until it is pressed again.
Like that a latching button resembles a light switch whereas a non-latching button resembles a door-bell switch.
With this property this behavior can be enabled.
Default value: Disabled
Possible values: Enabled or Disabled

Pressed State

Switch between the release or pressed state of the button. This is important for having a preview of the object and also for adding pressed/released objects for the button (see above).
Note: If only one child is assigned to the button the other state will be shown as empty on PC tool but on the device this one child will show permanently - button shadow should be used to show button state in this scenario.
This property has no influence to runtime behavior.
Default value: Disabled
Possible values:
- Enabled (button is in pressed state)
- Disabled (button is in released state)

Pressed Object ID

Not used for this object type.

Released Object ID

Not used for this object type.

Toggle Background Image

With this property the usage of the Pressed Image and Released Image can be turned on or off.

Pressed Image

The image set in this property will be used as the pressed image. By default, this will be set to the standard lamp pressed image.

Released Image

The image set in this property will be used as the released image. By default, this will be set to the standard lamp released image.

Events

The following events can be configured for the button object:
- OnPress
- OnRelease
- OnLongPress
- OnLongRelease
- OnDblClick
- OnOpenForInput
- OnEnabledWithFocus
- OnEnabledWithoutFocus
- OnValueChange
- OnCancel

See [here] [430] for the descriptions of all event and possible actions that can be assigned.

6.4.8   Linear Bargraph

The linear bargraph is an object for showing a numerical value in a graph.
The linear bargraph shares a lot of common properties with other objects. See [here] [415] for a description. Here only the linear bargraph specific properties are described.

[Download this project] to see an example how to use the linear bargraph. A PDF is included which explains how the project works.

Linear Bargraph Specific Properties

Direction

The direction of the fill or value line of the bargraph.
For vertical bargraphs a positive direction means that the value grows from bottom to top.
A negative direction means that the value grows from top to bottom.
For horizontal bargraphs a positive direction means that the value grows from left to right. A negative direction means that the value grows from right to left.
Default value: Positive
Possible values: Positive or Negative

Axis Orientation

A bargraph can either be vertical or horizontal. Depending on this property the drawing of the bargraph itself and the value changes accordingly.
Default value: Vertical
Possible values: Vertical or Horizontal

Draw Target Line (Theme dependent)

When enabling this property, a line will be drawn at the position of the [Target Value] [427] property value.
Default value: Enabled
Possible values Enabled or disabled.

Target Line Color (Theme dependent)

The color in which the target line (see above) will be drawn.
Default value:black (RGB 0, 0, 0)

Possible values: Any 24Bit RGB color

Fill Mode (Theme dependent)

If Fill Mode is enabled, the value will be shown as a filled area in the bargraph. If Fill Mode is disabled only a line will be drawn at the value's position.
Default value: Enabled
Possible values: Enabled or disabled.

Draw Ticks (Theme dependent)

It is possible to draw ticks - small lines at the side of the bargraph. The color and the number of the ticks can be configured with the properties below.
Default value: Enabled
Possible values: Enabled or Disabled

Tick Color (Theme dependent)

Define the color of the ticks that shall be drawn with this property. Only has an effect if *Draw Ticks* is enabled.
Default value: black (0, 0, 0)
Possible values: any 24Bit RGB color

Number Of Ticks (Theme dependent)

As the name of this property implies, the number of the ticks that shall be drawn if Draw Ticks is enabled, can be configured with this property.
The ticks are evenly positioned at the bargraph.
Default value: 12
Possible values: 0 ... 32767 (but not greater than the height of a vertical bargraph or width of a horizontal bargraph)

Needle Image (Theme dependent)

A so called Needle Image can be assigned to a bargraph as a value indicator. It will be moved at the horizontal or vertical axis of the bargraph according to it's value.
Please note that there is a little different behavior for SVG images. If an SVG image shall be used as needle, it needs to have the same size as the bargraph and the needle has to be placed directly in the middle of the needle SVG image. The preview in the design tool will also be different to the look at runtime.
Default value: No image
Possible image: Any image from file system

Events

The following events can be configured for the linear bargraph object:
- OnOpenForInput
- OnEnabledWithFocus
- OnEnabledWithoutFocus
- OnValueChange
- OnEnterValue
- OnEnterNewValue
- OnEscape

See here [430] for the descriptions of all event and possible actions that can be assigned.

6.4.9 List

 Tutorial video about the list object
https://www.youtube.com/watch?v=zNNjsMrO4Hk

The list object is a useful object to show other objects based on a value (e.g. two icons in a list showing a different state).
A list always contains other objects. It is possible to either display only one object at a time or multiple objects, this can even be defined for the different navigation states of the list (see properties *Items Visible (Normal Mode)* and *Items Visible (Editing Mode)*). This makes it possible to show something like a drop down menu (1 item visible in normal mode, more visible in editing mode).
Note: At runtime the list object's height is automatically calculated based on the number of currently displayed child items (and their item height). This also means that a list that has no child items will NOT be displayed at all. In the designer the list is always shown in full height based on the properties *Items Visible (Normal Mode)* or *Items Visible (Editing Mode)* even if no or less children are present.
As with all other objects there are some common object properties that are also used by the list object. Se here [415] for a description of the common properties.
The list specific properties are listed below.

List Specific Properties

Item Height

The height of a list object can not be changed via the common *Height* property. The height of the list is defined by a combination of the *Item Height* property, the *Border Width* (and *Draw Border*) and the number of visible items items in the list (see *Items Visible (Normal Mode)* and *Items Visible (Editing Mode)* below).
Use this property to define the height of each item (other object) in the list.
Default value: 30 pixel
Possible values: 1 ... 1280 pixel

List Item Mode

This property is only a preview for the two list modes that are available (normal or editing). It has no effect on runtime behavior.
Note: This property can only be changed if the property *Open For Input* is enabled.
Default value: Normal
Possible values: Normal or Editing

Items Visible (Normal Mode)

With this property it can be defined how many items (or objects) are visible in normal mode of the list (when it is currently not edited (not open for input)).
Default value: 1
Possible values: 1 ... 255

Items Visible (Editing Mode)

When a list is edited (made open for input by the user) it is possible to show more items than in normal mode. Like this it is possible to create a drop down box that shows the user more options.

Note: This property can only be changed if the property *Open For Input* is enabled.
Default value: 1
Possible values: 1 ... 255

List Items

When editing this property a dialog will pop up that provides some ways to change the items in a list
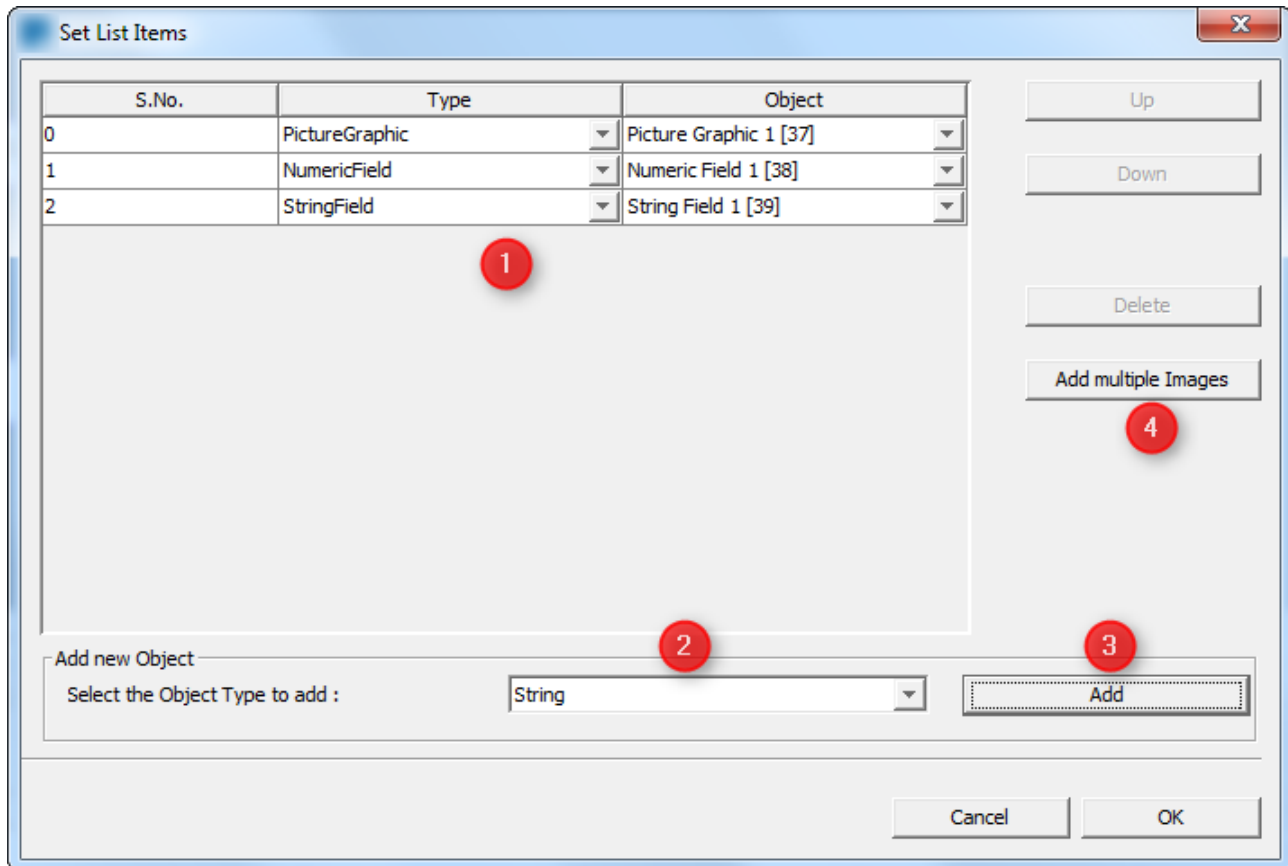
-Add a single object

(1) In the upper part of the dialog, the items of List is shown.
(2) In the lower part of the dialog, objects can be added to the selected list. To change which type of object shall be added, the drop down box *Select the Object Type to add* can be used.
(3) When the settings are made, the *Add* button can be pressed to add the object to the list.
(4) With the button *Add multiple Images* many images can be loaded into the list object at once. New Picture Graphic objects will automatically be created, one for each image.

-Delete a single object

Just select one object in the table and hit the *Delete* button on the right side of the dialog.

-Add multiple images

It is possible to add multiple images to the list in one step.
Click the Add multiple Images button on the right. A file dialog will appear. In this file dialog open as many images (from the same directory) as wanted.
For each selected image a new picture graphic object will automatically created (with default settings) and added to the list.

-Change the order of the items

The table in the dialog shows all items in the list. The one of the top is the one that will be shown when setting value 0 the list. The one on the bottom of the list is the one that is shown when the highest value is set.
To change the order simply select one item in the table and then click either the *Up* or *Down* button on the right side of the dialog.

-Number Of List Items

This property can not be edited. It just shows the number of items inside the list.

-Touch Input (new in 2020.4.2)

This property is not visible in the OPUS Projektor property window. It can be set with JavaScript.
The syntax is as follows:

```
setProperty(<list_id>,"Touch Input",true);
```

As a result, a list object in editing mode will now look like this:
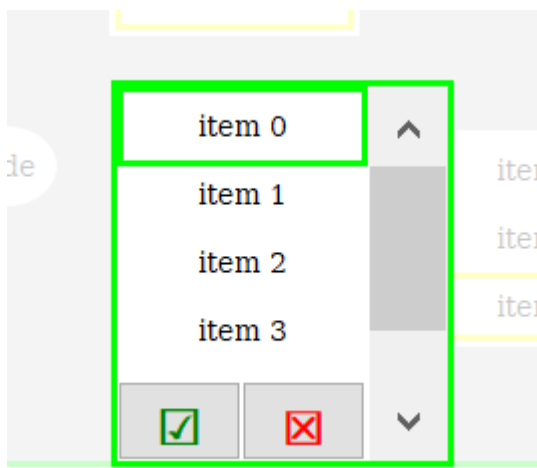
There are two buttons, one Enter, one Escape, and the background of the page will be half-transparent to put visual focus on the edited list.
The default value for this is false, so it needs to be activated e.g. on the event OnPageInit of the page where the list object is.

Behavior details

Some details as to how the list objects behaves in certain special situations:

If only one item is shown the list object shows
- **the item belonging to the value** if the item is valid (not a NULL-ID and not rendered invisible via visibility)
- **nothing** if value is below max value and the item is invalid
- **the last item** if the value is equal to or above max value (and it is visible)

If more than one item is shown the list widget
- **shows as many items as configured** if there are enough valid ones

- **shows all valid items** if there are less than items configured for the list (note that this will result in reduced height of the widget)
- current value/selected item is identified by a focus frame using the 'ENABLED_WITH_FOCUS' state of the item
- if the item does not have such property (e.g. because it's a container) the respective setting of the list is used
- if an invalid item is selected, no focus frame is shown
- the last item is shown if the value is equal to or above max value (and it is visible)
- add an invalid item at the end of the list if nothing should be shown for values above the list's range
- If there is only one valid item in a multi-item list, the list is expanded to two items so that the valid one is always visible
- This shows the 'null item' at the end of the list but avoids showing nothing where as single item list would do that
- While editing only valid items can be selected and the value will be set accordingly

FAQ

How to add an object to a list

There are different ways to add an object to a list:
1. Drag a new object from palette to the list on the designer area
2. Drag an existing object from the project tree to the list in the project tree
3. Edit the property *List Items* (see description below).
4. Copy an existing object and paste it into the list (the size of the pasted object is automatically adjusted so take care when using paste as reference since the original object will also be modified)

How can I show / hide / customize a scroll bar for all list objects when the list is open for input?

Go to the project properties of the project and the to the *Object* section.
On the bottom of the dialog there is an option *Size of List object scrollbar*.
Set this to 0 to disable all scroll bars.
Set this to the width the scroll bars shall have to customize it (if touchscreen editing is preferred, this should be set to at least 50 pixel so that the user is able to touch the scroll bar).
Note that the scroll bar is only shown at runtime.

Events

The following events can be configured for the list object:
- OnOpenForInput
- OnEnabledWithFocus
- OnEnabledWithoutFocus
- OnValueChange
- OnGetValue
- OnEnterValue
- OnEnterNewValue
- OnEscape

See here ^{⧠430}for the descriptions of all event and possible actions that can be assigned.

6.4.10 Meter

 Tutorial video about the meter object
https://www.youtube.com/watch?v=l9mp9Z5xAjM

 Tutorial video about the hour glass demo project
<%HOUGLASS_METER%>

The meter can be used to design gauges or other instruments requiring a needle (indicator). The meter shares many properties with other objects. Find a description of them here[415].

Angles

The angles of a meter can be changed with the Start Angle and End Angle properties (see below).
The spanning direction for the meter is always counterclockwise:
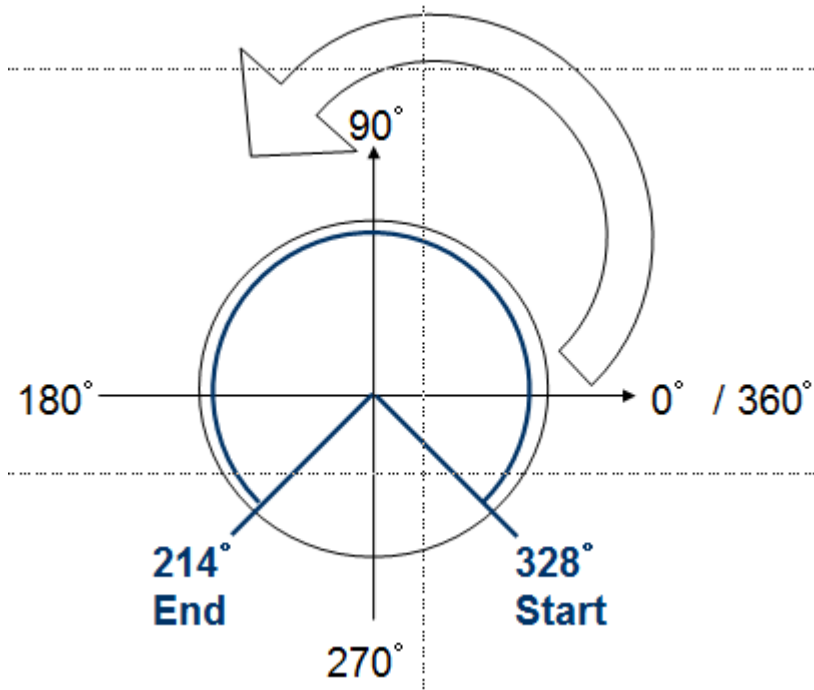
Screenshot with some meter examples:

Start Angle 0°
End Angle 0°

Start Angle 45°
End Angle 0°

Start Angle 0°
End Angle 315°

Start Angle 315°
End Angle 225°

Start Angle 135°
End Angle 225°

Start Angle 270°
End Angle 90°
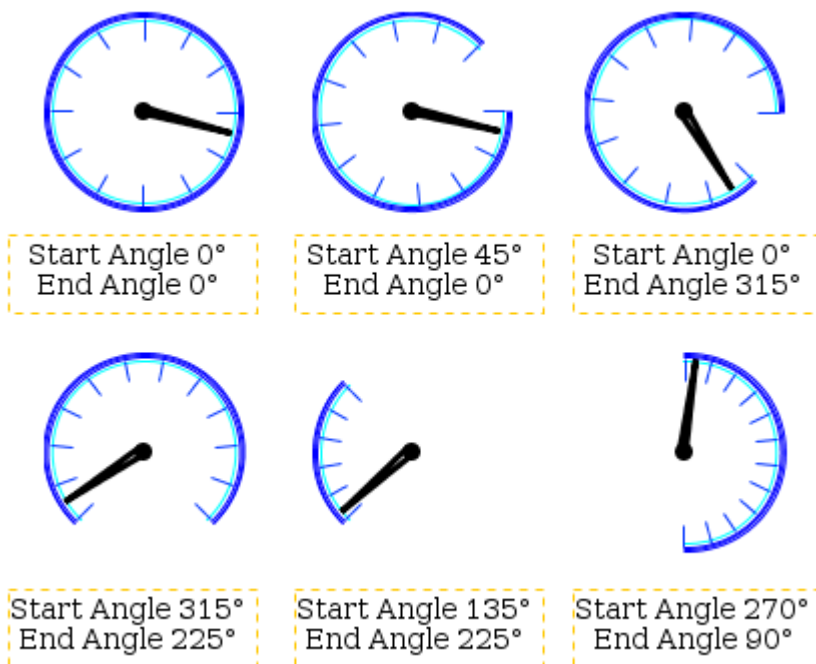
Meter Specific Properties

Start Angle

The start angle of the meter can be defined with this property.
The start angle's 0° position is at 3 o clock.
Default value: 0°
Possible values: 0° ... 360°
See examples above

End Angle

The end angle of the meter can be defined with this property.
The start angle's 0° position is at 3 o clock.
Default value: 0°
Possible values: 0° ... 360°
See examples above

Deflection Direction

The deflection direction defines how the value grows in the meter. Either clockwise or anti clockwise.
Default value: ClockWise
Possible values: ClockWise or AntiClockWise

Needle Color (Theme dependent)

With this property the color of needle can be defined.
Note: This property has no effect if a *Needle Image* is used.
Note: Ses also common properties *Color Above Maximum* and *Color Below Minimum* which also have influence on the needle color.
Default value: Signal blue (RGB 0, 51, 255)
Possible values: Any 24Bit RGB color

Needle Image (Theme dependent)

Instead of the automatically drawn needle, a needle image can be assigned to the meter to show a more beautiful indicator.
The needle image should have the same size as the meter and the needle in the image should always point to 3 o'clock position.
Example needle image:

Default value: No image
Possible values: Any image with supported format (bmp, jpeg, png, svg)
Note: To make the needle image look good when rotated, it is recommended to use scalable vector images (SVG). This prevents steps or anti aliasing artifacts.

Draw Arc (Theme dependent)

The arc is a small colored line around the meter (see *Arc Color* below).
With this property the drawing of the arc can be enabled or disabled.
Default value: Enabled
Possible values: Enabled or disabled

Arc Color (Theme dependent)

The color of the arc of the meter (see above).
Default value: Cyan (RGB 0, 255, 255)
Possible values: Any 24Bit RGB color

Draw Ticks (Theme dependent)

Ticks (small lines as value indicators) can be drawn around the meter. With this property the drawing of the ticks can be enabled or disabled.
Default value: Enabled
Possible values: Enabled or disabled

Tick Color (Theme dependent)

The color in which the meter's ticks are drawn.
Default value: Signal blue (0, 51, 255)
Possible icons: Any 24Bit RGB color

Number Of Ticks (Theme dependent)

The ticks around the meter are evenly spread. The number of ticks can be defined with this property.
Default value: 12
Possible value: 0 ... 360

Events

The following events can be configured for the meter object:
- OnOpenForInput
- OnEnabledWithFocus
- OnEnabledWithoutFocus
- OnValueChange
- OnEnterValue
- OnEnterNewValue
- OnEscape

See [here](#) [430] for the descriptions of all event and possible actions that can be assigned.

6.4.11 Multimedia Player

Tutorial video about the Multimedia Player
https://www.youtube.com/watch?v=ZDMf8XfCmaY

[Download this project](#) to see a basic implementation of the Multimedia Player in a project.

The multimedia player is a complex object which contains other basic objects. Each basic object is registered in the player itself and takes a special function. The player represents a predefined construction kit which is fully customizable.
A media player can be placed on a frame and is used for playing audio or video files. As many other objects, the multimedia player has a lot of common properties. See [here](#) [415] for a description. Only multimedia player specific properties will be described on this page.

Hint: The Multimedia Player object is only available for device panels with audio capabilities. Otherwise you won't see this object in the palette.

<u>Please note:</u>

If you want to use the MP4 formats in your project it is necessary that you purchase a license at https://www.mpegla.com/.
MP3 seems to be free of licenses as of now:
https://www.iis.fraunhofer.de/en/ff/amm/consumer-electronics/mp3.html.
For MP4 you need to license "MPEG-4 Visual (Part 2)" and "AVC/H.264 (MPEG-4 Part 10)" in order to use both formats. If you only want to playback MP4 video files the H.264 license should be enough.
If in doubt, please contact the companies and ask for license assistance.

Multimedia Player Controls / Components

	Name	Type	Events / Actions	Description
1	Play	Button	OnPress Set Media Player Command Command: play Value: Value Not Set OnRelease Set Media Player Command Command: pause Value: Value Not Set	If the "play" command is executed, the current media file will be played. The media time string field (14) and position bargraph (7) will be updated with the current playback time. If no media is set or if there is a playback error, the play button will automatically set to the released state again. If the "pause" command is executed while playback is in progress, the playback will be paused. Executing the pause command while the playback is paused or stopped will have no effect. The button includes two picture graphics for showing play or pause graphics depending on the pressed state.
2	Stop	Button	OnRelease Set Media Player Command	Executing the "stop" command while playback is in progress will stop the playback. The value of the

	Name	Type	Events / Actions	Description
			Command: stop Value: Value Not Set	media time string field (14) will be set to "00:00 / <totalTime>" and the position bargraph's (7) current value will be set to 0. The play button will automatically be set to the released state. If the stop command is executed while no playback is in progress, it will have no effect.

3	Previous	Button	OnRelease
			   Set Media Player Command
			     Command: previous
			     Value: Value Not Set
				When the "previous" command is executed, the media file before the current file in the playlist (17) will be selected. If the current file is the first file in the playlist, pressing the previous button will have no effect.

4	Next	Button	OnRelease
			   Set Media Player Command
			     Command: next
			     Value: Value Not Set
				When the "next" command is executed, the media file after the current file in the playlist (17) will be selected. If the current file is the last file in the playlist, pressing the next button will have no effect.

5	Position Decrease	Button	OnRelease
			   Set Media Player Command
			     Command: set position
			     Value: -10
				By executing the "set position" command without a value, the current value of the position bargraph (7) is used to set the playback position of the current file. If a value is provided, this value is used as an offset to set the current position as long as it is not longer than the total length of the current media file.

6	Position Increase	Button	OnRelease
			   Set Media Player Command
			     Command: set position
			     Value: +10	

7	Position Bar	Linear Bargraph	OnValueChange
			   Set Media Player Command
			     Command: set position
			     Value: Value Not Set	

8	Shuffle	Button	OnPress
			   Set Media Player Command
			     Command: shuffle
			     Value: Value Not Set
			
			OnRelease
			   Set Media Player Command
				If the "shuffle" command is executed, the playlist order will be randomized. If the media player is in play state, the current selected file shall be still continued.
				
				The button includes two picture graphics for showing shuffle or not

	Name	Type	Events / Actions	Description
			Command: shuffle Value: Value Not Set	shuffle graphics depending on the pressed state.

9	Loop	List	OnValueChange
			   Set Media Player Command
			      Command: loop
			      Value: Value Not Set
				By executing the "loop" command, the next loop mode will be applied. The inital loop mode is 2.
				
				This list shall have three picture graphics as children. Each represents a different loop mode:
				
				At value 0: Loop off (image shows a not-closed ellipse)
				
				At value 1: Loop current file (image shows ellipse with two arrow heads)
				
				At value 2: Loop all (when end of playlist was reached, jump to start of playlist) (image shows ellipse with one arrow head)

10	Volume Decrease	Button	OnRelease
			   Set Media Player Command
			      Command: set volume
			      Value: -10
				When executing the "set volume" command without a value, the current value of the volume bargraph (12) is used to set the volume of the current playback. If a value is provided, this value is used to set the volume of the current playback. The volume bargraph will then be updated with the set value.

11	Volume Increase	Button	OnRelease
			   Set Media Player Command
			      Command: set volume
			      Value: +10	

12	Volume Bar	Linear Bargraph	OnValueChange
			   Set Media Player Command
			      Command: set volume
			      Value: Value Not Set
				The volume settings will also work if currently no media file is selected, they will be applied as soon as a media file is set. The volume settings are applied to all media files of this multimedia player and are not specific to one certain file.

13	Volume ON/OFF	Button	OnPress
			   Set Media Player Command
			      Command: mute
			      Value: Value Not Set
			
			OnRelease
				When executing the "mute" command, the volume of the currently played back media file is internally set to 0. The value of the volume bargraph (12) will not be changed and remains the same.

	Name	Type	Events / Actions	Description
			Set Media Player Command 　　Command: unmute 　　Value: Value Not Set	When executing the "unmute" command, the current volume of the volume bargraph (12) is used to set the volume of the currently played back media file. The button includes two picture graphics for showing mute or unmute graphics depending on the pressed state.
14	Media Time	String Field		This information string field shows the current time of the playback and the total time of the media file (e.g. "00:42 / 03:14")
15	Media Title	String Field		The Field shows the current played or paused media file
16	Display	Container		The display is an content area which showing the video or the album art of the audio file. (If not available, nothing is shown here).
17	Playlist	Table		The playlist shows the currently selected media files at run and design time. Hints: At design time there are no scrollbars visible. At runtime you can scroll the playlist by changing its "Preview Value" or via encoder after enabling the "Set As Input" property.

Multimedia Player State Machine

The following graphic shows the implemented state machine of the multimedia player object at runtime.
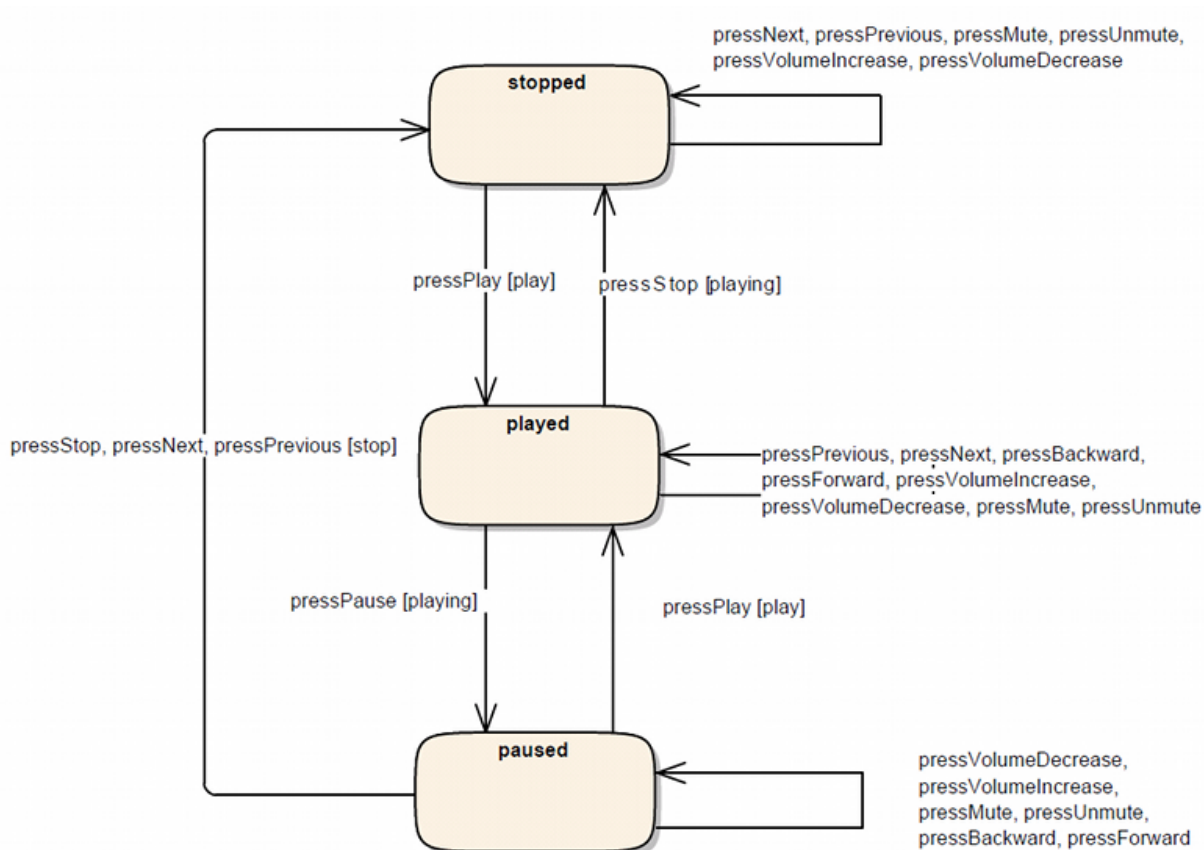
Multimedia Player Specific Properties

The graphic below shows the default property states of the multimedia player object.

Fit To Frame Size

Enable this property to adapt the size of the multimedia player to the parent frame object.

Default value: Disabled
Possible values:
- Disabled: The Multimedia Player can be dynamically scaled to the desired size.
- Enabled: The Multimedia Player will be scaled to the parent frame size.

Auto Rearrange Controls

This property activates / deactivates the internal layout manager of the multimedia player object. The manager takes

care about the positioning and sizing of all included objects in design time. The layout manager has no effect at runtime.
Resizing is only done when the object size is changed.

Default value: Enabled
Possible values:

- Disabled: The Multimedia Player objects will not be sized and positioned when changing the size of the Multimedia Player object.
- Enabled: The Multimedia Player objects will be sized and positioned automatically when changing the size of the Multimedia Player object.

Show Display

This feature removes the display container to use the multimedia player as a pure audio player or as a playback control bar:

Default value: Enabled
Possible values:

- Disabled: Removes the display container.
- Enabled: Adds a display container.

Show Playlist

The playlist shows the currently selected media files at runtime and design time. But in the latter case without scrollbars.

Default value: Disabled
Possible values:

- Disabled: Removes the playlist table.
- Enabled: Add a new playlist table.

Show Backward / Forward Buttons

If the target device has no encoder to set the media position by using the position bar, this property can be enabled
to add a backward and a forward button to the multimedia player. With these buttons the media position can be increased or decreased

Default value: Disabled
Possible values:

- Disabled: Removes the backward / forward buttons from the multimedia player.
- Enabled: Adds the backward / forward buttons to the multimedia player.

Show Volume Bar Buttons

If the target device has no encoder to set the volume by using the volume bar, this property can be enabled to add a volume increase and decrease button to the multimedia player. With these buttons the volume value can be increased or decreased

Default value: Disabled

Possible values:
- Disabled: Removes the volume increase / decrease buttons).
- Enabled: Add's the volume increase / decrease buttons.

Full Path In Title

This property shows the full path of the played media file in the title string field at runtime instead of only the file name.

Default value: Disabled
Possible values:
- Disabled: The media title stringfield shows only the filename
- Enabled: The media title stringfield shows the full path of the played media file.

Playlist Media Files

By clicking on this property, a media file chooser dialog will be displayed. Here the user can modify his playlist.
This dialog supports the following operations: "Add" (1), "Remove" (2), "Up" (3) (playlist position), "Down" (4) (playlist position).

Default value: empty
Possible values: Multiple media files in the supported formats (mp3, mp4, mka, mkv, mp4, oga, ogg, ogv, wav, wma, wmv).
See the FAQ[484] section below about codec information
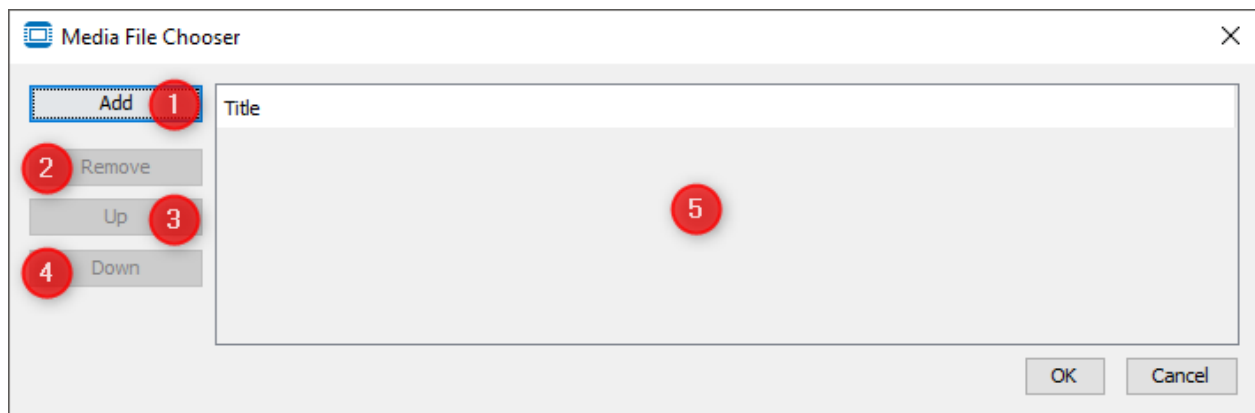
Playlist Width

This value adjusts the width of the playlist if it is shown. The state of this property is depending on the value of the "Show Playlist" property.

Default value: Disabled
Possible values: 1 ... width of the multimedia player object - 2 x (default component gap = 2 + border width) -1

Multimedia Player Control Properties

The ID's of all the control objects are stored as properties in the media player object. If an object is not needed , it can be deleted.

The id of deleted control objects will be set to "-1." Of course it is also possible to add / register a control object again.

Restrictions for the registration of new control objects:

- Only objects which are included in the multimedia player component can be registered.

- The object which is to be registered must contain all required events, because the default events/actions will be registered again.
(e.g. it is also possible to register an arched bargraph object as volume bar instead the default linear bargraph object, because both objects have the "OnValueChange" event)

Hint: If the assignment of the new control object fails, the last valid id will retain

The group "Multimedia Player Controls" includes the following properties:

Multimedia Player Controls	
Play Button ID	65
Stop Button ID	66
Previous Button ID	67
Next Button ID	68
Shuffle Button ID	69
Loop List ID	70
Volume Button ID	72
Volume Bar ID	71
Volume Increase Button ID	-1
Volume Decrease Button ID	-1
Position Bar ID	75
Forward Button ID	-1
Backward Button ID	-1
Title StringField ID	73
Position StringField ID	74
Display ID	76
Playlist ID	-1

Events

The following events can be configured for the multimedia player object:
- OnPress
- OnRelease

See <u>here</u> [430] for the descriptions of all event and possible actions that can be assigned.

FAQ

What video and audio codecs are supported?

Video

The following video codecs have been verified on the device:
- H264 - MPEG-4 AVC (part 10) (avc1)
- MPEG-1/2 Video (mpgv)
- Motion JPEG Video (MJPG)
- Google/On2's VP8 Video (VP80)

The following video codecs are currently NOT SUPPORTED on the device:
- Theora Video
- MPEG-4 Video (DX50) => DIVX version 5
- Any Windows media format in an AFS container (VC1, WMV)

Audio

The following audio codecs have been verified on the device:
- MPEG Audio layer 1/2/3 (mpga)
- Vorbis Audio (vorb)
- Raw wave files in different format (tested: PCM S16 LE (araw), PCM U8 (araw))

Why should VLC be installed?

For using the multimedia player object it is NOT neccessary that VLC is installed on your PC but it can help in the following topics:
- Check what codecs are used by specific media file
- Convert (transcode) a media file to a supported codec

VLC can be downloaded here. It is open source software and can be installed without charge (please consider a donation :)).

What is the suggested video and audio codec and how to convert my video?

We suggest to use .mp4 files with "H264 - MPEG-4 AVC (part 10) (avc1)" as video codec and "MPEG AAC Audio (mp4a)" as audio codec.
To convert to a supported codec, VLC (see above) can be used. Here is a short manual.

What is the difference between the multimedia playback on Windows simulation and on the device

The frontend for playing back media files with the MultimediaPlayer object is the same on both platforms. However, the backend is completely different. On the Windows PC the simulation uses the capabilities of the operating system and the installed software codecs. On the device the hardware decoding functions of the CPU are directly used to minimize CPU load.
Because of this difference, the playback can be different on both platforms. One of the biggest differences can be that a media file that can be played back in the simulation on the Windows PC cannot be played back on the device. This can happen if a codec is installed on the PC but is not supported by the CPU on the device. Other differences that may appear are different timings when executing a multimedia action or (e.g. when going to the next file in the playlist the total time of the next file might appear faster on one platform than on the other).
We strongly recommend that testing is not only done with the Windows simulation but also directly on the device!

Why is my media file not playing in the Windows simulation?

There might be a codec missing. The **K-Lite Codec Pack** supports all the codecs supported on the device (and many more). After installing this codec pack, restart the simulation and try again.
Download it here (the basic version should be enough).

Do I need to be concerned about performance when playing video or audio?

Yes and no. When using one of the supported codecs above, the built in hardware acceleration is used on the device to minimize CPU load (while playback the CPU load increses between 3% and 7% on a single core CPU).
BUT there are some things to keep in mind: When running on a single core CPU it is more likely that playback of multimedia has an influence on the system. Especially tight CAN timings may be a problem (e.g. sending out a CAN message every 10ms might not be completely precise anymore). We strongly recommend that the project is tested on the device while playing back media with a special look at CAN (send and receive) and script timings.
When running on a device with a dual or quad core CPU it is less likely that media playback has a negative influence on the overall performance.

What happens when the page is changed while playback?

It depends on what kind of media is played back:
If a video is played back when exiting the page with the multimedia player, the playback will be stopped automatically.
If only audio (music) is played back when exiting the page with the multimedia player, the playback will continue. If the audio playback shall be stopped when the page is switched, you have to stop the media player manually (e.g. via a multimediaCommand JavaScript[545] call onPageExit).

Why is the volume so low?

Please have a look at the volume FAQ section[754].

How can I control playback in JavaScript?

Use the multimediaCommand function[545] in JavaScript to control the multimedia player and it's playlist.

How can I switch the video to fullscreen and how to exit fullscreen mode?

Use the multimediaCommand function[545] in JavaScript with the command string "toggle fullscreen" to enable or disable fullscreen video. Fullscreen mode is automatically disabled again as soon as the user touches the screen or presses the escape key.

How can the display operator play his own media files?

It is possible to add media files that are stored on an USB device to the playlist of the MultimediPlayer. To do this JavaScript[514] needs to be used. Here is an example script that adds all files from the folder "music" on an usb stick to the playlist:

```
//check if USB stick is inserted:
var usbInserted = getVariableValue("@USBMemStatus");
if (usbInserted != 0)
{
    //USB stick present, get the mount path
    var usbPath = getVariableValue("@USBMountPath");
    //add "music" folder to the usb path
    usbPath = usbPath + "/music";
    //remove all files from playlist
    multimediaCommand(34, "clear playlist");
    //now add all files from the music directory to the
playlist
    multimediaCommand(34, "add directory to playlist",
usbPath);
```

```
}
else
{
//no usb stick present, show alarm 1 that shows an eror
message
    setVariableValue("@AlarmShow", 0x8001);
}
```

Why can't I change the position in an MP3 file?

Changing position (seeking) for MP3 encoded audio files is currently not supported by the backend. If seeking is important, the MP3 files can be converted to e.g. MPEG AAC or Vorbis audio files.

6.4.12   Numeric Field

Tutorial video about the Numeric Field
https://www.youtube.com/watch?v=LMaRboYi8W4

The numeric field object is designed to show numbers on the screen. It can be customized like any other object using it's properties.
The numeric field has a lot of common properties. See here [415]for a description of them. Here only the numeric field specific properties are described.

Numeric Field Specific Properties

Format

Choose between normal decimal representation of the displayed number or a scientific exponential representation.
Examples:
Decimal value: 15, scientific notation / floating-point representation: 1.5e+1
Decimal value: 0.1, scientific notation / floating-point representation: 1.0e-1
Decimal value: 32767, scientific notation / floating-point representation: 3.2767e+4

Default value: Decimal
Possible values: Decimal or Exponential

Leading Zeros

It is possible to display zeros (0) before the actual value of the numeric field.
This is useful if always a fixed number of digits shall be displayed (e.g. when showing the time: without leading zeros 9:4, with leading zeros: 09:04).
The number of zeros that are displayed is dependent on the width of the object. The whole available space will be filled.
Default value: Disabled
Possible values: Disabled or enabled

Display Zeros As Blanks

If this property is enabled, the value 0 will not be displayed in the numeric field.
This could be useful if e.g. only a deviation shall be displayed but nothing if the value is right at target.

Default value: Disabled
Possible values: Enabled or Disabled

Offset, Scale, Offset 2

Two offsets and a scale value can be defined that will be used in the following way:
displayed value = (original value + Offset 1) * scale + Offset 2
offsets:
Default value: 0.0
Possible values: -2147483648.0 ... 2147483647.0

scale:
Default value: 1.0
Possible values: -2147483648.0 ... 2147483647.0

Values can be entered in decimal form or in scientific notation / floating-point representation.

No Of Decimals

The number of decimals property allows to define how many digits will be displayed behind the decimal point.
Note that values behind the decimal point can only be displayed if the Scale property is not 1.0.
Default value: 1
Possible values: 0 ... 10

Events

The following events can be configured for the numeric field object:
- OnOpenForInput
- OnEnabledWithFocus
- OnEnabledWithoutFocus
- OnValueChange
- OnEnterValue
- OnEnterNewValue
- OnEscape

See [here](#) [430] for the descriptions of all event and possible actions that can be assigned.

6.4.13 Picture Graphic

The Picture Graphic object is a simple object to display static images on the screen.
It is also capable of displaying animated GIF images that makes it possible to show small movies and animations. See GIF properties below.

Note: A free online GIF editor / creator can be found [here](#).

The picture graphic shares some [common properties](#) [415] with other objects.

This section only describes the properties that are Picture Graphic specific or differ with the common properties.

Picture Graphic Specific properties

Image To Display

To choose the image that shall be displayed in a Picture Graphic use this property.
Any supported image file can be selected from the file system. Supported formats are:
- Bitmap Windows Bitmap (*.bmp)
- JPEG Joint Photographic Experts Group (*.jpg, *.jpeg, *.jpe)
- PNG Portable Network Graphics (*.png)
- GIF Graphics Interchange Forrmat (*.gif)
- SVG Scalable Vector Graphics (*.svg)
- TIFF Tagged Image File Format (*.tif, *.tiff) (only single layer images. With multilayer images only the first layer will be used)

In general all images are scaled to size of the Picture Graphic object and converted to PNG format when saving the project.
Two exceptions exist:
- GIF images will neither be scaled nor converted. They will be saved 1:1 to the device so that GIF animations can be replayed. Note that GIF files are only supported as Image to Display
- SVG images are also not scaled or converted when saving the project. They will be scaled directly on the target.

In general, please use images in the exact resolution that they will be displayed in on the device. Higher resolutions don't improve the quality, they lead to a higher memory demand and the scaling can produce a lower image quality.

Fit To Image Size

Enable this property to change the size of the Picture Graphic object to the size of the image selected with Image To Display property.
Default value: Enabled
Possible values:
- Disabled: Image will be scaled to size of Picture Graphic
- Enabled: Picture Graphic will be scaled to image size

Picture Graphic GIF properties

The GIF properties are not displayed in the properties table of the Picture Graphic object. They can only be accessed with the JavaScript functions getProperty()[531] and setProperty()[538].
WARNING: Using large animated gif images can slow down the device at runtime significantly. Testing gif images on the target device is strongly recommended!
The GIF properties allow to control the playback of GIF animations on the device. The following properties are available:

GIF State

Use the GIF State property to control the playback of the video. It can be started, paused or stopped. The values for the GIF State are integers, see below.
Default value: 0 (Stop)

Possible values:
- 0 (Stop)
- 1 (Play)
- 2 (Pause)

Example for how to use this property in a JavaScript:

```
var objectID = 42;

//Start the GIF animation
setProperty(objectID, "GIF State", 1);

//Pause the GIF animation:
setProperty(objectID, "GIF State", 2);

//Stop the GIF animation
setProperty(objectID, "GIF State", 0);
```

Note that the default value is stopped. This means that each animation on each page has to be started manually in a script. If this is not done, only the first frame of the animation will be displayed.

GIF Speed

The playback speed of a GIF animation can be set with this property. If this property is not changed, the animation will run with the speed configured in the GIF itself.
Note that large GIF animations might run a little slow when playing them the first time (the images will be cached the first time it is run).
The value for this property is the percent speed of the original speed defined in the GIF (100% means unchanged speed).
Default value: 100%
Possible values: 1 ... 34359738367% (it is not recommended to set such a high speed)
Example for setting 50% (half) speed for an animation:

```
var objectID = 42;
setProperty(objectID, "GIF Speed", 50);
```

GIF Loops

This property is a boolean property that controls if the animation will loop to the start of the animation when it's end is reached.
Default value: true
Possible values:
- true - Begin at start of animation when end of animation is reached
- false - Stop animation when end of animation is reached

Integrated in GIF files is a number of loop repetitions information, so even if looping is set to true, after a certain amount of repititions the animation will stop. Sadly this information is hardly editable in GIF-editors.
Example for setting the GIF Loops property:

```
var objectID = 42;

//Stop animation when end is reached:
setProperty(objectID, "GIF Loops", false);

//Enable looping:
setProperty(objectID, "GIF Loops", true);
```

GIF Frame Number

It is possible to jump to a certain frame number of the GIF animation. Like this an animation can be played frame by frame or jump to a special frame.
Default value: 0
Possible values: 0 (first frame) ... number of frames of the GIF animation
If a number greater than the actual number of frames of the animation is set, it will be ignored.
Example:

```
var objectID = 42;

//Jump to frame 5
setProperty(objectID, "GIF Frame Number", 5);
```

Note: Frame counting starts at 0.

6.4.14   Power Switch

The power switch object is based on the [button object](#)[463]. It can display two states: On or off.
The power switch can be turned on or off by a variable reference. 0 or false equals off, 1 or true equals on.
In difference to the button object, the images for the power switch are not included as separate Picture Graphic objects.
Instead, there are two new parameters, Pressed Image and Released Image.
For common object properties of the object refer to [this page](#)[415] . Here only the power switch specific properties are described.
Please note that for this object the "Latching" property is activated by default, since a power switch is usually used like that.

Power Switch Specific Properties

Draw Shadow

A button shadow can be drawn around the button that changes according to pressed or released state of the button. This is a visual help for the user to determine if the button is just pressed or not.
Disable this property if the design of the button is completely custom and the press/release state is distinguishable using the pressed and released objects (see above).
If this property is enabled, a small region between button background (or object inside button) and border of the button is reserved for the shadow. If it is disabled, the background of the button (or object inside the button) goes directly to the border.
Note: Even though the lamp is a round object visually, the button is still rectangle, and so is the shadow, if this property is enabled.
Default value: Disabled
Possible value: Enabled or Disabled

Shadow Color

If "*Draw Shadow*" is enabled one can define the shadow color used to draw the shadow.
The background color of the shadow is the button background color which includes that. the background of the shadow is transparent if the button is configured as transparent.
The button shadow consists of two areas where one is drawn in the opaque color selected and the other is drawn in a half-transparent version of the same color.

Both colors are switched when button is pressed to simulate a kind of 3D effect of the pressed/released button.
Default value: Black.
Possible value: Any color allowed by standard color chooser.

Latching

Latching means: When a button is pressed and released, it toggles state and stays in the toggled state until it is pressed again.
Like that a latching button resembles a light switch whereas a non-latching button resembles a door-bell switch.
With this property this behavior can be enabled.
Default value: Disabled
Possible values: Enabled or Disabled

Pressed State

Switch between the release or pressed state of the button. This is important for having a preview of the object and also for adding pressed/released objects for the button (see above).
Note: If only one child is assigned to the button the other state will be shown as empty on PC tool but on the device this one child will show permanently - button shadow should be used to show button state in this scenario.
This property has no influence to runtime behavior.
Default value: Disabled
Possible values:
- Enabled (button is in pressed state)
- Disabled (button is in released state)

Pressed Object ID

Not used for this object type.

Released Object ID

Not used for this object type.

Toggle Background Image

With this property the usage of the Pressed Image and Released Image can be turned on or off.

Pressed Image

The image set in this property will be used as the pressed image. By default, this will be set to the standard power switch pressed image.

Released Image

The image set in this property will be used as the released image. By default, this will be set to the standard power switch released image.

Events

The following events can be configured for the button object:
- OnPress
- OnRelease

- OnLongPress
- OnLongRelease
- OnDblClick
- OnOpenForInput
- OnEnabledWithFocus
- OnEnabledWithoutFocus
- OnValueChange
- OnCancel

See [here](#) [430] for the descriptions of all event and possible actions that can be assigned.

6.4.15   Push Switch

The push switch object is based on the [button object](#)[463] . It can display two states: On or off.
The push switch can be turned on or off by a variable reference. 0 or false equals off, 1 or true equals on.
In difference to the button object, the images for the push switch are not included as separate Picture Graphic objects.
Instead, there are two new parameters, Pressed Image and Released Image.
For common object properties of the object refer to [this page](#)[415] . Here only the push switch specific properties are described.

Push Switch Specific Properties

Draw Shadow

A button shadow can be drawn around the button that changes according to pressed or released state of the button. This is a visual help for the user to determine if the button is just pressed or not.
Disable this property if the design of the button is completely custom and the press/release state is distinguishable using the pressed and released objects (see above).
If this property is enabled, a small region between button background (or object inside button) and border of the button is reserved for the shadow. If it is disabled, the background of the button (or object inside the button) goes directly to the border.
Note: Even though the lamp is a round object visually, the button is still rectangle, and so is the shadow, if this property is enabled.
Default value: Disabled
Possible value: Enabled or Disabled

Shadow Color

If "*Draw Shadow*" is enabled one can define the shadow color used to draw the shadow.
The background color of the shadow is the button background color which includes that. the background of the shadow is transparent if the button is configured as transparent.
The button shadow consists of two areas where one is drawn in the opaque color selected and the other is drawn in a half-transparent version of the same color.
Both colors are switched when button is pressed to simulate a kind of 3D effect of the pressed/released button.
Default value: Black.
Possible value: Any color allowed by standard color chooser.

Latching

Latching means: When a button is pressed and released, it toggles state and stays in the toggled state until it is pressed again.
Like that a latching button resembles a light switch whereas a non-latching button resembles a door-bell switch.
With this property this behavior can be enabled.
Default value: Disabled
Possible values: Enabled or Disabled

Pressed State

Switch between the release or pressed state of the button. This is important for having a preview of the object and also for adding pressed/released objects for the button (see above).
Note: If only one child is assigned to the button the other state will be shown as empty on PC tool but on the device this one child will show permanently - button shadow should be used to show button state in this scenario.
This property has no influence to runtime behavior.
Default value: Disabled
Possible values:
- Enabled (button is in pressed state)
- Disabled (button is in released state)

Pressed Object ID

Not used for this object type.

Released Object ID

Not used for this object type.

Toggle Background Image

With this property the usage of the Pressed Image and Released Image can be turned on or off.

Pressed Image

The image set in this property will be used as the pressed image. By default, this will be set to the standard push switch pressed image.

Released Image

The image set in this property will be used as the released image. By default, this will be set to the standard push switch released image.

Events

The following events can be configured for the button object:
- OnPress
- OnRelease
- OnLongPress
- OnLongRelease
- OnDblClick
- OnOpenForInput
- OnEnabledWithFocus
- OnEnabledWithoutFocus

- OnValueChange
- OnCancel

See [here](#) [430] for the descriptions of all event and possible actions that can be assigned.

6.4.16 Rocker Switch

The rocker switch object is based on the [button object](#) [463] . It can display two states: On or off.
The rocker switch can be turned on or off by a variable reference. 0 or false equals off, 1 or true equals on.
In difference to the button object, the images for the rocker switch are not included as separate Picture Graphic objects.
Instead, there are two new parameters, Pressed Image and Released Image.
For common object properties of the object refer to [this page](#) [415] . Here only the rocker switch specific properties are described.
Please note that for this object the "Latching" property is activated by default, since a rocker switch is usually used like that.

Note that the Rocker Switch comes in horizontal and vertical. They are identical except for the orientation of the rocker switch images.

Rocker Switch Specific Properties

Draw Shadow

A button shadow can be drawn around the button that changes according to pressed or released state of the button. This is a visual help for the user to determine if the button is just pressed or not.
Disable this property if the design of the button is completely custom and the press/release state is distinguishable using the pressed and released objects (see above).
If this property is enabled, a small region between button background (or object inside button) and border of the button is reserved for the shadow. If it is disabled, the background of the button (or object inside the button) goes directly to the border.
Note: The shadow will not be painted for this object, it doesn't make sense to enable it.
Default value: Disabled
Possible value: Enabled or Disabled

Shadow Color

If "*Draw Shadow*" is enabled one can define the shadow color used to draw the shadow.
The background color of the shadow is the button background color which includes that. the background of the shadow is transparent if the button is configured as transparent.
The button shadow consists of two areas where one is drawn in the opaque color selected and the other is drawn in a half-transparent version of the same color.
Both colors are switched when button is pressed to simulate a kind of 3D effect of the pressed/released button.
Default value: Black.
Possible value: Any color allowed by standard color chooser.

Latching

Latching means: When a button is pressed and released, it toggles state and stays in the toggled state until it is pressed again.
Like that a latching button resembles a light switch whereas a non-latching button resembles a door-bell switch.

With this property this behavior can be enabled.
Default value: Disabled
Possible values: Enabled or Disabled

Pressed State

Switch between the release or pressed state of the button. This is important for having a preview of the object and also for adding pressed/released objects for the button (see above).
Note: If only one child is assigned to the button the other state will be shown as empty on PC tool but on the device this one child will show permanently - button shadow should be used to show button state in this scenario.
This property has no influence to runtime behavior.
Default value: Disabled
Possible values:
- Enabled (button is in pressed state)
- Disabled (button is in released state)

Pressed Object ID

Not used for this object type.

Released Object ID

Not used for this object type.

Pressed Image

The image set in this property will be used as the pressed image. By default, this will be set to the standard rocker switch pressed image.

Released Image

The image set in this property will be used as the released image. By default, this will be set to the standard rocker switch released image.

Events

The following events can be configured for the button object:
- OnPress
- OnRelease
- OnLongPress
- OnLongRelease
- OnDblClick
- OnOpenForInput
- OnEnabledWithFocus
- OnEnabledWithoutFocus
- OnValueChange

See here [430] for the descriptions of all event and possible actions that can be assigned.

6.4.17   String Field

The string field object is designed to show text on the screen. It can be customized like any other object using it's properties.
The string field has a lot of common properties. See here [415] for a description of them. Here only the string field specific properties are described.

Besides the Auto Wrap property, Texts in String Fields can also have custom line breaks by inserting "\n" (without the "") in the text.
The line break can also be included at runtime via JavaScript by adding \n in a String variable and having that referenced to the String Field or, alternatively by putting the text into the String Field with [setProperty](#)⁵³⁸ .

Please note that for existing projects in the rare case the text "\n" should be used in a String Field, this will now be interpreted as a line break.
In that case, please change "\n" to e.g. "\ n" to avoid the line break.

String Field Specific Properties

Auto Wrap

When Auto Wrap is enabled, the text inside the string field will be automatically wrapped to the next line if the text does not fit inside the string field. If possible, wrapping will leave the words in the text complete. If the string field is to narrow to display even a single word, the single word will be splited.
If Auto Wrap is disabled, text that does not fit in the string field will be clipped.
Default value: Disabled
Possible values: Disabled or Enabled

Max String Length (Characters)

The maximum length of the text displayed in the string field can be adjusted with this property.
Note that the value of a referenced variable or the Preview Value of the string field may have more characters but they are just not displayed if it's more than Max String Length.
Default value: 32
Possible values: 0 ... 65536

Events

The following events can be configured for the string field object:
- OnOpenForInput
- OnEnabledWithFocus
- OnEnabledWithoutFocus
- OnValueChange
- OnEnterValue
- OnEnterNewValue
- OnEscape

See [here](#) ⁴³⁰ for the descriptions of all event and possible actions that can be assigned.

6.4.18 Table

A table can be used to display multiple lines of text. Each line of this text consist of the table cell values delimited by a configurable delimiting character.
Example text for a table with **;** as delimiter character:

| Line1Column1;Line1Column2;Line1Column3;Line1Column4 |
| Line2Column1;Line2Column2;Line2Column3;Line2Column4 |
| Line3Column1;Line3Column2;Line3Column3;Line3Column4 |

The source for the text can either be automatic (e.g. from J1939 DM settings) or set manually via a JavaScript (set the text directly (with setProperty and property "Table Data") or read e.g. a log file, see JavaScript readFile function [539] for an example).
It is not possible to give a preview of the text in the designer tool. The text will be only available at runtime.

Properties of the table

General

The table shares the same general properties with all other objects.
The table may not be language or unit dependent.

Variable References

Variable Reference

Here a number variable can be assigned to the table. With this variable the table can be scrolled if the content can not be dsiplayed completely. Set value 0 to scroll to top, any other value will scroll the value down (depending on number of lines in the table).
Default: No Variable
Possible Values: Any number variable

Bit Mask

Same as for other objects.

Skin Properties

Skin properties are the same as for other objects.

Visibility

Visibility properties are the same as for other objects.

Input Configuration

The input configuration properties are the same as for other objects. Like with an assigned variable it is possible to manually scroll through the table if the property **Set As Input** is enabled.

Value Related

The **Preview Value** property will have no influence on the table in the tool. With JavaScript you could set that property to scroll through the table but it is recommended to just set the property **Set As Input** or attach a variable for scrolling.

Formatting

Font Attribute

Same as for other objects. The font attribute selected here will be applied to the text shown in the table.

Font

Same as for other objects. The font selected here is used as the font for the table's text.

Horizontal Alignment

Same as for other objects. All text in the table will be aligned either center, left or right.

Vertical Alignment

Same as for other objects. All text in the table will be aligned either center, top or bottom.

Table Specific

Grid Color (Theme dependent)

This represents the color of the "grid" (the inner border) of the table.
NOTE: If you want to change this property via JavaScript, the property name "Foreground Color" needs to be used instead of "Grid Color".
Default: black (0, 0 ,0)
Possible values: any color

Table Data Delimiter

This is the delimiter char that is used to mark the beginning of a new column inside each line of the table's text data. If the text data source contains another character as delimiter, enter this here.
Default: **;**
Possible values: any single character

Table Heading Background Color (Theme dependent)

If the property **Show Table Heading** is enabled, a heading line will be displayed at top of the table. With this property a background color for the heading can be set.
Default: grey (153, 153 ,153)
Possible values: any color

Table Heading Font Color (Theme dependent)

If the property **Show Table Heading** is enabled, a heading line will be displayed at top of the table. With this property a font color for the heading can be set.
Default: white (255, 255 ,255)
Possible values: any color

Table Heading Text

A static text that will be displayed if Show Table Heading is enabled. Text for all columns can be entered, every column is delimited by the Table Data Delimiter.
Default: <no text>
Possible values: any text
Example (with delimiter ;):
This;is;a;table
will result in:

This	is	a	table

Language dependancy is not possible for table heading texts. If you need language dependant headings, please use String Fields, put them above the table and configure them as language dependant.

Show Table Heading

Show or hide the heading of the table. See other heading properties above.

Default: disabled
Possible values: enabled or disabled

Number of Table Columns

This property displays the current number of columns of the table.
The value will automatically set by enter the **Table Column Spacing** property below.

Table Column Spacing

With this property the number of columns and their spacing can be configured. To do this the spacing value in percent of each column needs to be entered delimited by the **Table Data Delimiter**.
Default: 25;25;25;25 (each column uses 25% of the table width)
Possible values: any combination that's sum is 100%
NOTE:
If the sum is not 100%, the last column will increased to fit the 100%
If the sum is over 100%, the value will reverted to the last valid
This property has an effect to other properties that use that value: **Table Heading Text** and **Number of Table Columns**

Highlighted Row

This property is not visible in the designer and can only be accessed via JavaScript's getProperty() and setProperty() functions.
It is possible to highlight a row in the table. Row counting starts at 0. If a row is highlighted, the font and background color will be shown inverted. If a row is highlighted that is currently not visible, the table will automatically scroll to that row.
Default: -1 (no row highlighted)
Possible values: -1 (no row highlighted) ... row count of the table

Events

The following events can be configured for the table object:
- OnOpenForInput
- OnEnabledWithFocus
- OnEnabledWithoutFocus
- OnValueChange
- OnEnterValue
- OnEnterNewValue
- OnEscape

See [here](#) ⬚430 for the descriptions of all event and possible actions that can be assigned.

6.5 Video Properties

The video output of cameras connected to the device can be displayed on the screen.
Whether video input is supported depends on the used device.
The following differences exist on the different hardware platforms:

Devices with i.MX35 CPU:
- 0, 1 or 3 analog video inputs (depending on the device)
- Only one video can be shown at the same time

- Analog video is always drawn as the most-backward graphics layer and the field of the video objects will be made invisible so that the video image gets visible.
Any other objects (e.g. other non-video frames on top of a video frame/page) will be displayed in front of the video image

Devices with i.MX6 CPU:
- 1, 2 or 3 analog video inputs (depending on the device)
- Two cameras can be shown at the same time (Please note: Due to hardware limitations, input 1 and 3 can not be displayed at the same time, only 1 and 2 or 2 and 3)
- It's possible to show a video stream over Ethernet with either H.264 or MJPEG. See below⁵¹¹ for details and limitations
- Video is drawn as the background texture of the video frame or page.
- For i.MX6 devices with a quad core processor it is possible to show two analog camera inputs plus an Ethernet camera at the same time.

Video Properties

Use this property to enable showing video in this object (only analog video will be mirrored):
- **None**
Disable video for this object
- **Unmirrored**
Enable video, no mirror

- **Horizontally Mirrored**
Enable video, mirror at the horizontal axis (up will be down)
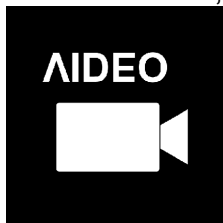
- **Vertically Mirrored**
Enable video, mirror at the vertical axis (left will be right)
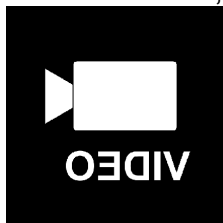
- **Both**
Enable video, mirror it horizontally and vertically (only analog video will be mirrored)

Note that the camera preview image in the tool changes with the different mirror settings.

Video Type

Here the type of the video can be chosen between analog video input or digital H.264 stream over Ethernet:

- **Analog Video Input**
  One of the device's analog video inputs shall be used (select which with Video Camera property below)
- **Ethernet Stream**
  Show a video stream received via Ethernet (see below for requirements and limitations)

Stream Type

Here the type of the Ethernet camera stream can be chosen:

- **H.264 via UDP/RTP**
- **MJPEG via UDP/RTP**
- **H.264 via RTSP**

UDP Port

If the [Video Type]510 was set to Ethernet stream and a Stream Type with UDP protocol was selected, the UDP port used by the camera can be configured here.

RTSP URL

If the [Video Type]510 was set to Ethernet stream and a Stream Type with RTSP protocol was selected, the RTSP URL can be written here.
The typical URL for Axis cameras is: rtsp://user:password@<IP>/axis-media/media.amp (e.g. rtsp://root:root@192.168.135.90/axis-media/media.amp)
Anonymous viewing can be enabled in the camera settings. Then user:pass@ can be omitted.
Please check the camera documentation for correct url.

Video Rotation

The camera image can be displayed rotated in 90° steps. This is useful if the camera is e.g. mounted sideways. Please note that an automatic rotation is added when running the device in portrait orientation.
Note the camera preview image changes with different rotation settings.

Video Camera

This property only has an effect if [Video Type]510 is set to Analog Video Input. Select one of the 3 video inputs of the device.

Video Alignment

This property only has an effect if Video Type is set to Analog Video Input. Generally the device tries to fill the frame or page completely with the video image. Since we maintain the aspect ratio of 4:3 with analog cameras, it is not always possible to fill the whole object

(especially a page on a 16:10 display). Use this property to position the video image either centered or in one of the corners. If the video is displayed in a frame, keep the frame size in a 4:3 format to avoid blank spaces.
Note that H264 video always fills the complete object. If the stream is in a 16:9 format, give the frame the same format to avoid distortions.

Scaling Factor

This property currently has no effect. Analog video will always be scaled so that the whole image is shown in a 4:3 format. H264 video will always be scaled to fill the whole object (adjust object size to stream format, to avoid distortions).

To further customize the look of the video image, the camera variables[174] can be used. Note that these only have an effect on the analog camera inputs.
The properties above are all applied as soon as the application switches to the page that contains the video frame. Other settings that were made with the camera variables[174] are overwritten.

Please note that the size and position variables shall not be set to higher values than the according device resolution, e.g. for a device with resolution 800 x 480, the x position and the width should not be set to a higher value than 800, and y position and height should not be set to a higher value than 480.

Ethernet camera requirements and limitations

General:

- The properties of the camera itself like resolution, IP, etc. can currently not be changed by the device. A tool to change camera settings via ISO 17215 is available in the program. See here[790] .
- Rotation is supported.
- The higher the resolution and/or bandwith of the camera is, the worse the performance on the screen will be. A rough recommendation is to keep it under 15 Mbit/s.
- It is possible to show the stream together with analog video on the same page. It is however recommended to do this only on devices that have multi-core CPU. On single core CPUs the performance might drop a lot.
- Only one stream can be shown per page.
- Ethernet needs to be activated on the device for the stream to be shown, see @Ethernet0_applynstart[174] and @Ethernet0_UpAtStart[174]
- @VideoStatus needs to be set to 1

H.264 via UDP/RTP / MJPEG via UDP/RTP:

- H.264 / MJPEG video streams over Ethernet are only supported on devices with i.MX6 CPU.
- The camera needs to use the RTP protocol via UDP/RTP to deliver the stream.
- The camera can either send the stream directly to the IP address of the device or via UDP broadcasts.
- The UDP Port that the camera uses needs to be set in the "UDP Port" property of the frame or page that shall show the stream.

H.264 via RTSP:

- RTSP video streams over Ethernet are only supported on devices with i.MX6 CPU.
- The camera needs to use the RTSP protocol to deliver the stream.

The program has a tool included with which Ethernet cameras can be configured. It can be accessed via the menu Tools -> See here[790] .

Download this project to see a basic Ethernet camera configuration or check out the welcome project[669] sample[130] .

JavaScript

7   JavaScript

Please Note the following changes in May 2016:

The complete JavaScript engine was exchanged!

Due to some problems experienced with the old Google v8 JavaScript engine, the engine was replaced with QtScript. The new script engine solves all issues currently known to us and has some more advanteges (like a debugger that can be used with the Windows simulation, see here[514]).

One importand difference to the old engine is that global variables are handled differently which might make changes in existing script neccessary:

In the old engine a variable that was declared as "var myGlobalVariable = 0;" in script A could be used in script B after script A was run once. **THIS WILL NOT WORK ANYMORE! myGlobalVariable will not be accessible in script B with the new engine.**
If it is really neccessary to use global variables, you can declare them as "myGlobalVariable = 0;" in script A and then use it in script B after script A was run once. **Note the missing "var" keyword!**
The same goes for functions. You need to declare functions without the keyword "function" to use them in other scripts.

We anyway do not recommend to use global JavaScript variables. It is far more reliable and secure to store important variable values that are needed in more than one script in a dedicated PClient variable (use setVariableValue[530] and getVariableValue[530] functions).

Other than that, the new JavaScript engine should not make any more changes in existing scripts neccessary.

JavaScript can be used to introduce highly-dynamic aspects into the project. It serves as a powerful, yet easy to use scripting language that enhances the possibilities of the tool. Seamlessly integrating into the architecture, it offers access to all the events and variables that can occur.

Do not use Java Scripts for extensive calculations! Also a frequent call of scripts is bad idea. This will slow down the complete terminal; instead the scripts should be small and fast.

If your desired functionality can not be realized with a small script or with other methods provided, please contact the support team. Most often there are other ways to get the result you want.

Debugging

Since May 2016, the new JavaScript engine provides a way to debug JavaScripts with the Windows Simulation. Here[596] you can configure if the debugger shall be off, on (open on exception) or on (open on start).

The complete documentation of the used debugger can be found here: QtScriptDebugger Manual. There you can learn how the debugger can be used and e.g. how breakpoints can be set.

Some quick notes:

- Scripts are only visible in the debugger if they ran once.
- A breakpoint can be set before a script was run (if you want a breakpoint in line 42 of myScript.js simply type ".break myScript.js:42" into the console of the debugger).
- The debugger can be triggered from within a script by using the statement *debugger;*. Execution will stop at that line.

The only way to debug JavaScripts on the device is to check the outputs on the serial console. Debug outputs can be added with the [print](#)[529] function.

7.1 The JavaScript Editor

Tutorial video about the JavaScript editor
https://www.youtube.com/watch?v=5LTKEISIeBY

Since the version 1.12.X of the program, a new JavaScript editor is integrated and provides many new features.

Create a new script file

JavaScript files can be created by right clicking on the **JavaScripts** node (1) insides the project tree and clicking **New File...** (2). Already existing scripts can also be used with **Import File...** (3). This functions copies the selected script file into the "terminal_files\scripts" folder of the corresponding project. Scripts from other projects can copied to the target project by using **Paste** (4)
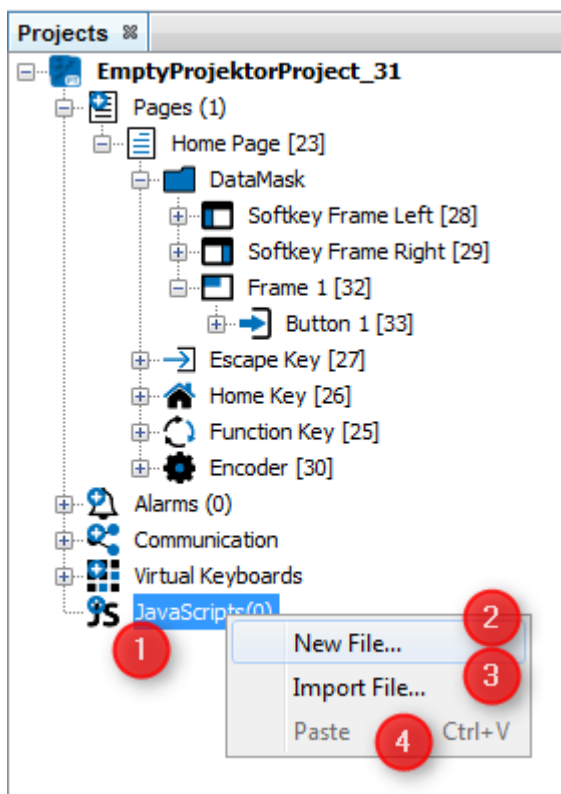
A wizard for creating a new file will be appear. In the category **JavaScript** an empty JavaScript file can be created. The category JavaScript Samples includes different templates. These templates can help to understand the basics of the JavaScript programming language or show best practices for different use cases.
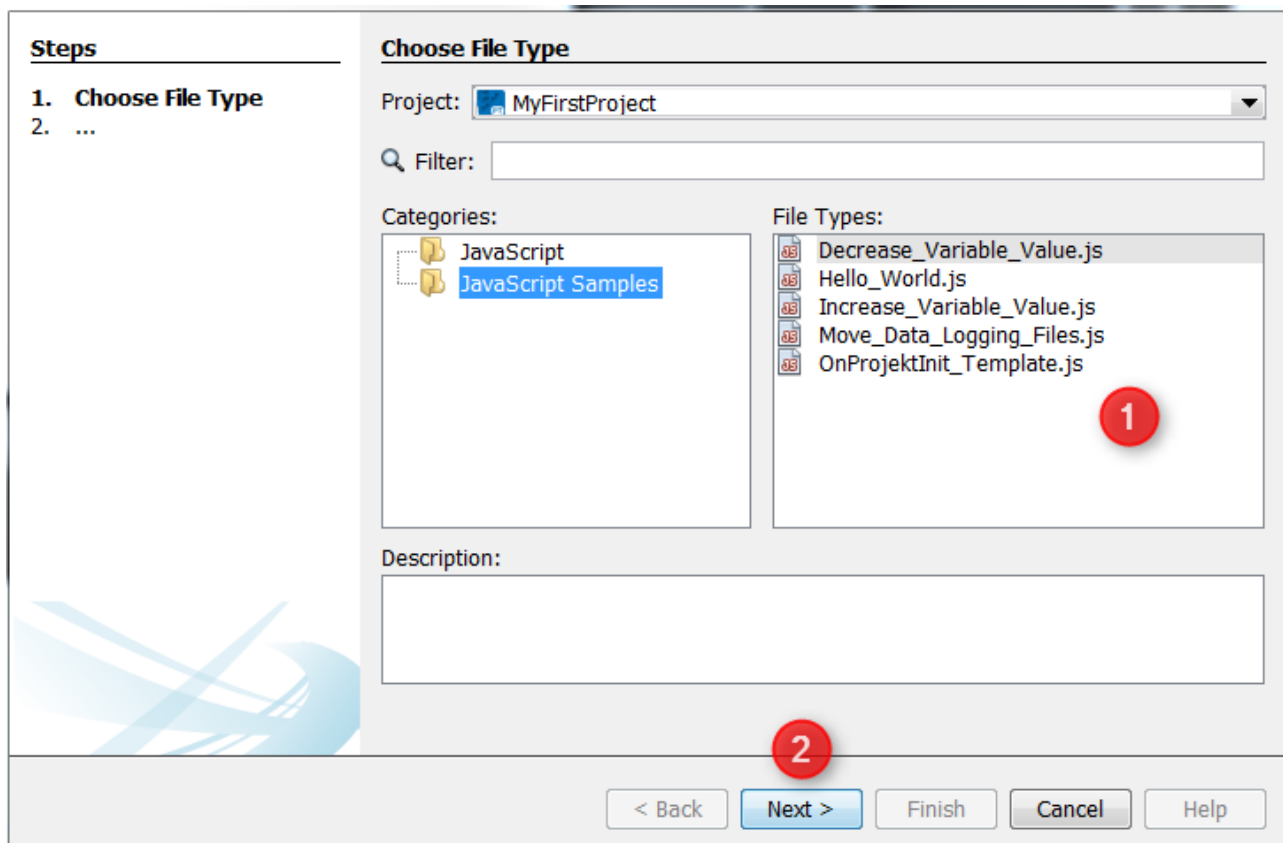
On the next page the wizard shows some information about the location of the new file. The folder "terminal_files\scripts" shouldn't be changed. Only scripts in this folder will be displayed in the JavaScipts node inside the project tree. The absolute path of the file is shown in the **Created File** text field (1). The name of the JavaScript file can be changed in the File Name section.
After clicking **Finish** (2), the new file will be created.
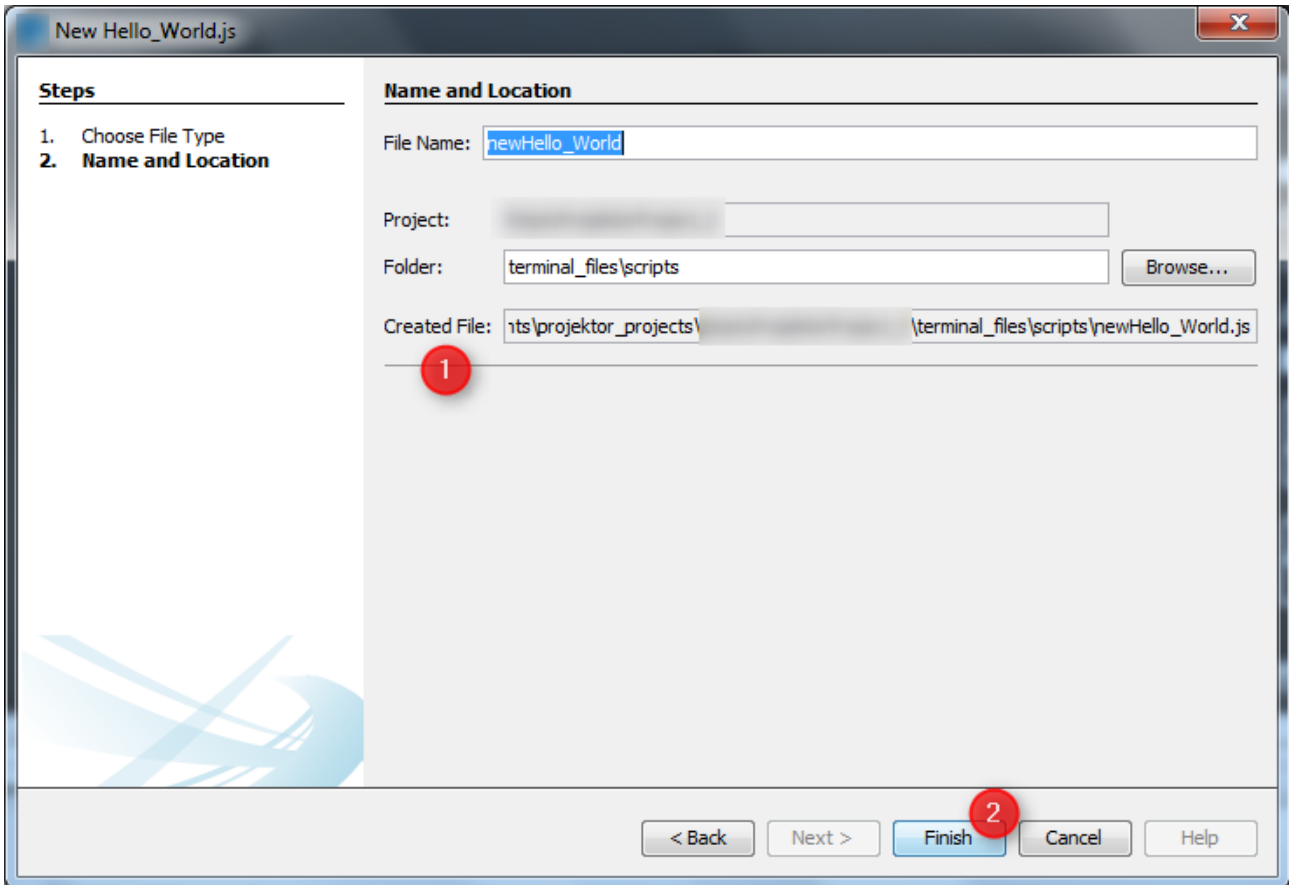
Editing a script

A script file can be edited by double clicking the corresponding file (1) insides the JavaScripts node. A new JavaScript editor tab (2) will be opened. In our example we have created an "Hello World" application with an template. The script file includes a function call "helloWorld()" and the definition of the function which prints out a String.

To edit a script just write your source code in the content area (2). Syntax highlighting is supported. A summary of all functions will displayed in the **Navigator View** (3).
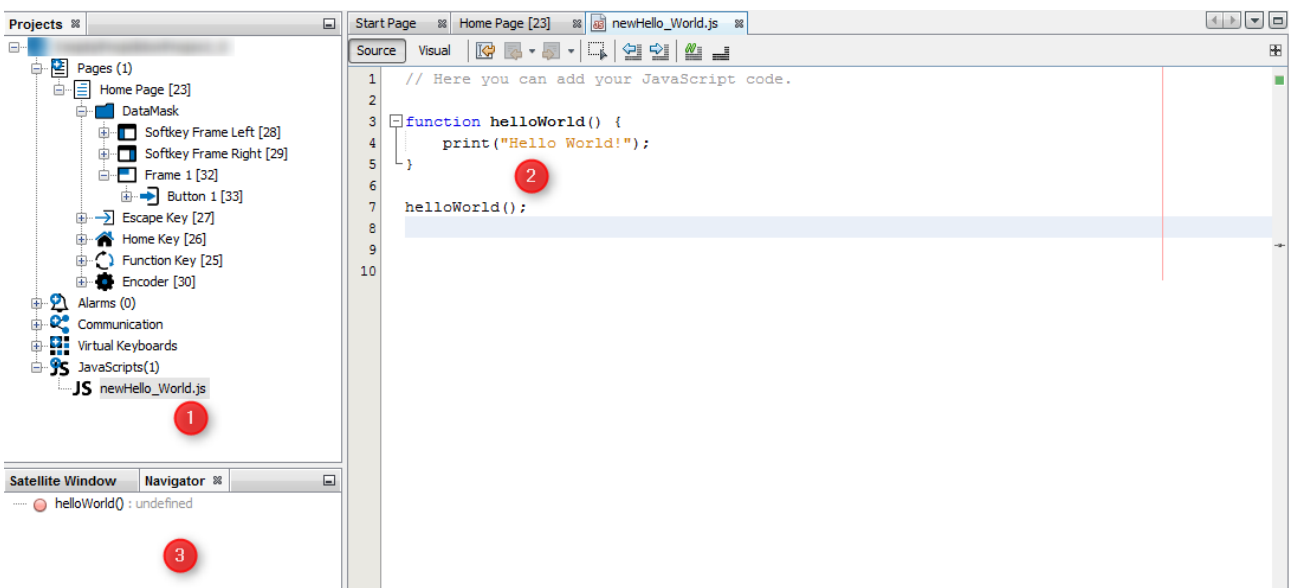
To help writing a script with variable interactions, the **Variable View** can be used to get / set the value of all variables that exist in the current project. They are sorted by variable group. A variable can by dragged into the context area by DnD (Drag & Drop).
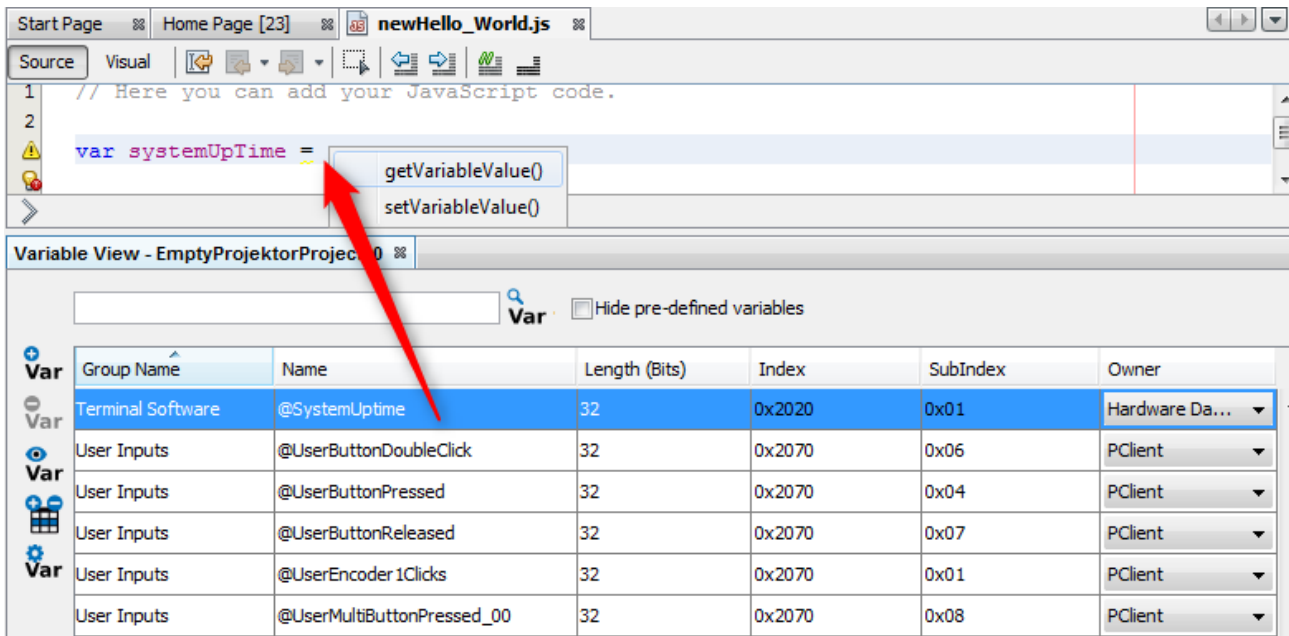
Also objects can be dragged into the context area by DnD (Drag & Drop).
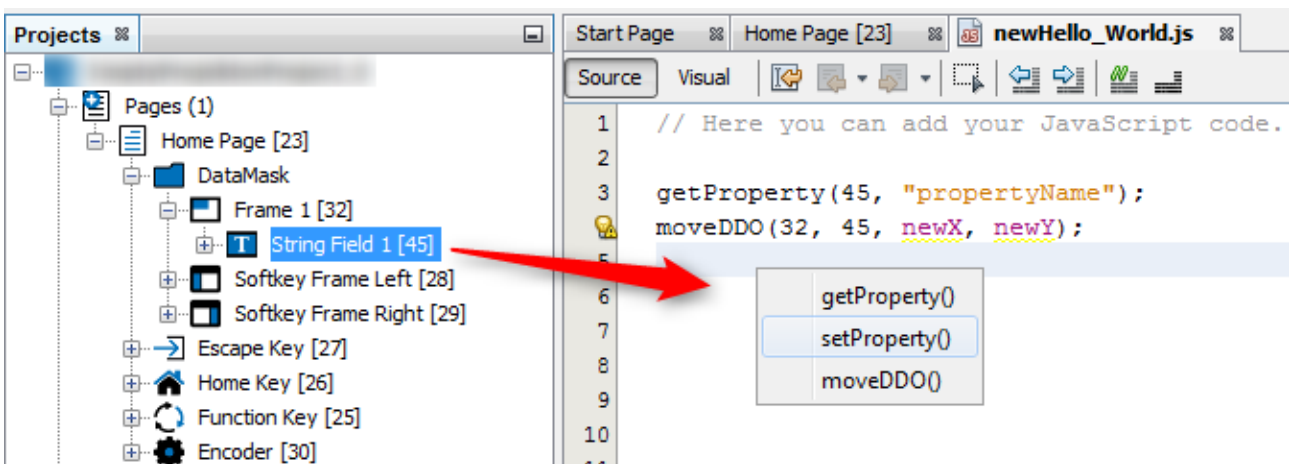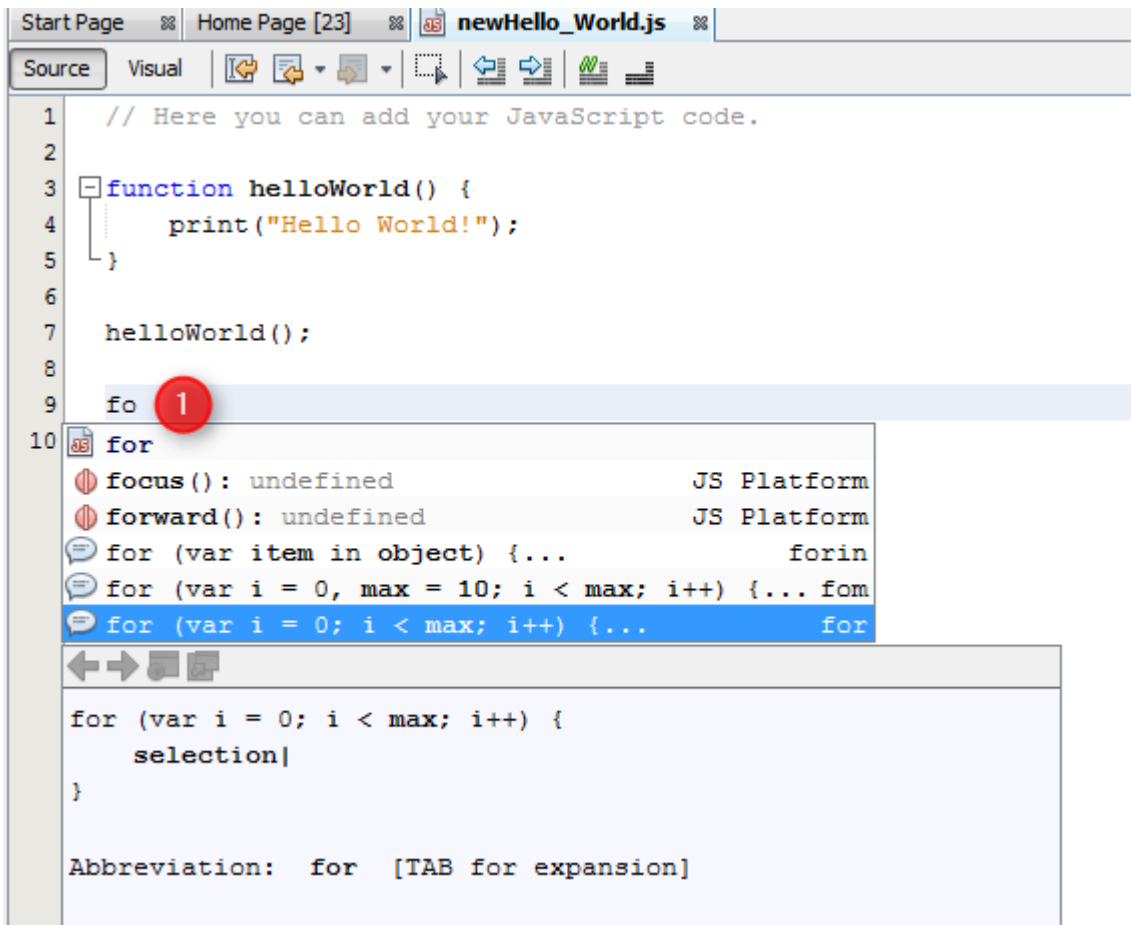
The current code line and

Auto Completion

Auto Completion is a feature in which an application predicts the rest of a word a developer is typing. Developers can easily press CTRL+SPACE to see one or more suggestion(s) and can use the arrow keys or mouse to accept one of several. Autocomplete speeds up human-computer interactions enormous.

The new powerful auto completion feature supports the whole JavaScript programming language syntax.

For example, we have only typed "fo" in the content area and press CTRL+SPACE. A context menu pops up with the keyword completion "for" and additional suggestions for whole "for loop" templates.

Note that not all expressions offered in the CTRL + Space popup are supported by our devices.

Auto Format

Another nice feature is the possibility to **Format** (1) the JavaScript code in the content area by default rules to get a pretty human readable code.
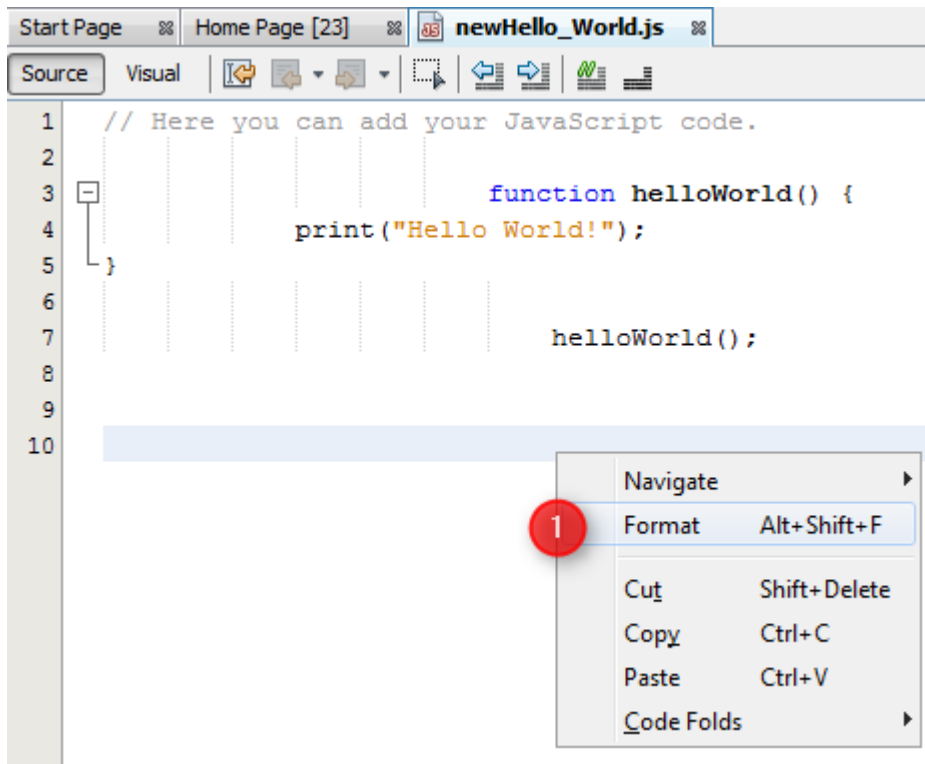
Error and Warning Hints

A further important feature highlight is the just in time detection of errors and warnings. The script below shows an error by the assignment of an value to the array position (1) caused by a missing closing bracket. For some errors the JavaScript editor suggest solutions to solve the problem by clicking on the lightbulb. (2) Shows an warning caused through an value compare with the equality operator instead the identity operator. The hint will shown by mouse over on the warning triangle.
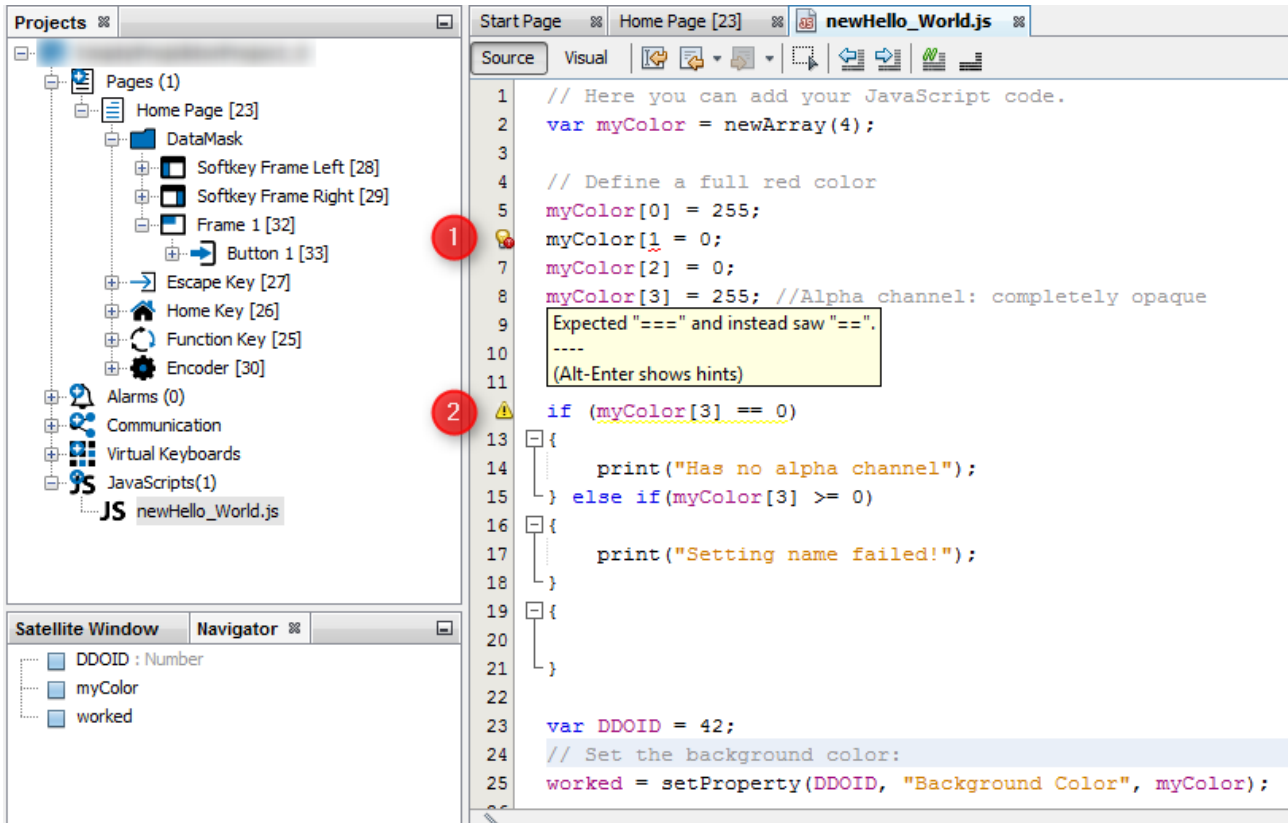
Shortcuts and icons

The folloging shortcuts can be used in the JavaScript editor:

CTRL + E	Delete the current line
CTRL + SHIFT + UP	Create a copy of the current line one line above
CTRL + SHIFT + DOWN	Create a copy of the current line one line below
ALT + SHIFT + UP	Move the current line one line up
ALT + SHIFT + DOWN	Move the current line one line down
ALT + SHIFT + LEFT	Move the current line one tab to the left
ALT + SHIFT + RIGHT	Move the current line one tab to the right
ALT + UP	Moves to the previous occurence of the marked expression (an expression has to be marked)
ALT + DOWN	Moves to the next occurence of the marked expression (an expression has to be marked)
CTRL + SHIFT + D	Opens a history of the last 9 copied text elements for pasting
CTRL + SHIFT + Click (left mouse key)	Adds an additional text cursor to the clicked position (multi cursor editing)
CTRL + SHIFT + L	Pastes multi-line copied text to one line per multi-cursor.
CTRL + SHIFT + R	Toggle rectangular selection
CTRL + SHIFT + C	Comment / Uncomment the current line
CTRL + Q	Move cursor to the last editing positions

CTRL + SHIFT + F	Opens a dialog with which all opened JavaScript files can be searched for a certain text phrase.
CTRL + SHIFT + H	Opens a dialog with which all opened JavaScript files can be searched for a certain text phrase to replace it with another.
ALT + SHIFT + F	Auto-format the currently selected code

The following icons are available in the JavaScript editor toolbar:

Source	Switch to the source code view
Visual	Switch to the Visual Programming view
	Move cursor to the last editing positions
	Toggle rectangular selection
	Move the current line one tab to the left / right
	Comment / uncomment the current line
	Save the current JavaScript file. Note that the project still needs to be saved before a download to the device!

Copy / Cut and Paste of script files

Already existing JavaScript files can be Copy / Paste into the same project or other projects.
Also Rename and Delete of the files is supported and updates all references on events.
Hint: Do not rename the script file over the file system when the file is referenced to object events, because then the references will be broken.
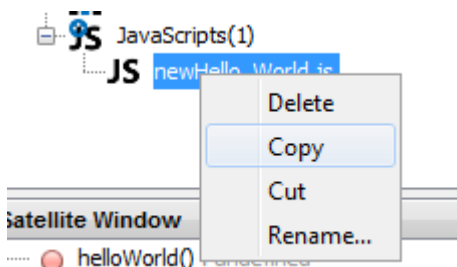
Handling of non ANSI characters

We have included a plugin that allows JavaScripts with non-ANSI characters (UTF-8 format).
By default, all script files will be opened and interpreted as UTF-8.
When a script file is opened in the editor, the encoding format will be shown in the lower right corner of the program.
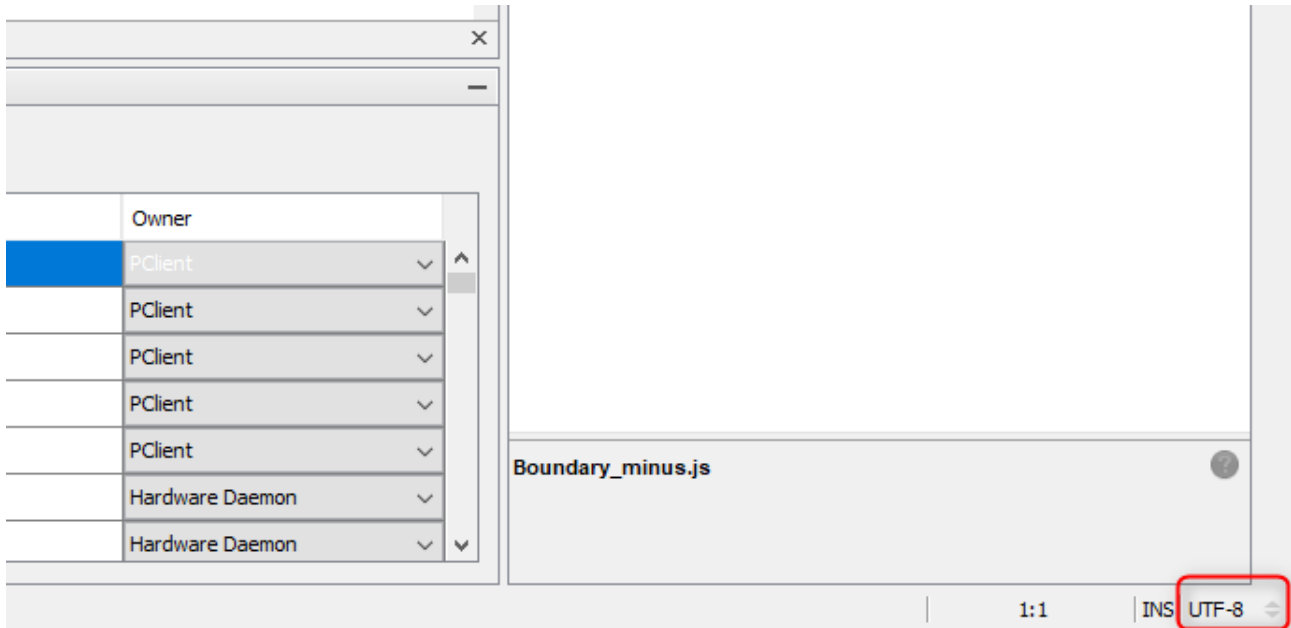
The encoding can also be changed by pressing this place. Normally there is no reason to use any other encoding than UTF-8.

Additionally it is possible to load and save a file in a certain format. There are entries in the menu File called Open in Encoding... and Save in Encoding... where a format can be chosen in which the selected file will be interpreted.

7.2   Assign a JavaScript file to an Event

Java Scripts can be configured to be executed at certain events. You can easily drag an existing JavaScript file to an object on scene and can select the desired event.
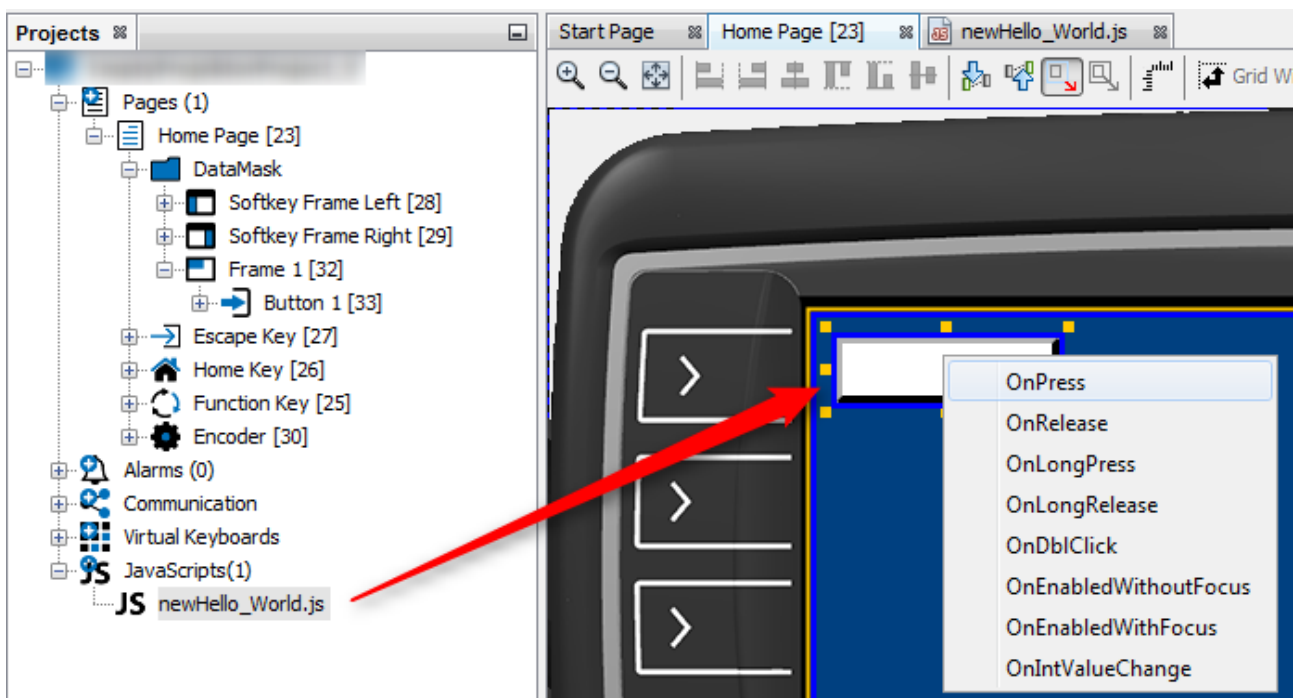
The assigned event will be displayed at the events tab of the Object (1). You can configure such an **Execute Script Action** by clicking on the button (2). Here you can choose an existing **Script File** (3) for execution or create a **New Script File** (4). Already existing JavaScripts can also be used in the project by using the function **Import Script File** (4). Files can be modified by clicking **Edit selected Script File** (5). After selecting the script file it is possible to configure how often the script shall be executed using the **Execute on every Xth Event** field.
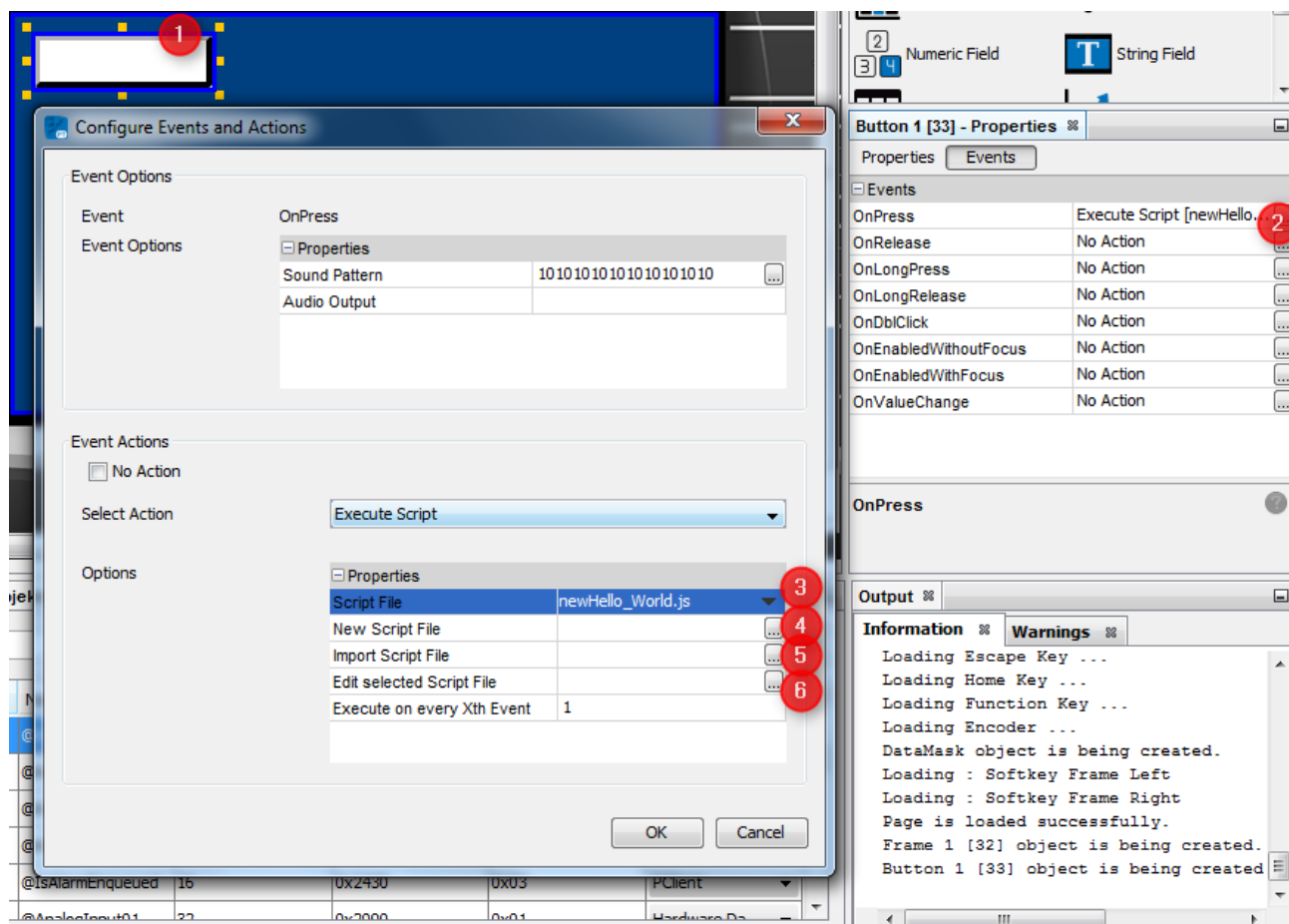
If the script shall be executed for the **OnPageRepeat** or **OnProjectRepeat** events, the **Period (100 ms)** property also needs to be set in the event options. This property defines how often the script will be called.

Please note that the execution time for repeat actions is a multiple of 100ms. We can not guarantee that we will always be exact in executing the repeat events. The scripts should not depend on an exact execution time.

7.3 JavaScript Basics

The devices can execute any standard JavaScript code that is not related to HTML or browser functions (like document object, location object, etc.). It is possible to use loops (do/while, while, for), conditions (if, switch) and standard JS objects (Array, Date, Math, Number, String). The syntax is very easy to learn and use. A good starting point for learning JavaScript is https://www.w3schools.com/js/

Comments

Like many other programming languages, comments start with a double slash "//"

```
//This is a comment
```

Comments can also be spread over several lines:

```
/*
This
is
a
multiline
comment*/
```

Note that every command in a JavaScript should end with a semicolon ";".

JavaScript Variables

JavaScript is a weakly typed scripting language which means that no type declaration is needed when creating variables and differently typed variables can be assigned to each other.

```
//To create a variable use the keywork "var":
//Creates an integer variable with value 1
var numberVariable = 1;
//Creates a string variable with value "myString"
var stringVariable = "myString";
// Creates an array with the size of 4
var myArray = new Array(4);
```

Please note that the initialization "var variablename" leads to the deletion of the variable after the script was processed. For JavaScript variables to be used globally, they need to be initialized without the keyword var. It is preferred to use PClient variables to store important data. JavaScript variables should not be used for that.

Operators

The following operators can be used for coding:

Operator	Example	Result	Comment
+	12 + 30 "Java" + "Script"	42 "JavaScript"	
-	69 - 27	42	
*	16 * 3	42	
/	126 / 3	42	
%	3 % 2 4 % 2	1 0	Modulo operator
++	i = 41; i++, ++i	42	increment
--	i = 43; i--, --i	42	decrement
== == ==	3 == 2 42 == "42" 42 == 43	false true true	weak compare

Operator	Example	Result	Comment
=== ===	42 === 42 42 === "42"	true false	strong compare
<	42 < 43 "w" < "W"	true false	less than
<=	42 <= 42 42 <= 43	true true	less or equal
>	42 > 43	false	greater than
>=	42 >= 43	false	greater or equal
=	i = 42		Value 42 is assigned to variable i
+=	i += 1	43	i = i + 1;
-=	i -= 1	42	i = i - 1;
*=	i *= 2	84	i = i * 2;
/=	i /= 2	42	i = i / 2;
%=	i %= 2	0	i = i % 2;
	i = 2; j = 5;		
&&	(i <= 2) && (j < 7)	true	both statements need to be true (AND)
||	(i%2 == 0) || (j%2 == 0)	true	only one statement need to be true (OR)
!	(i ==2) && ! (j%2 == 0)	true	negation of a statement (NOT)
	i = 2; j = 7;		
&	i & j	2	Check if bit is set. In binary: 0010 & 0111 = 0010
|	i | j	7	Set a bit. In binary: 0010 | 0111 = 0111
^	i ^ j	5	Toggle a bit. In binary: 0010 & 0111 = 0101
<<	2 << 1	4	Shift bits to the left. In binary: 0010 << 1 = 0100
>>	2 >> 1	1	Shift bits to the right. In binary: 0010 >> 1 = 0001
>>>	i = 10; i >>> 2	2	Shift right and fill with zeros. In binary: 1010 >>> 2 = 0010

Loops

Loops can be used in JavaScript similar to many other programming languages.

WARNING: A loop stops execution of the main application until it is finished. So make sure that only very short loops are used.

WARNING: Loops should and can NOT be used to "wait" within a JavaScript program. You can NOT "spend time" within one JavaScript. If you need to time something, use the repeated

execution of scripts, so instead of a for loop, just write x := x + 1; and after 10 repetitions the appropriate code can be executed.

Three different loop types are available:
- The **while loop**. It only begins if the condition is true and stops as soon as the condition is not true anymore:

```
//Example: loop 10 times:
var i = 0;
while (i < 10)
{
    i = i + 1; //same as i++;
}
```

- The do **while loop**. This kind of loop will execute the code inside the loop at least once. Only after this the break condition is evaluated:

```
var k = 11;
do
{
    k = k + 1;
} while (k < 10);
//now k is == 12
```

- The **for loop**. This loop is used to run the code inside for specified number of times. The for loop statement includes 3 statements. The first is normally used to initialize a (counter) variable. The second statement is the break condition (when the loop will end) and the third is a statement that will be executed after each loop (normally to increment the counter):

```
for (var x = 0; x < 10; x++)
{
    //do something
    //x will have the values 0..9
}
```

To exit a loop inside a loop, the command **break;** can be used. To skip the current loop iteration and directly start with the next iteration, the command **continue;** can be used.

Conditions

Two condition statements exist in JavaScript that allow code execution if one or more conditions are true.
- The **if condition** statement. If the condition inside the "if" clause is true, the line behind (or the code block inside the brackets) gets executed. To execute special code if the condition is not true, an "else" clause can be added behind the code of the "if" clause. Both can be combined using else if.

```
var i = 42;
if (i < 42)
{
    //execute this code if i is less than 42
}
else if (i === 42)
{
    //execute this code if i is equal to 42
}
else
```

```
{
    //execute this code if none of the above conditions were
true (in this case i is greater than 42)
}
```

- The switch statement. The switch statement makes it easy to check the value of exactly one variable without the need to write complex if-else statements:

```
var i = 42;
switch(i)
{
    case 1:
        //do something if i is equal to 1;
    break;
    case 2:
        //do something if i is equal to 2;
    break;
    case 42:
        //do something if i is equal to 42;
    break;
    case 48:
    case 50:
        //do something if i is equal to 48 or 50;
    break;
    default:
        //do something if none of the above cases was true
    break;
}
```

Note the **break;** statement in every case. If this is missing, the next case will also be executed even if the case condition is not true (for values 48 and 50 the same code will be executed). It is good practice to have a **default case** at the end of the switch. This gets executed if no other case matched.

Data types

JavaScript also provides the **typeof** operator. With this the data type of a variable can be checked:

```
var myNumber = 42;
var myString = "Hello World";
print("My number is of type " + (typeof myNumber) + ", my
string has type " + (typeof myString));
```

The return values of typeof are:
- boolean
- string
- number
- function
- object
- undefined

7.4   Custom JavaScript functions for device specific functionality

Various custom functions exist that help JavaScript interact with project variables and objects. By pressing CTRL+SPACE in the JavaScript editor content area (1), a list with auto complete suggestions will appear. On the top of the list are functions from the category **Custom Functions** (2). A Tooltip (3) for the selected function shows a short description and gives informations about the parameter(s) / return types.
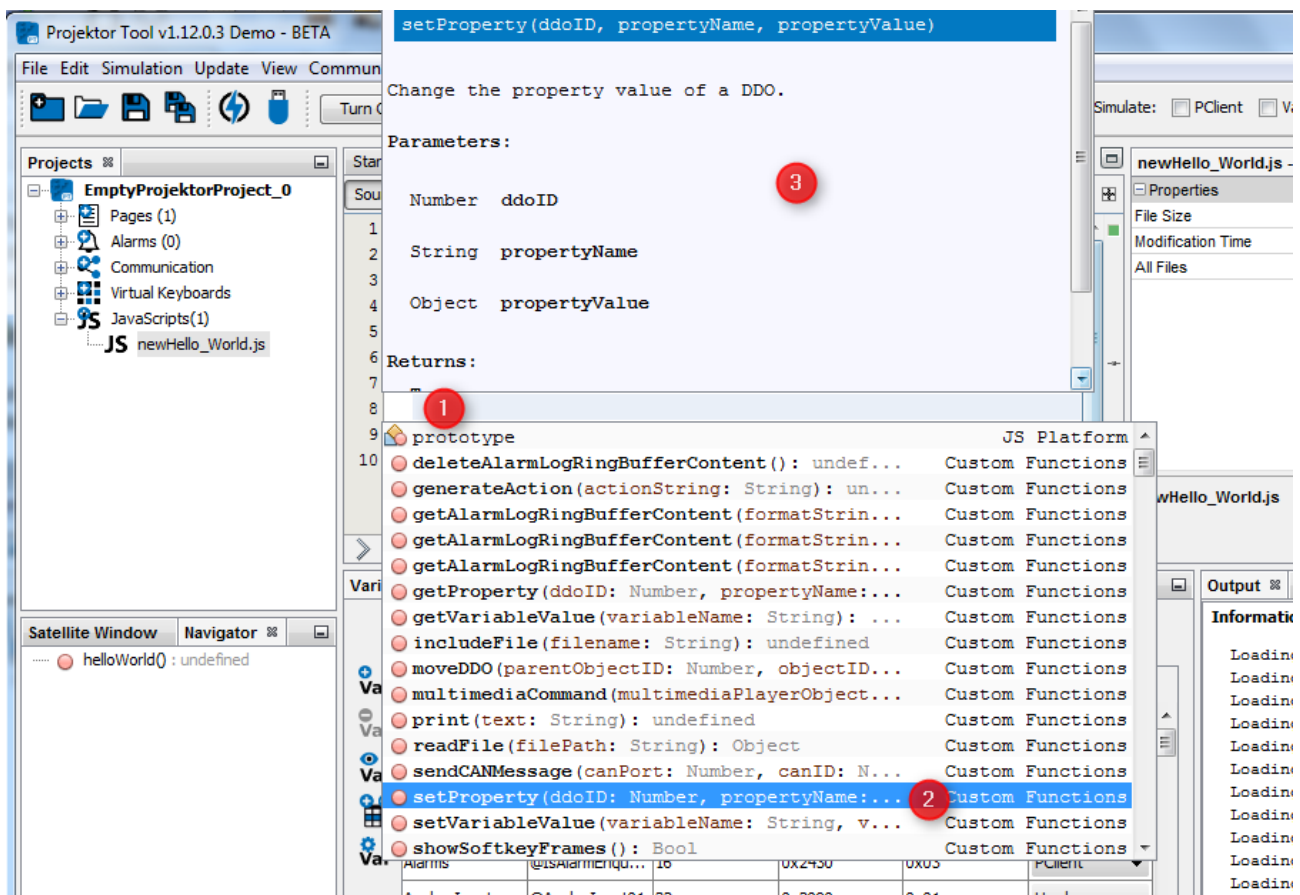
print - print debug output to console

The print function allows to print debug messages to the serial console. This can be very helpful in debugging JavaScripts during development.
Note that when a lot of debug messages are used the system performance can be slightly decreased. When releasing a project the print statements should be commented.

Prototype:
```
print (string stringToPrint);
```

Examples:
```
var result = 21 * 2;

// output: String to print
print("String to print");

// output: Result was 42
print("Result was " + result);
```

```
// output: Strings and numbers can be concatenated!
print("Strings " + "and " + "numbers " + "can " + "be " +
"concatenated!");
```

Please note that the print function also works with the PClient simulation on the PC. The output will be written in the file
%USER%\AppData\Roaming\%program version%
\extracted_components\pclient\simulation\console_output.txt

getVariableValue - retrieve the value of a project variable

In a project a lot of variables are used to store runtime data and you can configure your own variables in a project. To retrieve the value of such a variable the getVariableValue function can be used.

Prototype:
```
var result = getVariableValue(string nameOfVariable);
```

The returned value of that function contains the value of the variable. This can be either a string or a number depending on the variable type.

Examples:
```
var backlightValue =
getVariableValue("@DisplayBacklightIntensity");
// result is a number representing the backlight intensity
print("Value of backlight is: " + backlightValue);

var terminalName = getVariableValue("@Terminal");
// result is a string representing the name of the device this
is executed
print("I am running on a " + terminalName);
```

setVariableValue - change the value of a project variable

To set a value to a pre-defined or self-created variable you can use the setVariableValue function. The return value of the function will be **true** (setting value worked) or **false** (setting value failed).

Prototype:
```
var result = setVariableValue(string nameOfVariable, number
newValueForVariable);
//or
var result = setVariableValue(string nameOfVariable, string
newValueForVariable);
```

Example:
```
var worked = setVariableValue("@DisplayBacklightIntensity",
50);
if (worked === true)
{
    print("Setting display backlight to 50% worked!");
}
else
{
```

```
    print("Failed to set the display backlight intensity to 50%
:(");
}
```

getVariableValid - retrieve information if a project variable has a valid value

With this function you can check if a variable has a valid value. Returns true if the variable has a valid value. Returns false if the variable doesn't have a defined value. Returns "undefined" if the variable doesn't exist or if the function was not called with one parameter only.

Prototype:
```
var result = getVariableValid(string nameOfVariable);
```

The returned value of that function contains the information about the variable.

Examples:
```
var result = getVariableValid("Speed");
if (result === true)
    print("The variable has a valid value.");
else if (result === false)
    print("The variable does NOT have a valid value.");
else
    print("The variable does not exist or the function was
called inappropriately.");
```

getProperty - retrieve a property value of a DDO

All graphical objects (DDO => Display Data Object) in a project have a lot of properties. In some cases these properties can be of interest during runtime in a JavaScript. To retrieve the value of such a property the function getProperty can be used.

Prototype:

```
var value = getProperty(number DDO-ID, string propertyName);
```

The return value of the **getProperty** function will be either the value of the property or **undefined** if something did not work. Since the properties of a DDO consist of various data types, the returned property value can also be a different data type depending on the property. Here is a list of all the properties that can be read and/or changed through JavaScript.

Javascript propertyName	Data type	Get	Set	Object type(s)	Remarks
Absolute Max Value	Integer	x	x	Numeric Fields Bargraphs Meter object	
Absolute Min Value	Integer	x	x	Numeric Fields Bargraphs Meter object	
Add To Page History	Boolean	x	x	Pages	Pages where this is set to false will not be added to the browsing history (they will not be accessable by pressing the escape key)
Alarm Number	Integer	x		Alarm objects	

Javascript propertyName	Data type	Get	Set	Object type(s)	Remarks
Alignment	Integer	x	x	String Fields Numeric Fields	0 - left aligned 1 - center aligned 2 - right aligned
Arc Color	Color	x	x	Bargraphs	Array object containing 4 values (RGBAlpha)
Auto Wrap	Boolean	x	x	String Fields	Text wraps and new lines are created automatically if set to true
Background Color	Color	x	x	All objects except container	Array object containing 4 values (RGBA)
Background Image	String	x	x	All objects except container and virtual keyboards	absolute path to the image on the device
Behaviour	Integer	x		Alarm objects	0 - Emergency 1 - Warning 2 - Informational
Bit Mask	Integer	x	x	Bargraphs Meter objects Numeric Fields	
Border Image	String	x	x	All objects except container, virtual keyboards and pages	
Border Width	Integer	x	x	All objects except container, virtual keyboards and pages	
Color Above Maximum	Color	x	x	Numeric Fields Bargraphs Meter object	Array object containing 4 values (RGBA)
Color Below Minimum	Color	x	x	Numeric Fields Bargraphs Meter object	Array object containing 4 values (RGBA)
Connect All Softkey Frames	Boolean	x	x	Softkey Frames	Use the page ID as the object ID and set to TRUE or FALSE TRUE - softkey level changes happen on both sides equally FALSE - softkey level changes are independent on both sides
Current Softkey Level	Integer	x	x	Softkey Frames	
DecimalButtonID	Integer	x		Virtual keyboards	ID of the decimal button in the virtual keyboard
Deflection Direction	Integer	x	x	Meter objects Arched bargraphs	
Dependency	Integer	x		All objects except container, virtual	

Javascript propertyName	Data type	Get	Set	Object type(s)	Remarks
				keyboards and pages	
Display Video	Integer	x	x	Frames	The number corresponds to the video input
Display Zeros As Blank	Boolean	x	x	Numeric Fields	
DoneButtonID	Integer	x		Virtual keyboards	ID of the Done button in the virtual keyboard
Draw Arc	Boolean	x	x	Meter objects	
Draw Border	Boolean	x	x	All objects except container, virtual keyboards and pages	
Draw Target Line	Boolean	x	x	Bargraphs	
Draw Ticks	Boolean	x	x	Bargraphs Meter objects	
Encoder Movement	Integer	x	x	Bargraphs Meter objects Numeric Fields Tables List objects	0 - linear 1 - dynamic
End Angle	Integer	x	x	Arched bargraphs Meter objects	
Fadeout	Boolean	x	x	Softkey frames	
Fill Mode	Boolean	x	x	Bargraphs Arched Bargraphs	TRUE - bargraph is filled FALSE - bargraph is not filled (just a line at value position)
FirstSetBit	Integer	x	x	Bargraphs Meter objects Numeric Fields	First bit that is used for the bit mask (correlates with "Choose number of bits" in the tool GUI)
Flashing	Boolean	x	x	All objects except container, frames, virtual keyboards and pages	lets objects flash with a given frequency (see below)
Flashing Cycle	Integer	x	x	All objects except container, frames, virtual keyboards and pages	min 100 ms, max 5000 ms
Font	String		x	Numeric Fields String Fields Tables	
Font Attribute	Integer	x	x	Numeric Fields String Fields Tables	
Font Color	Color		x	Numeric Fields String Fields Tables	Hexadecimal value 0xRRGGBBAA (red, green, blue, alpha)

Javascript propertyName	Data type	Get	Set	Object type(s)	Remarks
Foreground Color	Color	x	x	Bargraphs Tables	Array object containing 4 values (RGBA). For table objects this will change the "Grid Color" property.
Format	Integer	x	x		
GIF Frame Number	Integer		x	Animated GIF images	
GIF Loops	Boolean	x	x	Animated GIF images	
GIF Speed	Integer	x	x	Animated GIF images	
GIF State	Integer	x	x	Animated GIF images	
Height	Integer	x		All objects except pages and virtual keyboards	
Highlighted Row	Integer	x	x	Table	
Horizontal Alignment	Integer	x	x	Numeric Fields String Fields	0 - left aligned 1 - center aligned 2 - right aligned
ID	Integer	x		All objects	
Image To Display	String	x	x	Picture graphic objects	absolute path to the image on the device
Item Height	Integer	x		List objects	
Items Visible (Normal Mode)	Integer	x	x	List objects	
Items Visible (Editing Mode)	Integer	x	x	List objects	will be cropped to the maximum number of items that can be displayed with next edit
Key Number	Integer	x		Virtual keyboards	
Key Pressed Bitmap	String	x		Virtual keyboards	
Leading Zeros	Boolean	x	x	Numeric Fields	
Level	Integer	x		Softkey frames	
List Highlighted Position	Integer	x		List objects	
Max String Length (Characters)	Integer	x	x	String Fields	
Max Value	Integer	x	x	Numeric Fields Bargraphs Meter object	
Min Display Time	Integer	x	x	Alarm objects	
Min Value	Integer	x	x	Numeric Fields Bargraphs Meter object	

Javascript propertyName	Data type	Get	Set	Object type(s)	Remarks
Name	String	x	x	All objects	
Needle Color	Color	x	x	Meter objects	Array object containing 4 values (RGBA)
Needle Image	String	x	x	Meter objects Gauge objects Bargraphs	absolute path to the image on the device
No Of Decimals	Integer	x	x	Numeric Fields	
Number Of List Items	Integer	x		List objects	
Number Of Ticks	Integer	x	x	Bargraphs Meter objects Gauge objects	
Object Status	Boolean	x	x	CAN Freestyle / J1939 mappings	If set to true, the mapping is active, if false, the mapping is not active.
Offset / Offset 2	Float	x	x	Numeric Fields	
Periods	Integer	x	x		
Pressed State	Boolean	x	x	Buttons Softkeys	
Preview Value	Integer / String	x	x	Bargraphs Gauge objects List objects Meter objects Numeric Fields String Fields	For String Fields, the combination \n will insert a line break
Roll Over	Boolean	x	x	All editable objects	For frames: When choosing objects to edit with the encoder, the focus jumps from the last to the first if this is set to true For other objects: When reaching the abs. maximum value of the object, the value jumps to the abs. minimum value of the object
RTSP URL	String	x	x	Frames with Ethernet camera input	See RTSP URL in Video Properties[510] . Note that video has to be stopped before switching to a new URL: setVariableValue("@VideoStatus", 0); setProperty(47, "RTSP URL", "rtsp://root:root@192.168.135.90/axis-media/media.amp"); setVariableValue("@VideoStatus", 1);
Scale	Double	x	x	Numeric Fields	

Javascript propertyName	Data type	Get	Set	Object type(s)	Remarks
Scaling Factor	Integer	x	x	Frames	Only applicable if Display Video is >0
Send Value	Boolean	x	x	Bargraphs List objects Meter objects Numeric Fields Tables	Equals "Send value directly"
Set As Input	Boolean	x	x	All editable objects	
Show Unused Levels	Boolean	x	x	Softkey frames	
Sound Pattern	Integer	x	x	Alarm objects	
Start Angle	Integer	x	x	Arched bargraphs Meter objects	
Step Width	Integer	x	x	Numeric Fields	
Stream Type	Integer	x	x	Frames with Ethernet camera input	0 - H.264 via UDP/RTP, 1 - MJPEG via UDP/RTP - 2 - H.264 via RTSP
Table Data	Array	x	x	Tables	String array with one element per line, semicolon (;) as a default delimiter per column (can be changed in table properties)
Target Line Color	Color	x	x	Bargraphs	Array object containing 4 values (RGBA)
Target Value	Integer	x	x	Bargraphs	
Tick Color	Color	x	x	Bargraphs	Array object containing 4 values (RGBA)
Timeout	Integer	x	x	Alarm objects	
Touch Input	Boolean	x	x	List objects	Paints a gray border around the list object when it's being edited if this is set to true
Transparent	Boolean	x	x	All objects except container and pages	TRUE - object background is not painted FALSE - object background is painted
Transparent Border Color	Color	x	x	All objects that have a border	Array object containing 4 values (RGBA)
Type	Integer	x		Alarm objects	0 - Page 1 - Frame
Type ID	Integer	x		Alarm objects	Alarm type ID
UDP Port	Integer	x	x	Frames with Ethernet camera input	UDP port for H.264 via UDP/RTP or MJPEG via UDP/RTP protocol. Note that video has to be stopped before switching to a new port:

Javascript propertyName	Data type	Get	Set	Object type(s)	Remarks

					setVariableValue("@VideoStatus", 0);
					setProperty(47, "UDP Port", 5005);
					setVariableValue("@VideoStatus", 1);

Unit Group	String	x		All objects except pages, list objects and virtual keyboards	
Value	Integer / String	x		Numeric Fields String Fields	Integer for numeric fields, string for string fields
Value Change Factor	Integer	x	x	Bargraphs Gauge objects List objects Meter objects Numeric Fields Tables	
Vertical Alignment	Integer	x	x	Numeric Fields String Fields	0 - top aligned 1 - center aligned 2 - bottom aligned
Video Alignment	Integer	x	x	Frames	Only applicable if Display Video is >0 0 - not mirrored 1 - mirrored horizontally 2 - mirrored vertically 3 - mirrored horizontally and vertically
Video Rotation	Integer	x	x	Frames	Only applicable if Display Video is >0 Can only be set to 0, 90, 180 or 270. This will change @CamXRotation and @CamXMirror accordingly.
Visibility	Boolean	x	x	All objects except pages and virtual keyboards	
Visibility Bit	Integer	x	x	All objects except pages and virtual keyboards	
Visibility Variable Index	Integer		x	All objects except containers and virtual keyboards	
Visibility Variable Subindex	Integer		x	All objects except containers and virtual keyboards	
Visible	Boolean	x	x	All objects except containers and	

Javascript propertyName	Data type	Get	Set	Object type(s)	Remarks
				virtual keyboards	
Width	Integer	x		All objects except pages and virtual keyboards	
X Location	Integer		(x)*	All objects except pages and virtual keyboards	Always relative to its direct parent *Use the function moveDDO⌐542
Y Location	Integer		(x)*	All objects except pages and virtual keyboards	Always relative to its direct parent *Use the function moveDDO⌐542

Examples for getProperty:

```
var DDOID = 42;
var myID = getProperty(DDOID, "ID");
print("ID of DDO no. " + DDOID + ": " + myID);

var myColor = new Array(4);
myColor = getProperty(DDOID, "Background Color");
// Now the array contains the Red, Green, Blue and Alpha
components
print("The background color is: R: " + myColor[0] + " G: " + my
Color[1] + " B: " + myColor[2] + " A: " + myColor[3]);

var imagePath = getProperty(DDOID, "Background Image");
print("The image path is: " imagePath);
```

setProperty - change the property value of a DDO

Setting properties is as easy as getting them. The setProperty function can be used for that.

Prototypes:

```
var result = setProperty(number DDO-ID, string propertyName,
integer value);
var result = setProperty(number DDO-ID, string propertyName,
double value);
var result = setProperty(number DDO-ID, string propertyName,
bool value);
var result = setProperty(number DDO-ID, string propertyName,
string value);
var result = setProperty(number DDO-ID, string propertyName,
Array array);
```

Examples:

```
var DDOID = 42;
var myColor = new Array(4);
// Define a full red color
myColor[0] = 255;
myColor[1] = 0;
myColor[2] = 0;
myColor[3] = 255;     //Alpha channel: completely opaque
```

```
// Setting a new name to a DDO:
var worked = setProperty(DDOID, "Name", "MyNewName");
if (worked === true)
{
    print("Setting new name worked!");
}
else
{
    print("Setting name failed!");
}

// Set the background color:
worked = setProperty(DDOID, "Background Color", myColor);
// Set a new background image
worked = setProperty(DDOID, "Background Image",
"/opt/images/bg.png");
```

readFile - read the contents of a file

The contents of a file on the file system or external media can be read and stored in a vector containing strings. Each vector entry represents one line of a file. If the reading was not successful, the result is undefined.

Important note: The current maximum line length that is read is 1024 characters. If the file contains a line that is longer, it is possible that the file can not be read correctly

Important note 2: The maximum number of bytes is currently restricted to 2MB. If the file is larger than 2MB, only the first 2MB of the file are read, the rest is discarded.

Prototype:
```
var result = readFile(string filePath);
```

Example:
```
//read the contents of the file /proc/cpuinfo
var myData = readFile("/proc/cpuinfo");

//check if successful:
if (undefined == myData)
{
    //reading failed!
    print("Error reading file /proc/cpuinfo");
}
else
{
    //reading successful, myData is now an array
    //print contents to console:
    for (var i = 0; i < myData.length; i++)
    {
        print("Line " + (i + 1) + ": " + myData[i]);
    }
}
```

To display the contents of the retreived vector the Table Object can be used:

```
setProperty(TableID, "Table Data", myData);
```

For details see the manual chapter for the table object.

writeToFile - write a string to a file on the file system

With the writeToFile function it is possible to write strings to a file.

For security reasons the file can only be located on a connected USB stick or on a dedicated folder of the device. If the **filePath** provided is not on the connected USB stick, the file is automatically created in the device's dedicated directory (on the device this is usually in "/opt/user_files/", on the windows simulation this directory is in the designer's install path (pclient/user_files/). Note that a file is automatically created if it does not exist yet (this will fail if the path contains directories that do not exist).
Alternatively, the user_files directory can be created using the [createUSBDirectory](#)[552] function.

If the file is located on a USB stick, the size of the file is not restricted (write will fail if there is no space left on the device). If the file is located on the file system of the device, the memory that is available in the /opt/user_files directory is limited to 100 MB (you can create many smaller files but not more than 100 MB in the directory). Writing will fail if this limit is reached.

From this version onwards, files can also be saved in the directory /opt/data. This folder will be backed up during an OS / application update, which is advantageous to keep data intact (only for i.MX6 devices).

The **stringToWrite** parameter may contain any string. No line breaks are added at the end of each write, these need to be added to the stringToWrite if desired (add "\n" for a line break). If the string only consists of one line, a line break has to be added at the end.

The **option** parameter can either be "append" or "replace". If the string is appended, the string will just be added at the end of the file. If the string is replaced, the whole content of the file is discarded before writing the new string to the file. If any other string is provided as option parameter the writeToFile function will fail.

The optional **sync** parameter can be set to **true** if the file shall be directly written to disk (reduce the risk of corrupt files at total power loss). The sync to disk will be done after the content was written to the file. If the **sync** parameter is set to **true** the writeToFile function will **not return** until synchronization is finished! If the **sync** parameter is omitted or set to **false**, the synchronization will be done by the operating system after some time or when the device is shut down.

The return value of the writeToFile function is either **true** if writing to the file (and optional sanchronization) was successful or **false** if writing or synchronizing to disk failed.

```
Prototype:
var result = writeToFile(string filePath, string stringToWrite,
string option, bool sync);
Example:
var usbMountPath = getVariableValue("@USBMountPath");
var filePath = usbMountPath + "/myFile.txt";
var myTextWithLineBreak = "Hello, this is line 1.\nAndThis is
line 2.\n";
var doSync = false;
```

```
var result = writeToFile(filePath, myTextWithLineBreak,
"append", doSync);
if (result == true)
{
     print("Writing into file was successful!");
}
 else
{
     print("Writing into file FAILED!");
}
```

getAlarmLogRingBufferContent - Read the alarm log ring buffer

The ring buffer alarm log data can be similarly retrieved as the contents of a file. There will also be a vector returned which contains each alarm log entry. Additionally a format string and a delimiter character can be provided to get a formatted result. The format string can contain several keywords delimited by the delimiter character. The keywords that can be used are:

- *timestamp*: The date and time when the alarm was logged in seconds since 1970-01-01 00:00
- *timestring*: The date and time when the alarm was logged as a formatted text. The format is "DayOfWeek Month DayOfMonth Hour:Minute:Second Year", e.g. "Mon Sep 3 13:42:00 2012"
- *alarmtext*: The text that was configured for the logged alarm in the property "Alarm Log Text" of the alarm object.
- *alarmevent*: The event that caused the alarm log entry (either "Enqueued", "Shown", "Confirmed" or "Removed")
- *alarmvariables*: The values of the variables @AlarmShow, @AlarmCurrent and @AlarmCurrentPriority at the time of the logging entry. Each value in hex delimited by the delimiter character.

Example format string: "timestring;alarmtext;alarmevent"
Delimiter character: ";"
This would cause one entry of the returned vector to look like this: "Mon Sep 3 13:42:00 2012;Engine Oil Pressure too high;Shown"

Prototype:
```
var result = getAlarmLogRingBufferContent(string formatString,
string delimiterCharacter);
```

Example:
```
//create a new array and retrieve the ring buffer content
var myLog = new Array();
myLog = getAlarmLogRingBufferContent ("timestring;alarmtext",
";");

//check if successful:
if (myLog.length == 0)
{
    //reading failed! No entries in the buffer?
    print("Error reading Alarm Log Ring Buffer content");
}
else
{
```

```
    //reading successful, myLog is now an array
    //print contents to console:
    print("Alarm log:");
    for (var i = 0; i < myLog.length; i++)
    {
        print(myLog[i]);
    }
}
```

Note: The returned array can be easily passed to a table object so it can be shown on screen (setProperty(IDofTableObject, "Table Data", arrayWithLogContent);)

Note 2: If the formatting of the date and time is not what is desired, the timestring can be used to create a JavaScript Date object which provides easy access to each part of the date and time:

```
//assume that we already have an array filled with log data and
that the timestamp is the first part of each entry
var i = 0;
//extract the timestamp from the log entry
var timestampAsString = myLog[i].substr(0,
myLog[i].indexOf(";"));
Var timestamp = parseInt(timestampAsString);
var myDate = new Date(timestamp * 1000);
//now each part of the date/time can be accessed separately:
//e.g. myDate.getDate() <-- day of month
//or myDate.getSeconds() <-- seconds and so on...
```

deleteAlarmLogRingBufferContent - clear an alarm log ring buffer

Simply use the following function to delete the complete alarm log ring buffer:
```
deleteAlarmLogRingBufferContent();
```

moveDDO - move an object on the screen

The moveDDO function can be used to move an object to a new position. Please use this function with care, this can lead to a very high system load if this function is abused to e.g. create complex animations.

Prototype:
```
var result = moveDDO(number ParentObjectID, number
ObjectIDtoBeMoved, number newX, number newY);
```

The first parameter is the object ID of the parent of the object that shall be moved (check project tree to find the parent). The second parameter is the ID of the object that shall be moved. The third and fourth parameters are the new coordinates inside the parent.

Example:
```
//Move the picture graphic DDO 42. The parent of this ddo is
frame 21
var pictureDDO = 42;
var frameDDO = 21;

//now move to position 84/36:
var movingWorked = moveDDO(frameDDO, pictureDDO, 84, 36);
if (movingWorked)
```

```
{
    print("Moving picture was successful!");
}
else
{
    print("Moving picture failed! :( ");
}
```

Example project: Download this project to see how the MoveDDO function can be used.

showSoftkeyFrames - show faded-out softkey frames

If the fadeout time for softkey frames was set in project properties, the softkey frames will be hidden after that time. To show them again, a physical key needs to be pressed. If the softkey frames shall be shown again by some other event, the showSoftkeyFrames function can be used.

Prototype:
```
var result = showSoftkeyFrames();
```

Note: The function will always return true.

sendCANMessage - send a CAN message

First things first: This function should only be used if there is really no other way of sending the desired message using the CAN protocol settings provided! Implementing a CAN protocol in JavaScript is not a good idea!
Prototype:
```
var result = sendCANMessage(number CAN-Port,
                            number CAN-ID,
                            number DLC,
                            number Byte1,
                            number Byte2,
                            number Byte3,
                            number Byte4,
                            number Byte5,
                            number Byte6,
                            number Byte7,
                            number Byte8);
```
Description of the parameters:
- CAN-Port: Number of the CAN port that shall be used for sending the message (numbering starts with 1)
- CAN-ID: The CAN messages CAN ID
- DLC: Data length code: how many data bytes shall be sent
- Byte1...Byte8: The data bytes of the CAN message

Example:
```
//send CAN message with ID 0x601 and 8 data bytes on port 1:
var result = sendCANMessage(1, 0x601, 8, 0xde, 0xad, 0xbe,
0xef, 0xb0, 0x0b, 0xaf, 0xfe);
if (result)
{
    print("CAN message sent successfully!");
}
else
```

```
{
    print("Failed to send CAN messge!");
}
```

includeFile - include another JavaScript file

With this command it is possible to execute a JavaScript file from within another JavaScript file.
Example:

```
includeFile("example.js")
```

The included file will be executed as if the source code was written in the source file (except when it's a function). This is useful if you have a part of code that you need to execute in many different places.

function - use functions in JavaScript

It is possible to use functions in JavaScript. To create a function, use the following syntax:

Example:

```
function function_name (input variables)
{
<code>
return return_value;
}
```

The input variables as well as the return value are optional. The input variables will be forwardef from the calling JavaScript. Executing the function looks like this:

Example:

```
var result = function (input variables);
```

You can combine this with the includeFile function to create a set of multi purpose JavaScript functions that you can use in different places in your project (Note that you need to declare the functions globally for them to stay alive).

Please note that for a function to be declared once and stay "alive, it needs to be declared like this:

```
function_name = function function_name (input variables)
{
<code>
return return_value;
}
```

generateAction - simulate encoder actions

It is possible to call functions in JavaScript that simulate encoder actions:
Example:

```
generateAction("Navigate Up");//simulates a counterclockwise
encoder movement
generateAction("Navigate Down");//simulates a clockwise encoder
movement
generateAction("Navigate Confirm");//simulates an encoder press
```

deletePageHistory();

For the purpose of prohibiting going back to pages with a password access with the escape key, it is possible to delete the browsing history with this command. To prohibit certain pages from being accessable with the escape key, set the property "Add To Page History" to false.

multimediaCommand - send commands to a MultimediaPlayer object

The multimediaCommand function can be used to send commands to a MultimediaPlayer object. Various commands can be used to control playback or manipulate the playlist of the MultimediaPlayer object.

Prototype:

```
var result = multimediaCommand(number MultimediaPlayerObjectID,
string commandString, number eoptionalValue);
```

The first parameter is the object ID of the MultimediaPlayer object to which the command shall be sent.

The second parameter is the command string (see below for a list of available commands. Note that the command string can either be in upper or lower case letters (e.g. "play" == "Play" == "PLAY"))

The third parameter is an optional/mandatory value that can/needs to be provided to some of the commands.

Note:The multimediaCommand function returns true (if the command was successfully sent to the MultimediaPlayer object) or false (if the command failed). The only exception is: If the command string is "get playlist", an array containing the pathes of all files that are currently in the playlist is returned.

Available commands:
- **"play"**
  Optional Value: None
  Description: Begin playback of the currently selected media file
- **"stop"**
  Optional Value: None
  Description: Stop playback of the currently selected media file
- **"pause"**
  Optional Value: None
  Description: Pause playback of the currently selected media file
- **"back"**
  Optional Value: None
  Description: Jump to the previous file in the playlist
- **"next"**
  Optional Value: None
  Description: Jump to the next file in the playlist
- **"mute"**
  Optional Value: None
  Description: Mute the MultimediaPlayer
- **"unmute"**
  Optional Value: None
  Description: Unmute the MultimediaPlayer (current volume set)
- **"set volume"**
  Optional Value: volume offset, any integer number (volume is restricted to be between 0 and 100, so it only makes sense to set an offset between -100 and 100)
  Description: If no optional value is provided, the volume that is set in the volume bar graph of the MultimediaPlayer is used. If an optional volume is provided, this will be treated as an offset to the current volume.

- **"set position"**
  Optional Value: integer number (positive or negative)
  Description: Jump to a position in the currently played back file. If no optional value is provided, the position that is set in the position bar graph of the MultimediaPlayer is used. If an optional value is provided, the value is treated as an offset to the current position (in seconds).
- **"shuffle"**
  Optional Value: true or false
  Description: If no value is provided, the pressed state of the MultimediaPlayer's shuffle button is used. If true is provided as parameter, the playlist will be randomized. If false is provided, the playlist will return to it's original sequence.
- **"loop"**
  Optional Value: integer number between 0 and 2
  Description: If no value is provided, the value of the MultimediaPlayer's loop list is used. If a value is provided, the values have the following effect: 0: Don't loop, 1: loop current file, 2: loop playlist (after end of last file in playlist was reached, begin at start of playlist)
- **"add to playlist"**
  Mandatory Value: string with the path to the file that shall be added to the playlist
  Description: Add a media file to the playlist of the multimedia player.
- **"remove from playlist"**
  Mandatory Value: string with the path to the file that shall be removed from the playlist. If the file was added to the multimedia player at design time or if it was manually copied to PROJECT_PATH/terminal_files/multimedia directory, it is enough to provide the file name with suffix (e.g. "myMedia.mp3") to remove the file. If the file was added to the playlist at runtime (e.g. from an USB drive), the full path (e.g. "/disk/usbsda1/myMedia.mp3") needs to be used to remove the file from the playlist.
  Description: Remove a file from the playlist.
- **"select playlist entry"**
  Mandatory Value: number >= 1
  Description: Select a certein entry position in the playlist
- **"clear playlist"**
  Optional Value: None
  Description: Remove all files from the playlist. After that the playlist will be empty.
- **"add directory to playlist"**
  Mandatory Value: string that contains a path to a folder that contains media files that shall be added to the playlist
  Description: Add all files in the specified directory to the playlist. Make sure that only media files that can be played back are in the folder.
- **"toggle fullscreen"**
  Optional Value: None
  Description: Switch between fullscreen mode and normal mode
- **"get playlist"**
  Optional Value: None
  Description: Get an array that contains strings. The strings are the full qualified pathes to the media files that are currently in the playlist.

Example:
```
//Our MultimediaPlayer object has object ID 42:
var mmpID = 42;
//Start playback of the current file
```

```
var result = multimediaCommand(mmpID, "play");
if (result == true)
{
   print("Play command successfully sent");
}
else
{
   print("Error sending play command to MultimediaPlayer with
ID " + mmpID);
}
//From here we ommit the handling of the return value (to save
some space in the example code)
//Pause playback of the current file
multimediaCommand(mmpID, "pause");
//Jump to next file in playlist
multimediaCommand(mmpID, "next");
//Jump to 1 minute and 30 seconds in the current file
multimediaCommand(mmpID, "set position", 90);
//Add another file to the playlist
multimediaCommand(mmpID, "add to playlist",
"/disk/usbsda1/mySong.mp3");
//Remove all files from playlist:
multimediaCommand(mmpID, "clear playlist");
//Add the content of a directory to the playlist:
multimediaCommand(mmpID, "add directory to playlist",
"/disk/usbsda1/music/");

//Get the complete playlist and show all pathes in a table
(with ID 55)
var playlistEntries = new Array();
playlistEntries = multimediaCommand(mmpID, "get playlist");
setProperty(55, "Table Data", playlistEntries);
```

Please note that the example above is not meant to be executed (makes no sense to execute the commands in that order in one script).

save2DGraphHistory - Save the value history of a 2D Graph (to external media)

The value history of a 2D Graph object is stored in binary ring buffers on the non-volatile memory of the device. In order to process the data on a PC (e.g. spreadsheet application), it needs to be copied to an external media (e.g. usb drive).
As soon as this function is called, the copy process will start. The progress can be observed in the Copy Progress Variable[451] of a 2D Graph. While copying is in progress all new values of the 2D Graph will be buffered and will not get lost. After the copy has finished (Copy progress variable value is 100), the external media can be removed and the data can be processed with the Graph History Tool[459] on a PC.
Prototype:
var result = save2DGraphHistory(number Graph-DDO-ID, string targetPath);
The result of this function is true if the copy process could be started (correct object ID and target path was provided). This does not mean that the copy process has successfully finished.
Example (copy to USB drive):
```
//Get path of mounted USB drive:
var usbMountPath = getVariableValue("@USBMountPath");
```

```
//Save history to target path. The 2D Graph has object ID 42
var result = save2DGraphHistory(42, usbMountPath);
if (true == result)
{
    print("Copy 2D Graph history started");
}
else
{
    print("Starting copy 2D Graph history FAILED");
}
```

delete2DGraphHistory - Delete the value history of a 2D Graph object

Use this function to completely delete the value history of a 2D Graph object. This will instantly delete the binary ring buffer on the non-volatile memory and the currently shown value lines.
Prototype:

```
var result = delete2DGraphHistory(number Graph-DDO-ID);
```
The result of this function is true if a correct graph object ID was provided.

requestVariableValueFromOwner - Initiate a read request to the owner of a variable (CANopen SDO Upload Request)

This function can be used as a trigger to send out a read request to the owner of a variable. Currently this function can only be used to send out a CANopen SDO Upload Request. See SDO Upload Request Support[328]
This function can be called multiple times in one script for different variables. All requests will be buffered and send out one after another after the equivalent SDO Upload Responses was received (or after timeout).
Prototype:

```
var result = requestVariableValueFromOwner(string
NameOfCANopenVariable);
```

The result of this function is *true* if the variable is a CANopen variable and the Upload Request was accepted by CANopen. It returns false if either the settings of the variable were not correct (e.g. if it was no CANopen variable) or if the CANopen stack was not correctly configured (e.g. no appropriate SDO Client Channel settings were made for the CANopen ECU).

startTouchscreenCalibration - Start calibration of resistive touchscreen

Use this function to start the calibration process of a resistive touchscreen. Capacitive touchscreens do not need calibration.

When this function is called, the calibration process is displayed but PClient will continue to run (e.g. cyclic scripts, cyclic CAN messages) but won't accept touch events and will not paint anymore until calibration is finished.

Prototype:

```
var result = startTouchscreenCalibration();
```

The result of this function is *true* if the touch calibration process was started successfully and *false* if it is either not supported on the current device or an error occurred.

Important Notes:
- On devices with capacitive touch the calibration process might start but will have no effect. Capacitive touchscreens do not need calibration.

- On devices with I.MX35 CPU, the new calibration values are directly used after the calibration has finished. On devices with I.MX6 CPU, the new calibration values will only have an effect after a cold boot (write 0xb007 to variable @PWR_CurrentPowerMode).

writeEEPROM - Write a value to the EEPROM

The writeEEPROM function allows writing values to EEPROM. The values in EEPROM are very save and will survive a complete a software update as well as unexpected power losses (except during the writing process).

Note: The number of EEPROM write cycles is limited (device dependent). Do not write to EEPROM too often (e.g. not in a cyclic script or CAN message)!

The maximum usabe EEPROM size for all devices is 28672 bytes.

Prototype:
```
var result = writeEEPROM (number EEPROM_ADDRESS, number LENGTH,
number/bool/string VALUE);
```

Parameters:
- EEPROM_ADDRESS: The start address in the EEPROM to write the data to
- LENGTH: The data length. The following rules apply:
  - ? If writing a bool, the length needs to be set to 1
  - ? If writing a number, the length needs to be 1, 2, 3 or 4
  - ? If writing a string, the area starting at the address will be reset to 0 in EEPROM and after that the string will be written (if a string is shorter than the provided length, still the whole length is reset to 0 before the shorter string is written). If a string is longer than length, only the given number of bytes will be written to EEPROM.
  - ? EEPROM_ADDRESS + LENGTH may not be greater than the maximum EEPROM size (28672)
- VALUE: Currently the following JavaScript data types are supported: number, string and bool. If e.g. an array object is provided as value parameter, the writeEEPROM function will fail.
- Return value (result): If the provided value was written successfully, the number of bytes that were written are returned. If writing the value to EEPROM failed, undefined will be returned (observe console error messages when testing the project).

Note: The EEPROM functions will only work on the device, not in the PClient simulation.

Examples:

```
var ret = writeEEPROM(0, 4, 0xdeadbeaf);//write the number
0xDEADBEAF into 4 bytes starting at position 0
print("Write 0xdeadbeef: " + ret);
//console output: Write 0xdeadbeef: 4

var ret = writeEEPROM(4, 2, 32767); //write the number 32767
into 2 bytes starting at position 4
print("Write 32767: " + ret);
//console output: Write 32767: 2
```

```
var ret = writeEEPROM(6, 20, "Hallo Test!"); //write the string
"Hallo Test!" into 20 bytes starting at position 4
print("Write str: " + ret);
//console output: Write str: 11

ret = writeEEPROM(26, 1, true); //write a bool into 1 byte at
position 26
print("Write b: " + ret);
//console output: Write b: 1

ret = writeEEPROM(27, 1, -1); //write the number -1 (minus 1)
into 1 byte at position 27
print("Write minus1: " + ret);
//Write minus1: 1

ret = writeEEPROM(28, 2, 0x55); //write the number 0x55 into 2
bytes starting at position 28
print("Write 16bit 0x55: " + ret);
//console output: Write 16bit 0x55: 2
```

readEEPROM - read a value from the EEPROM

The writeEEPROM function allows reading values from EEPROM. The maximum usable EEPROM size is **28672** bytes.

Prototype:
```
var value = readEEPROM(number EEPROM_ADDRESS, number LENGTH,
string OPTIONAL_DATATYPE);
```

Parameters:
- EEPROM_ADDRESS: The start address in the EEPROM to read the data from
- LENGTH: The data length. The following rules apply:
  ? If reading a bool, the length will be ignored and 1 is always used
  ? If reading a number, the length needs to be 1, 2, 3 or 4
  ? EEPROM_ADDRESS + length may not be greater than max EEPROM size (28672)
- OPTIONAL_DATATYPE: This is a hint about the type of data that shall be read. The readEEPROM function will try to use the correct data type as its return value. If this parameter is ommited or a non supported value is used, a byte array will be used as return type (see Return value below)
  ? Possible values:
    ? "bool"
    ? "number"
    ? "string"
- Return value (value): In case reading failed, undefined will be returned (observe console error messages when testing the project). If reading worked, the value that was read from EEPROM will be returned. The data type of the returned value depends on OPTIONAL_DATATYPE parameter. If this parameter was omitted, a byte array object is returned that contains the single data bytes of the EEPROM (as numbers).

Note: The written and returned numbers are treated as signed 32Bit integers regardless of the actual data length! If e.g. -1 is written with a length of 1 and then

read again, 255 will be returned (see example below)! The handling of such data is up to the script programmer!

Examples (see writeEEPROM for the according write functions):

```
var x = readEEPROM(0, 4, "number"); //read 4 bytes from postion
0, interpret the data as a number
print("Read 0xdeadbeef: " + x);
//console output: Read 0xdeadbeef: -559038801
```

Why -559038801?
0xdeadbeef in binary is 1101 1110 1010 1101 1011 1110 1010 1111 (32 bits).
According to the note above, the 1 in the highest bit is interpreted as the sign.
1101 1110 1010 1101 1011 1110 1010 1111 as a 32 bit signed number is -559038801

```
var n = readEEPROM(4, 2, "number"); //read 2 bytes from postion
4, interpret the data as a number
print("Read 32767: " + n);
//console output: Read 32767: 32767
```

```
var str = readEEPROM(6, 20, "string"); //read 20 bytes from
postion 6, interpret the data as a string
print("Read str: " + str);
//console output: Read str: Hallo Test!
```

Why "Hallo Test!", even though 20 bytes are read?
As seen below, the positions 6 to 16 (11 bytes -> 11 characters in "Hallo Test!") are filled with the letters. The rest was set to 0. 0 is interpreted as the stop sign (i.e. the end of a string).

```
var b = readEEPROM(26, 1, "bool");
print("Read b: " + b);
//console output: Read b: true
```

```
var minus1 = readEEPROM(27, 1, "number");
print("Read minus1: " + minus1);
//console output: Read minus1: 255
```

Why 255? -1 in binary is, in 32 bit, 1111 1111 1111 1111 1111 1111 1111 1111).
But only 1 byte is stored. 1111 1111 equals 255.

```
var b16 = readEEPROM(28, 2, "number");
print("Read 16bit 0x55: " + b16);
//console output: Read 16bit 0x55: 85
```

```
var arr = readEEPROM(0, 29);
for (var i = 0; i < arr.length; ++i)
{
    print("arr[" + i + "]: " + arr[i]);
}
```

```
//console output:
// arr[0]: 175  -> 0xAF
// arr[1]: 190  -> 0xBE
// arr[2]: 173  -> 0xAD
// arr[3]: 222  -> 0xDE -> makes 0xDEADBEAF
// arr[4]: 255  -> 0xFF
// arr[5]: 127  -> 0x7F -> makes 32767
// arr[6]: 72   -> 0x48 -> ASCII H
// arr[7]: 97   -> 0x61 -> ASCII a
// arr[8]: 108  -> 0x6C -> ASCII l
// arr[9]: 108  -> 0x6C -> ASCII l
// arr[10]: 111 -> 0x6F -> ASCII o
// arr[11]: 32  -> 0x20 -> ASCII <blank space>
// arr[12]: 84  -> 0x54 -> ASCII T
// arr[13]: 101 -> 0x65 -> ASCII e
// arr[14]: 115 -> 0x73 -> ASCII s
// arr[15]: 116 -> 0x74 -> ASCII t
// arr[16]: 33  -> 0x21 -> ASCII !
// arr[17]: 0   -> String stop sign
// arr[18]: 0   -> String stop sign
// arr[19]: 0   -> String stop sign
// arr[20]: 0   -> String stop sign
// arr[21]: 0   -> String stop sign
// arr[22]: 0   -> String stop sign
// arr[23]: 0   -> String stop sign
// arr[24]: 0   -> String stop sign
// arr[25]: 0   -> String stop sign
// arr[26]: 1   -> bool true
// arr[27]: 255 -> 0xFF
// arr[28]: 85  -> 0x55
```

Why was this long array returned? No data type was given in the read command, so the bytes are returned as a JavaScript array.

Please note: Numbers are stored LSB first, Strings are stored linearly from beginning to end.

getCallerID - Retrieve the ID of the object/variable that triggered the script

With this function the ID of the object (graphical object or CAN mapping) or the name of the variable that triggered the execution of the current script.

Prototype:
```
var result = getCallerID();
```

result will contain the ID of the object if the script was triggered by a graphical object or a CAN mapping, or the name of the variable if the script was triggered by a variable.
If neither is the case (only theoretical, should never happen), the result will be undefined.

createUSBDirectory - Create a directory on the USB stick or in user_files

This function can be used to create a directory on a USB stick or in the /opt/user_files directory on the device itself.
provide a directory name or hierarchy as a parameter. If the path starts with /opt/user_files, the directory hierarchy is created inside the user_files directory. If the path is not

below /opt/user_files, the current USB mount path is automatically prepended to the path and the directory is created on the USB stick.
 The path may not contain a ".." or "*".

Prototype:

```
var result = createUSBDirectory(string directoryPath);
```

The result of the function will be true if creating the directory was successful (or the directory already exists) or false if
-the path contains a ".." or "*"
-the path is not inside of /opt/user_files and no USB stick is inserted
-creating the directory failed for unknown reasons

getDirectoryContent - Get the contents of a directory

With this function the contents of a directory can be read.
The first parameter is the path to the directory that shall be read. The second parameter is optional and can be used to filter the result to contain only files or only directories. Use "DIRECTORIES_ONLY" or "FILES_ONLY" as filter options. If the second parameter is omitted or has an other value than the two filter options, files and directories will be returned.

Prototype:

```
var result = getDirectoryContent(string directoryPath, string optionalFilter);
```

The function will return undefined if the contents of the provided directory could not be read. If the contents were read successfully, an array of strings will be returned that contains all file and/or directory names inside of the provided directory path.

deleteFileOrDirectory- Delete a file or directory on a USB stick or in user_files

With this function a file or directory can be deleted. A file or directory can only be deleted if it is either on a USB stick or in the user_files directory (/opt/user_files).
 The function's only parameter is the path to the file or directory that shall be deleted. To erase the contents of a folder an asterisk may be provided at the the end of the path (e.g. "/opt/user_files/*").
 A ".." may not be contained in the provided path.

Prototype:

```
var result = deleteFileOrDirectory(string directoryOrFilePath);
```

The function returns true if the directory or file(s) were deleted successfully. It returns false if the file(s) or directory could not be deleted.

setFocus - Set navigation focus to certain object

 Tutorial video of the setFocus function
https://www.youtube.com/watch?v=9IjFMmBQ45Q

This function sets navigation focus to object identified by DDO-ID and even opens it for input if the second parameter is 'true' or '1'.
 The second parameter can be skipped - then the object will not be opened for input.

The function throws exceptions if the parameters are not correct or if the object cannot be found in project.

Return value is 'true' if all preconditions are met and focus change gets triggered. Even in this case it may fail if the object is not shown on current page.
 Return value is false (and nothing is triggered) if
•The object is not an input.
•A button should be opened for input.
•The object is disabled.
•An object is currently 'open for input' i.e. operator is currently changing some value.

Prototype:

```
var success = setFocus(number DDO-ID, bool openForInput);
```

Example:

```
//set the focus to the object with ID 93
setFocus(93, "true");
```

copyFileOrDirectory - Copy file(s) or directory/ies

This function copies a file or directory or files or directories within the file system (including USB stick).
The two (obligatory) parameters are the source and destination file path.

Return value is 'true' if the copy process was successful.
Return value is false if the copy process was not successful or if the parameters were invalid or missing.

To be able to copy files into the user_files directory, it can be created using the function.

Prototype:

```
var success = copyFileOrDirectory(String sourcePath, String
destinationPath);
```

Examples:

```
//copy the file test.txt from /opt to /opt/test
copyFileOrDirectory("/opt/test.txt","/opt/test/");

//copy the file test.txt from /opt to /opt/test and rename it
test2.txt
copyFileOrDirectory("/opt/test.txt","/opt/test/test2.txt");

//copy all files from /opt/test_dir1 to /opt/test_dir2
copyFileOrDirectory("/opt/test_dir1/*","/opt/test_dir2/");
```

```
//copy the directory /opt/test_dir inluding all its content
to /opt/parent_dir (resulting in /opt/parent_dir/test_dir
copyFileOrDirectory("/opt/test_dir/","/opt/parent_dir/");
```

Please note: This function blocks the execution until it is finished. Blocks means: The thread that caused the execution of the script will block. Usually this is the PClient main thread including any painting operation. The programmer is advised to show a warning to the device user when a blocking function is called.
 If the script was triggered e.g. by a CAN stack (e.g. timeout or received event), it will also block that thread and will e.g. not send cyclic messages during the block!
The time it takes to finish this function depends on the size of the copied data.

moveFileOrDirectory - Move file(s) or directory/ies

This function moves a file or directory or files or directories within the file system (including USB stick). A file or directory can only be moved if it is either on a USB stick or in the user_files directory (/opt/user_files).
The two (obligatory) parameters are the source and destination file path.

Return value is 'true' if the move process was successful.
Return value is false if the move process was not successful or if the parameters were invalid or missing.

Prototype:

```
var success = moveFileOrDirectory(String sourcePath, String
destinationPath);
```

Examples:

```
//move the file test.txt from /opt/user_files to /opt/test
moveFileOrDirectory("/opt/user_files/test.txt","/opt/test/");

//move the file test.txt from /opt/user_files to /opt/test and
rename it test2.txt
moveFileOrDirectory("/opt/user_files/test.txt","/opt/test/test2
.txt");

//move all files from /opt/user_files/test_dir1
to /opt/test_dir2
moveFileOrDirectory("/opt/user_files/test_dir1/*","/opt/test_di
r2/");

//move the directory /opt/user_files/test_dir inluding all its
content to /opt/parent_dir (resulting
in /opt/parent_dir/test_dir
copyFileOrDirectory("/opt/user_files/test_dir/","/opt/parent_di
r/");
```

Please note: This function blocks the execution until it is finished. Blocks means: The thread that caused the execution of the script will block. Usually this is the PClient main thread

including any painting operation. The programmer is advised to show a warning to the device user when a blocking function is called.
If the script was triggered e.g. by a CAN stack (e.g. timeout or received event), it will also block that thread and will e.g. not send cyclic messages during the block!
The time it takes to finish this function depends on the size of the moved data.

syncFilesystem - Forces finishing of all file operations

This function performs a "sync" command.
This is relevant if the device is shut down after a copy / move process. The OS finishes these operations quite slowly in the background.

Prototype:

```
syncFilesystem();
```

Please note: This function blocks the execution until it is finished. Blocks means: The thread that caused the execution of the script will block. Usually this is the PClient main thread including any painting operation. The programmer is advised to show a warning to the device user when a blocking function is called.
If the script was triggered e.g. by a CAN stack (e.g. timeout or received event), it will also block that thread and will e.g. not send cyclic messages during the block!
The time it takes to finish this function depends on the size of the copied/moved data.

openPDFReader - Open the PDF Reader

This function opens the [PDF reader](442), optionally with a PDF file and / or a "Open File" button.

Prototype:

```
bool openPDFReader(optional:String pathToPDFtoOpen,
optional:bool showOpenButton);
```

Parameters:

- pathToPDFtoOpen: String with full path to the PDF file to open. No restrictions. Can be anywhere on the file system or e.g. USB stick.
- showOpenButton: Flag to indicate if the open button shall be shown or not. Note that if no path was provided, the open button is shown automatically.

Return value:

- true if starting the PDF reader was successful (this is no indication if the provided PDF could be loaded)
- false if starting the PDF reader failed

Note that the PDF reader cannot be opened in the PClient simulation, only on the device.

runSystemCommand - run a Linux system command

With great power comes great responsibility!

This function executes any command on the Linux system console.
We cannot stress enough that it is very easy to slow down or even completely stop the PClient misusing this command, it is even possible to destroy almost all data on the device.
But we also know that it can be very useful in certain situations, e.g. to be able to change the boot logo on a device at runtime.

Prototype:

```
var ret = runSystemCommand(string command, bool
waitForFinished, bool returnCommandOutput);
```

Parameters:

- command: A string with the command to execute
- waitForFinished: Optional bool flag. If set to true (which is also default if the parameter is not set), PClient main loop waits until the command has finished. If set to false, the function will return immediately but the caller will not know the final result of the command
- returnCommandOutput: If this flag is set to true, this function will return the complete console output of the executed command instead of the return value. This parameter is also optional, if not set, the default value is false. Note: If set to true, waitForFinished also needs to be true.

Return values:

- true if the command executed successfully while returnCommandOutput was set to false or not set
- A string containing the complete output of the executed command if returnCommandOutput was set to true
- If anything fails, an exception is thrown and the execution of the whole script file is interrupted at the line of the system call. Surround with try/catch block for error handling.

Good example:

```
var ret = runSystemCommand("top -n 1", true, true);
```

Bad example:

```
var ret = runSystemCommand("top", true, true);
```

Why is it bad? The example will never return because top will keep running, thus the JavaScript file will never finish.

Good example:

```
var ret = runSystemCommand("rm -rf /opt/data/own_data", true,
false);
```

Bad example:

```
var ret = runSystemCommand("rm -rf /opt/*", true, false);
```

Why is it bad? The example will delete ALL files and folders in /opt, killing the project, PClient, updater and all user data.

Once again: **Please use this function with care**. There are no limitations in this function, whatever is the content of <command> will be forwarded to the system.
There are plenty of commands that take a long time to process.

Always test the commands that should be used thoroughly before implementing them for field use!

getTriggerEvent - obtain the trigger for a script execution

This function returns the trigger that executed the currently running JavaScript file. The return is a string identical to the event names as they are used in the program.

Prototype:

```
var eventName = getTriggerEvent();
```

Return value:

The event that triggered the JavaScript.
Possible values are:

Execute Script
Jump To Page
Set Value
Scale Value
Next Softkey Level
Previous Softkey Level
Start Application
Send Event To Plugin
Open For Input
Navigate Up
Navigate Down
Navigate Confirm
Open PDF Reader
Set Media Player Command

OnProjektInit
OnProjektExit
OnPageInit
OnPageExit
OnProjektRepeat
OnPageRepeat

OnEncoderMove
OnPress
OnRelease
OnLongPress
OnLongRelease
OnDblClick

OnIntValueChange
OnStringValueChange
OnEnterValue
OnEnterNewValue
OnEscape
OnOpenForInput
OnEnabledWithFocus
OnEnabledWithoutFocus

OnValueChangeByOwner
OnValueChangeRequest
OnValidStateChange

OnPowerOn
OnLowPower
OnSleep

OnReceiveTimeoutAction
OnReceiveTimeoutResolvedAction
OnMappingReceived

OnHeartbeatTimeout
OnHeartbeatTimeoutResolved

OnEncoderPress
OnEncoderRelease
OnEncoderLongPress
OnEncoderLongRelease
OnEncoderDoublePress

setVariableValid - set a variable as valid / invalid

This function sets the validity state of a variable.

Prototype:

```
var result = setVariableValid(string variable name, bool
valid);
```

or

```
var result = setVariableValid(number index, number subindex,
bool valid);
```

Parameters:

- variable name: String with an existing variable name
- index/subindex: Alternatively to a variable name string a combination of a variable index and subindex.
- valid: Boolean with false = invalid and true = valid.

Return value:

- true if the operation was successful
- false if the operation was not successful (shouldn't really occur)
- undefined if the variable wasn't found

Note that the function works both for variables belonging to the PClient as well as external variables, so use with care.

RS232 JavaScript Functions □560

7.4.1   RS232 JavaScript Functions

Support for the RS 232 interface is provided by a JavaScript interface - there are no dedicated variables involved.

The approach requires instantiating an 'RS232Handler' object which represents one RS 232 interface.
Currently this uses the one port provided on the device.
If more than one port should exist in the future, then one instance per port would have to be created.

Once the object is created, all function calls have to be done for this object.

Note that this interface was designed mainly for serial printers - that's why the 'read' functionality is rather limited for now.

RS232Handler - constructor to get an RS232 instance

The constructor has no parameters - it creates an instance but does not open any interface.
If this handler should be used as long as the PClient is running, it should be created e.g. in a project init script and assigned to a global variable.

Prototype:
```
 var myRS232 = new RS232Handler();
```

Example:
```
 // Generate a global RS232 handler
 myRS232_handler = new RS232Handler();
```


RS232Handler.init - open interface and set baud rate

This method opens the RS232 interface with an optionally specified baud rate and a wait time for the RS232 read function.
If called without parameters or with unsupported values, the RS 232 interface is initialized with 115200 bit/s and 100 ms wait time.
it is always initialized to use:
8 data bits
No parity
1 stop bit
these settings can NOT be changed at the moment.

The following baud rates [bit/s] can be configured:
50
75
110
134
150
300
600
1200
1800
2400
4800
9600
19200
38400
57600
115200
230400
460800
500000
576000
921600
1000000
1152000
1500000
Every other value will result in a setting of 115200 bit/s.

The wait time means that the read function will wait for that time if the RS232 buffer is empty. This can be used when communicating with a device that needs some time to deliver an answer to a write.
The wait time can be set in steps of 100 from 0 to 25500 ms.

Prototype:
```
var result = <RS232Handler instance>.init(number
newBaudrate,number waitTime);
```

Examples:
```
// Open an RS232 interface with the default settings of 115200
baud / s and 100 ms wait time)
myRS232.init();

// Open an RS232 interface with 9600 baud / s and 100 ms wait
time)
myRS232.init(9600);

// Open an RS232 interface with 576000 baud / s and 0 ms wait
time)
myRS232.init(576000,0);
```

Return values:

0 - port successfully opened

2 - port cannot be opened
3 - parameters cannot be applied to port
10 - current settings cannot be read from port

RS232Handler.encoding - set the character encoding of the target

If the target machine (i.e. printer) should not accept UTF-8 encoded text, this property allows the modification of strings before they are sent.
Reading is not affected - the read method only returns an array of the raw bytes received.

The following encodings are supported:

    Big5
    Big5-HKSCS
    CP949
    EUC-JP
    EUC-KR
    GB18030
    HP-ROMAN8
    IBM 850
    IBM 866
    IBM 874
    ISO 2022-JP
    ISO 8859-1 to 10
    ISO 8859-13 to 16
    Iscii-Bng, Dev, Gjr, Knd, Mlm, Ori, Pnj, Tlg, and Tml
    KOI8-R
    KOI8-U
    Macintosh
    Shift-JIS
    TIS-620
    TSCII
    UTF-8
    UTF-16
    UTF-16BE
    UTF-16LE
    UTF-32
    UTF-32BE
    UTF-32LE
    Windows-1250 to 1258

Prototype:
```
myRS232.encoding = STRING <encoding name>;
```

Example:
```
// Change the encoding to Windows 1252
myRS232.encoding = "Windows-1252";
```

RS232Handler.getBaudrate - read the baud rate currently used

This method returns the baud rate currently used for the serial port.

Prototype:

```
var myBaudrate = myRS232.getBaudrate();
```

The method returns 0 if the port is not initialized yet.

Note that this method can't actually read the current baud rate. It returns the saved value of the last baud rate that has been set successfully with the .init method.

RS232Handler.getPort - read the name of the RS232 interface currently used

This method returns a string containing the path and device file currently opened by this instance.
This mainly exists for debugging.

Prototype:
```
var port = myRS232.getPort();
```

Example:
```
// Retrieve the filepath and device name of the RS232 port
var myPort = myRS232.getPort();
print("RS 232 port used : '" + myPort + "'.");
```

RS232Handler.writeFromBuffer - write a string or an array of numbers to the serial port

This method writes a string or number array to the serial port.
This only works if the RS232Handler instance has been created and initialized before.

It accepts a character string as well as an array of numbers as buffer to be sent via serial interface.
Strings will be converted to UTF-8 format used everywhere in PClient if 'RS232Handler.encoding' is not modified.

As a second parameter, one can provide the number of bytes to send from the buffer i.e. this allows to use a buffer which could contain more than one wants to send currently.
If an array of numbers is provided, the size is limited to the size of the array - even if a greater number is provided in 'bytesToSend'.
If a string is provided, the size is considered to be characters. Depending on the desired encoding the number of bytes sent can differ.

If the buffer is provided as string, the 'encoding' property of 'RS232Handler' is used to convert the string before sending (see examples).

Prototype:
```
var errorCode = RS232Handler.writeFromBuffer(<String or number
array>, optional number charactersToSend);
```

Examples:
```
// Since UTF-8 is used for character encoding in PClient the
special characters are sent as two or more bytes by default
// which in the following example can result in unprintable
characters.
// Special characters in this example are sent as two bytes
each (for default UTF-8).
```

```
myRS232.writeFromBuffer("Special characters: 'ÄÖÜ äöü'\r\n");

// If the receiver expects special characters according to an
extended ASCII encoding
// the special characters have to be provided as an array of
numbers.
// If 'latin 1' (ISO 8859) is expected the same text as above
could look like this:
myRS232.writeFromBuffer("Special characters: '");
myRS232.writeFromBuffer([0xc4, 0xd6, 0xdc, 0x20, 0xe4, 0xf6,
0xfc]);
myRS232.writeFromBuffer("'\r\n");

// In order to convert the data sent to another character table
one can define the 'encoding' property of 'myRS232':
myRS232.encoding = "Windows-1252";
// Now even the special characters will only be sent as one
byte:
myRS232.writeFromBuffer("Special characters: 'ÄÖÜ äöü'\r\n");
```

Return values:

0 - write successful
2 - port is not opened yet
6 - writing failed

RS232Handler.readIntoBuffer - read data from serial interface

Calling this method reads all data currently available on the RS232 interface and stores it in a number array.
Without a parameter, the method reads up to 1024 bytes.
With an optional parameter the number of bytes can be limited.

If no data is available, an empty array will be returned after the set wait time.
This method entirely relies on the buffering of the underlying driver.

Note that this is a very simple interface intended to e.g. read some configuration data from a printer.
It is by no means designed to frequently read data e.g. from a GPS receiver or other similar source.
If this warning should not prevent you from trying anyways: Expect lag in the PClient (drawing) performance (as both share the same thread) and be aware that you have to manage getting the data at the right time and in the right amount somehow without blocking the PClient with 'busy waiting'.

Prototype:
```
var number[] = RS232Handler.readIntoBuffer(number bytesToRead);
```

Example:
```
//Read the RS232 interface
var readArray = myRS232.readIntoBuffer();
```

```
//convert the numeric array into a String
var arrayAsString = String.fromCharCode.apply(null, readArray);
// show data received in String Field [57]:
setProperty(57, "Preview Value", arrayAsString);
```

7.5   Programming Guidelines

In this chapter we want to provide some general programming guidelines for JavaScript programming with our devices.
There is no absolute truth, no one best way, but avoiding the worst ways is a good start.

Check if a functionality exists

Before implementing something complex, check thoroughly if there is not a functionality implemented in the program that does what you want to do. If in doubt, contact your vendor for support and consulting.

No synchronization between PClient and JavaScript

JavaScript has no direct access to any variables or data in the PClient. Use our functions to read and write variable values[530] and object properties[531] .
But even doing that does not establish a connecttion between PClient and JavaScript.
Reading a PClient variable in JavaScript is like taking a momentary snapshot.

No time in a script

It is not possible to spend any "time" in a script, i.e. wait for a certain event. You cannot set a variable, then wait 5 seconds for the outcome of the change, then continue in the script. If you need a time component for e.g. a gradual value change, you need to use a cyclic script, e.g. on the event OnPageRepeat, and increment a variable with each run.

Don't put everything in OnProjektRepeat

It seems easy and convenient to just put all your code in the script OnProjektRepeat, this way everything will be executed all the time and all JavaScript variables will always be available. But such a script will become very big, and all code will be executed all the time, which will slow down your project.

Use the right events for the right code

Always consider which event is the right event for the code you want to execute.
E.g. if you have a calculation dependent of a variable, put the code to the event OnValueChangeByOwner of that variable.

Avoid using "global" JavaScript variables

It is possible to declare JavaScript variables globally, so that they will be available in other scripts. But this should be avoided as much as possible. It is safer to store values in PClient variables, read them at the beginning of the script and write them back after modification in the script.

Don't combine commands

We've had crashes and problems sometimes with code lines like

```
setVariableValue("test",(getVariableValue("test")+1);
```

In a million runs this might crash the PClient one time. Don't program like that, it does not save any resources over spreading it out like this:

```
var test_js = getVariableValue ("test");
test_js = test_js + 1;
setVariableValue ("test",test_js);
```

Functionalities

8   Functionalities

8.1   Alarms

This chapter describes the concept of alarms.
Alarms are popup pages or frames that can be shown to indicate certain events.
Alarm templates can be defined in the project properties.
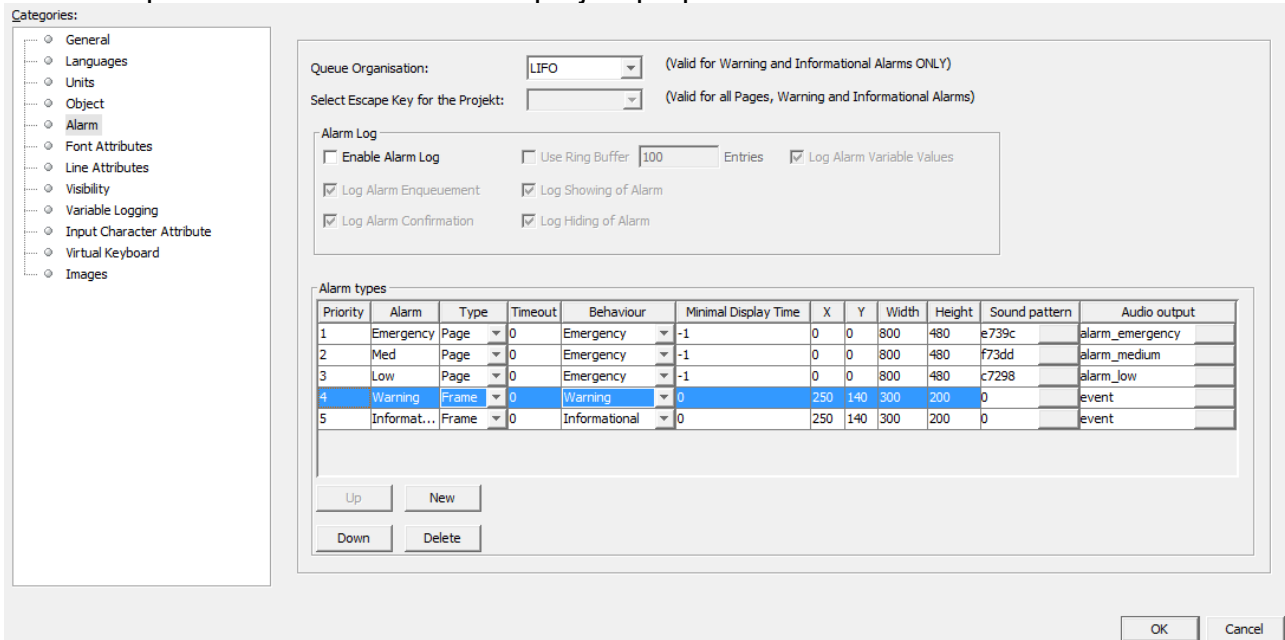
Alarm Properties

Alarms are characterized through the following properties:

- **Priority**: This property decides which alarm is shown in front if several alarms with different priorities are shown
- **Alarm**: A name for the alarm template. There are 3 pre-configured alarm templates, Emergency, Med and Low
- **Type**: Alarms can be shown in a full screen or as a popup above the currently displayed page
- **Timeout**: Enter a time in milliseconds after which an alarm of the selected type will disappear by itself. 0 means the alarm does not disappear by itself. Cannot be set for the alarm behaviors Emergency and Warning
- **Behaviour**: Emergency alarms can only be cleared by setting the **@AlarmShow** variable. Warning alarms can be cleared by setting the **@AlarmShow** variable and also by pressing the "Escape" key. Informational alarms can be cleared by setting the **@AlarmShow** variable, by pressing the "Escape" key and by a timeout (see above)
- **Minimal Display Time**: An alarm can only be cleared (by pressing the "Escape" key) after this time (in milliseconds)
- **X,Y**: Coordinates for the position of the alarm on the display. Fixed to 0,0 if the type of the alarm is set to full page
- **Width, Height**: Width and height of the alarm on the display. Fixed to the resolution of the display if the type of the alarm is set to full page
- **Sound pattern:** Here you can define a pattern that will be played when the alarm is shown. This sound pattern will then be played back with the beeper each time the event occurs. The volume of the beeps is dependent on what event happened (event

or alarm) and the value of the [beeper volume variables](#)[174]. A sound pattern consists of the number of of repetitions and a bit coded pattern of on/off states. In the dialog that pops up when changing this property, the pattern can be defined by clicking on the boxes.

Each box represents 100ms of Beeper on or off. Note that the sound pattern also works on devices with a speaker.

In the screenshot above, the pattern 0xffc00 was defined. In binary this is 11111111110000000000. This will switch the beeper on for one second and then off.

100ms ON, 100ms ON, 100ms ON, 100ms ON, 100ms ON, 100ms ON, 100ms ON, 100ms ON, 100ms ON, 100ms ON,

100ms OFF, 100ms OFF, 100ms OFF, 100ms OFF, 100ms OFF, 100ms OFF, 100ms OFF, 100ms OFF, 100ms OFF, 100ms OFF

The sound pattern can also be directly entered as a hexadecimal value.

- **Audio Output:** Here you can configure an audio file that will be played when the alarm is shown (only if the target device is equipped with an internal sound card (internal speaker and/or line-out)). For a detailed description of the Audio Output configuration see [Events and Actions](#)[432]

You can define a new alarm template by clicking the "New" button. A new alarm template with a pre-set name will be created. You can change all the above properties for that alarm.

You can change the priorities of alarm templates. If several alarms are active at the same time, alarms with a higher priority are shown in front. For alarms with the same priority showing at the same time, you can set at "Queue Organisation" whether the older or the newer alarms are shown in front.

You can also define which key shall be used as the "Escape" key.

Alarm Log

The device can log alarms. To enable alarm logging, click on "Enable Alarm Log". You can then set which events you want to log. You can also set whether you want to log all alarms or us a ring buffer. If you use a ring buffer, you can set the number of alarm events that you want to log.

Note: Alarm logging is only enabled when the variable @LogLevel is set to a value > 0.

Note: Logging for each alarm instance can be enabled or disabled. Simply enable or disable the property "Log this Alarm" in the properties of the alarm.

The following options are available for alarm logging:

- **Enable Alarm Log:** Enable or diasable logging of alarms.
- **Use Ring Buffer:** Enable this to log alarms into a binary ring buffer and select how much entries the ring buffer shall use. If the ring buffer is full, the oldest entries are overwritten. The ring buffer can be read and cleared with the JavaScript functions getAlarmLogRingBufferContent [541] and deleteAlarmLogRingBufferContent [542].
If this is disabled, the alarms will be logged into a text file located in the path stored in @LogPath
- **Log Alarm Variable Values:** If this is enabled, the values of the variables @AlarmShow, @AlarmCurrent and @AlarmCurrentPriority are written into the log with each entry.
- **Log Alarm Enqueuement:** If this is enabled a log entry is done when the alarm is put into the alarm queue.
- **Log Showing of Alarm:** If this is enabled a log entry is done when the alarm is shown on the screen.
- **Log Alarm Confirmation:** If this is enabled a log entry is done when the alarm is confirmed (e.g. user pressed escape key).
- **Log Hiding of Alarm:** If this is enabled a log entry is done when the alarm is closed and not shown anymore.

Creating an alarm

After you have created or configured alarm templates, you can add an alarm to your project. To do this, right-click on "Alarms()" in the project tree, go to "New Alarm Type" and choose the desired alarm template.
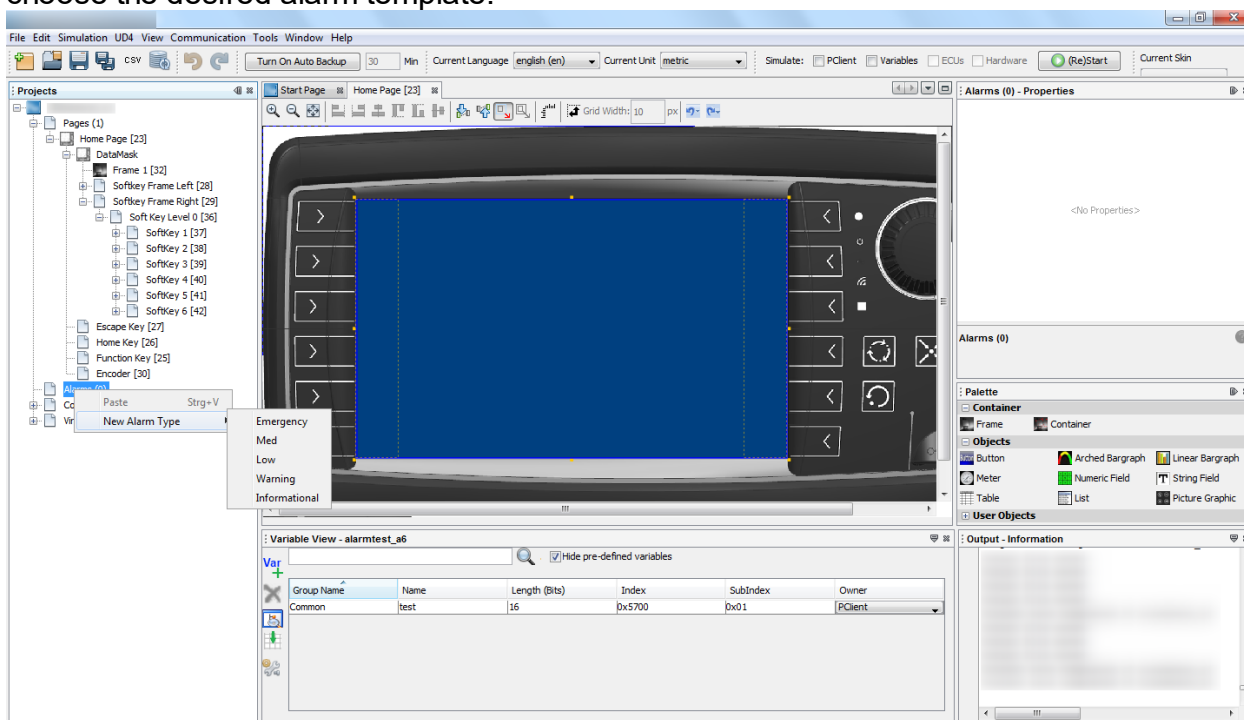
Then a dialog window appears where you have to enter a unique number (in decimal form) for that alarm. Numbers that were already used are displayed in the lower part of that dialog.
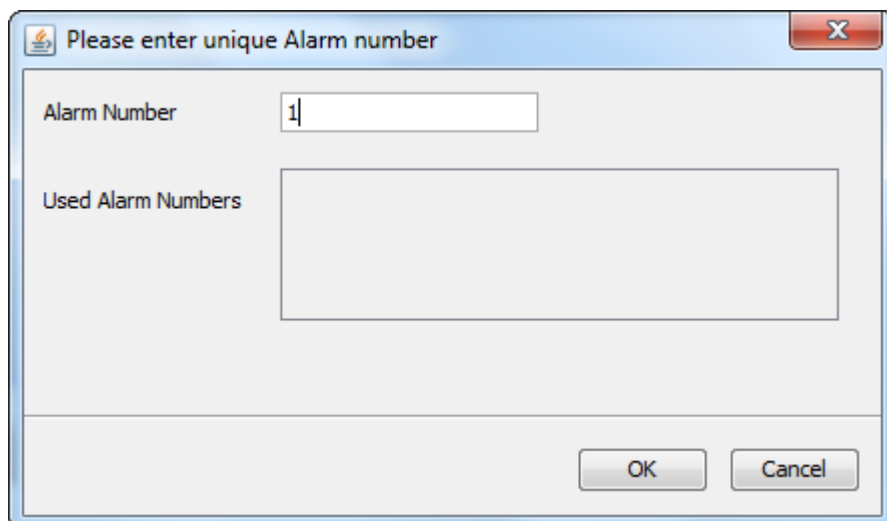
Press OK to create the alarm. The alarm will be listed in the project tree under "Alarms()".
To edit it, double click on the alarm in the project tree. A tab will be opened where you can
create a layout for the alarm similar to a page layout.

Activating and Clearing Alarms

Alarms can be activated by setting the variable **@AlarmShow**. To activate an alarm, the
variable value has to be set with 0x8000 **OR** alarm number. In other words, you have to use
an inclusive logical disjunction. The alarm number has to be used in hexadecimal form
here.
Example: To set an alarm with the number **15**, you have to set the variable **@AlarmShow**
to 0x8000 OR 0x000F = **0x800F**
To clear an alarm, simply set the variable **@AlarmShow** to the number of the alarm.
Example: To clear an alarm with the number 45, you have to set the variable
@AlarmShow to **0x002D**
Activating and clearing alarms can be done over the CAN bus by including the variable
@AlarmShow in a receive mapping, by calculations in a Javascript or by pressing a
button and using the [SetValue]⁴⁴⁰ action.
Additionally, Warning and Informational alarms can also be cleared by pressing the ESC
key.
Informational alarms can also be configured to close automatically after a certain time.

Active alarms will stay hidden behind other applications such as PDF reader on i.MX 6
based devices.

8.2   Day/Night Mode

With this feature it is possible to switch the [theme of objects]⁵⁸⁰ at run time. The behavior is
similar to modern navigation apps that switch their UI for day or night mode. This has the
advantage that the display shows e.g. only dark colored objects at night to protect the eyes of
the user.

Day/Night Mode can be activated for all visual objects in the palette. It is disabled by default
and has to be enabled for each object or for all objects through the project property settings. If
Day/Night Mode is disabled, the object uses the Day theme only.
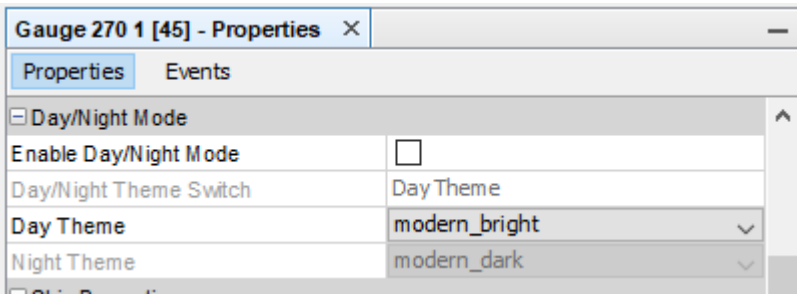
To enable Day/Night Mode for an object, the property "Day/Night Mode" has to be activated. After that you can choose a theme for day and one for night (properties "Day Theme" and "Night Theme".

Per default, the theme "modern_bright" will be used as the day theme and "modern_dark" as the night theme.
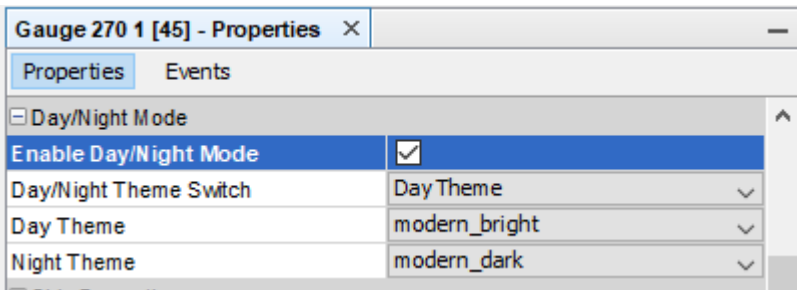
With the property "Day/Night Theme Switch" it is possible to preview the Day and Night view in the editor window to check if the object is configured properly.
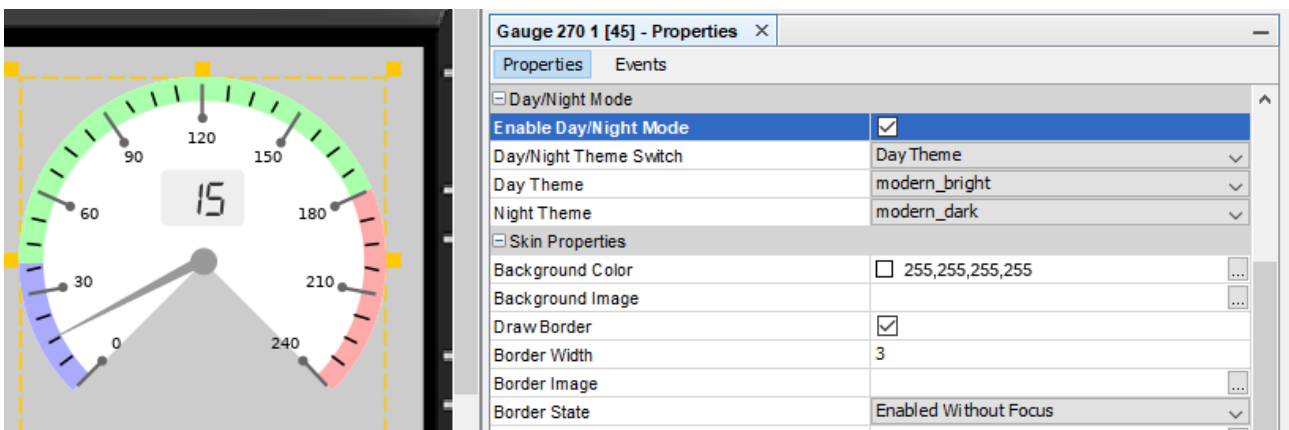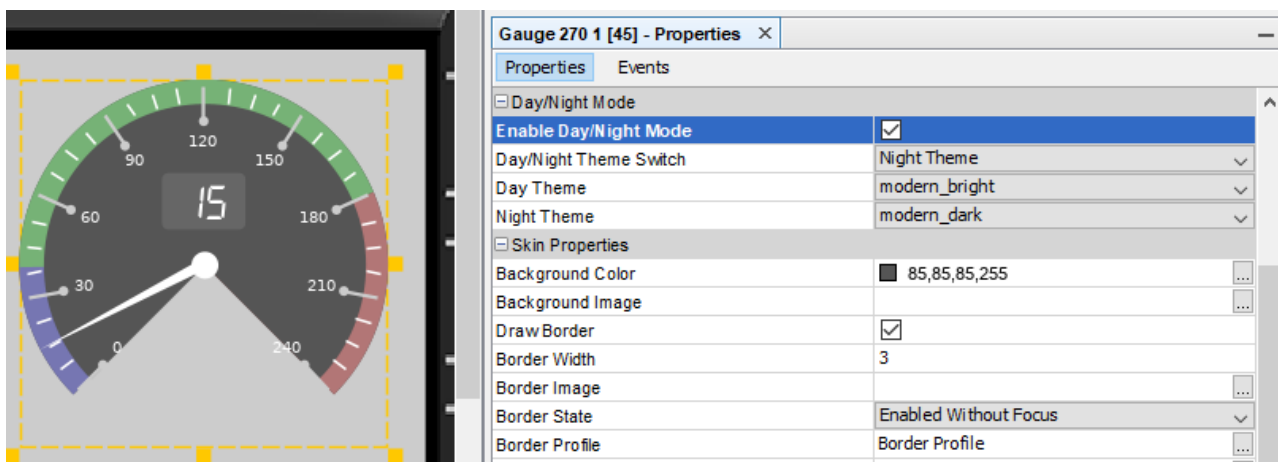
To show some details of the new theme handling, a new theme will be created, called "new_dark", as a duplicate of "modern_dark", with the background color of the 270° gauge changed to dark blue. The preview of the new look is shown in the dialog.
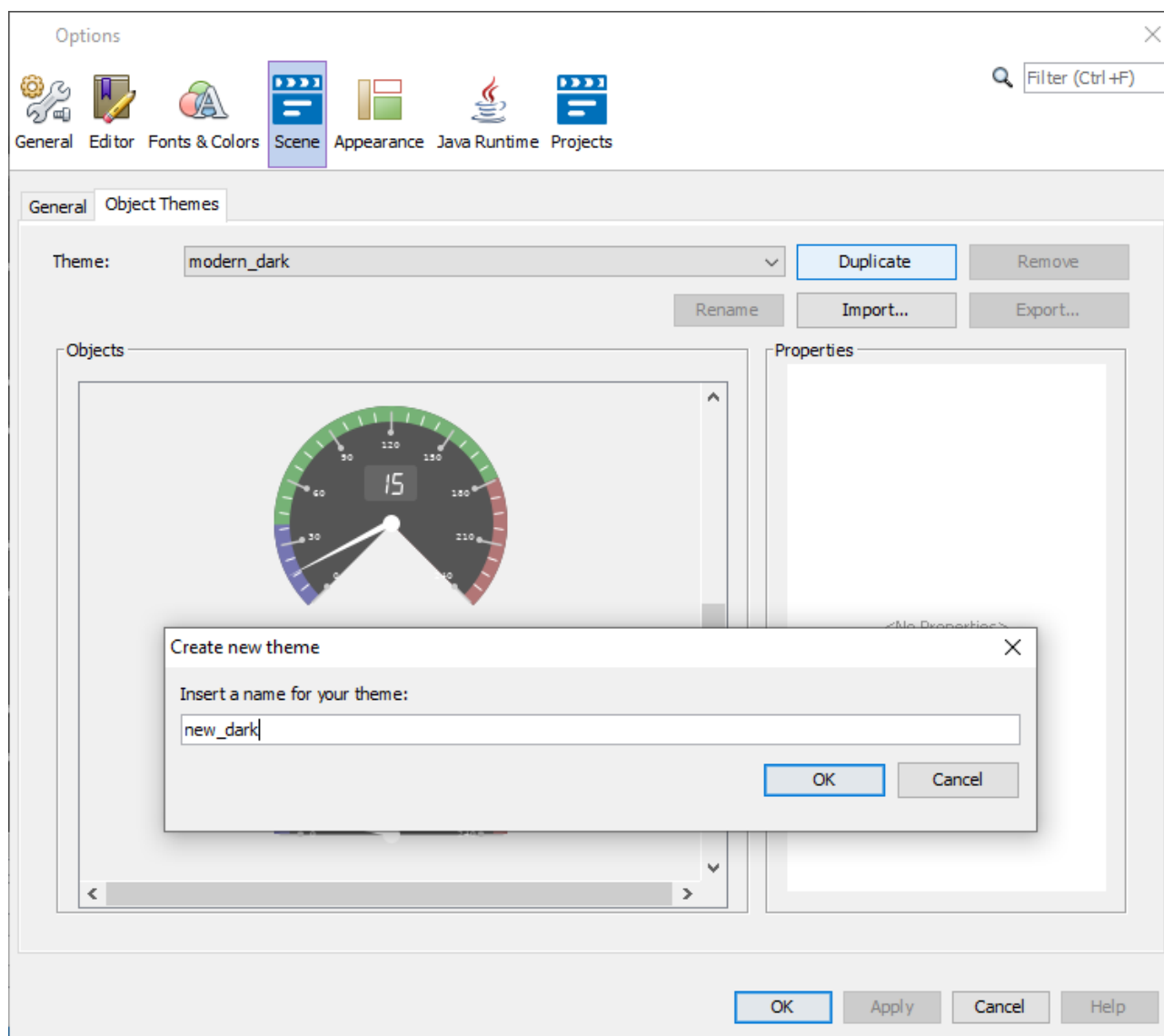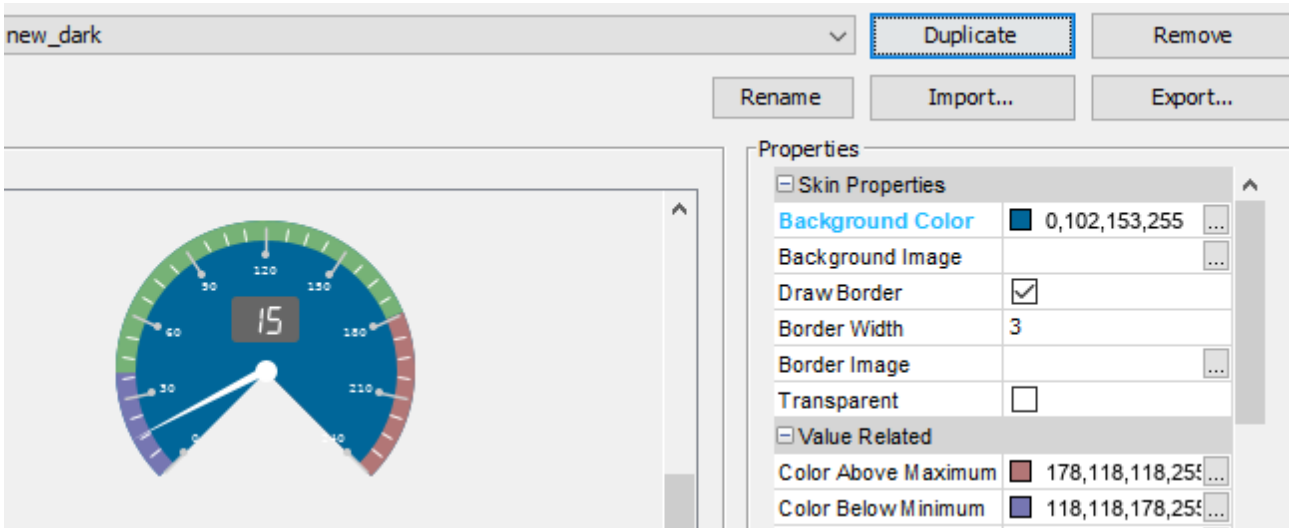
The theme can now be chosen in the object properties for "Day Theme" and/or "Night Theme".
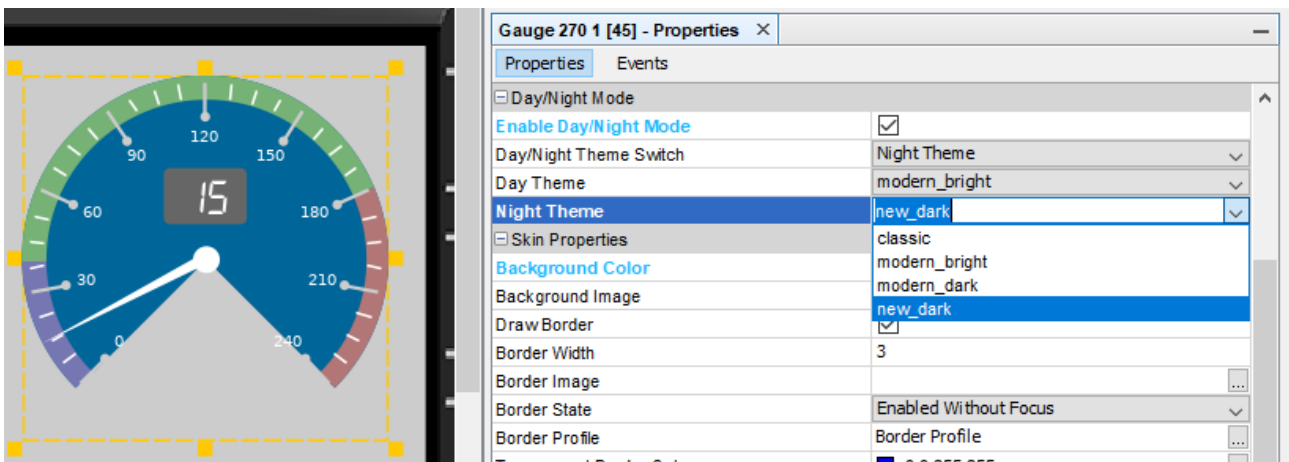
Note: Changing the theme of the object overwrites all custom property settings made previously (only for theme properties)!

It is of course possible to change theme properties (i.e. visual properties that are configurable in themes) of single objects with a normal property configuration for the object.
If Day/Night Mode is enabled, the properties can be set separately for Day and/or Night Mode. The property will be saved for the mode currently selected in the property "Day/Night Theme Switch".
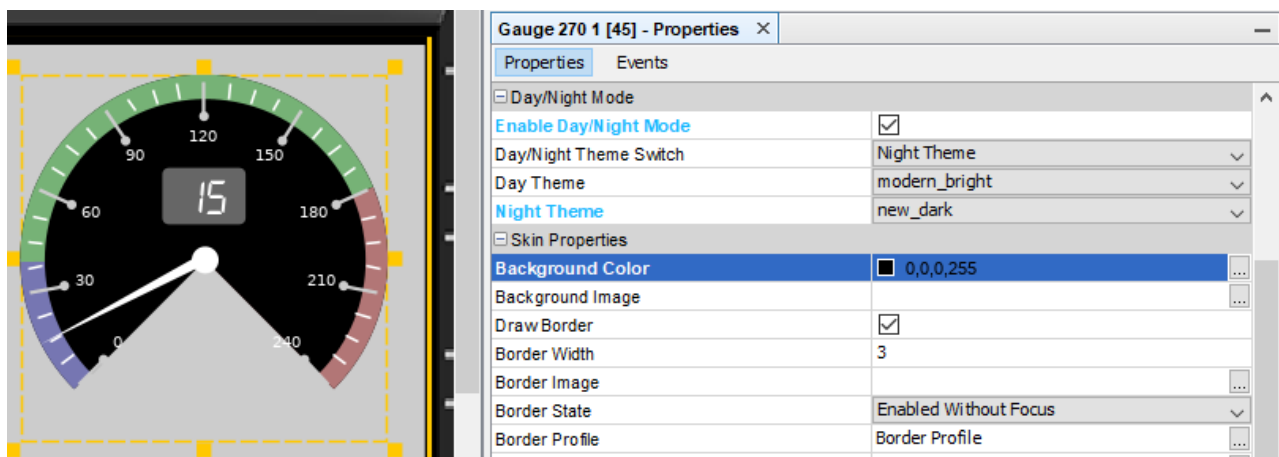
At run time, the day / night mode can be switched with the variable @CurrentThemeIndex[174]

Project Properties Settings
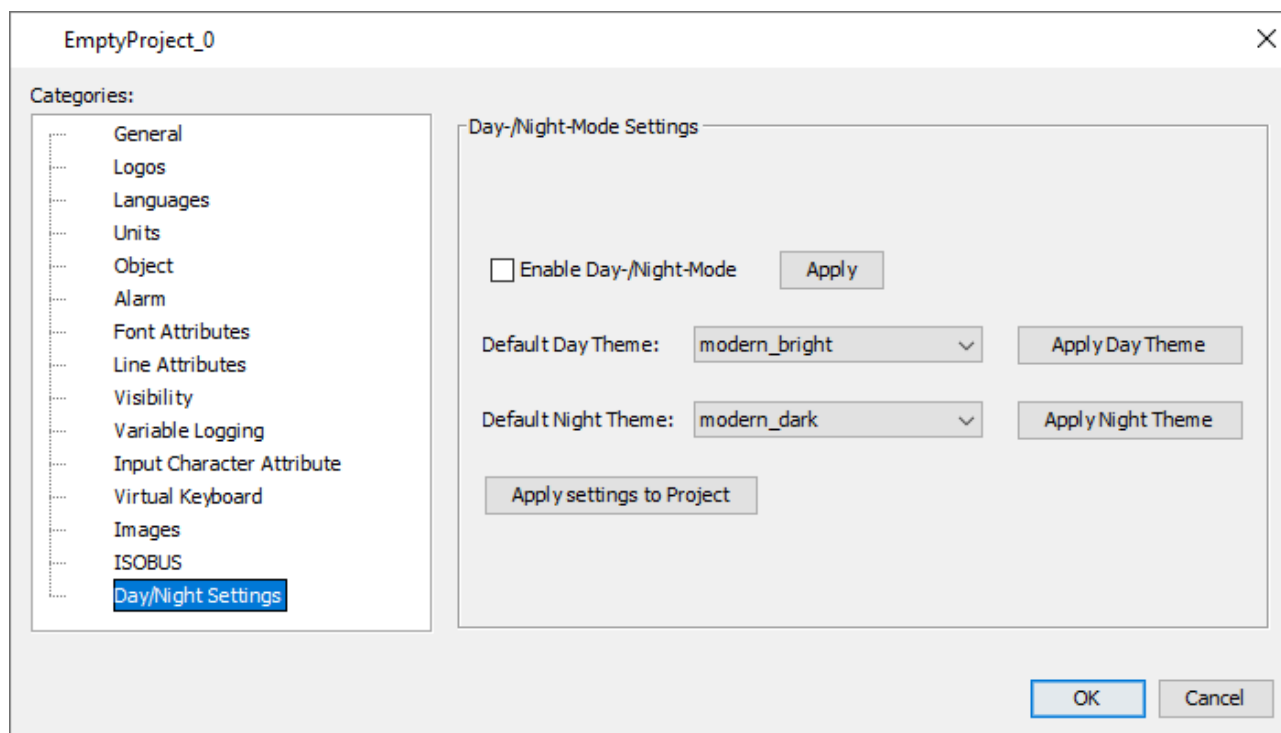
The following settings can be made in this dialog:

Enable Day/Night Mode (for newly created objects)
When this box is checked, new objects will have Day/Night Mode enabled by default.

Apply "Enable Day/Night Mode" to Project
When this button is pressed, ALL existing objects in the project will have Day/Night Mode enabled, with their current Day and Night Theme set in their properties.

Default Day Theme
This selection will be the default Day Theme for newly created objects

Apply "Default Day Theme" to Project
When this button is pressed, ALL existing objects in the project that have Day/Night Mode enabled will have the current default Day Theme set as their day theme.
Note: This overwrites all theme properties of all existing objects!

Default Night Theme
This selection will be the default Night Theme for newly created objects

Apply "Default Night Theme" to Project
When this button is pressed, ALL existing objects in the project that have Day/Night Mode enabled will have the current default Night Theme set as their night theme.
Note: This overwrites all theme properties of all existing objects!

Apply settings to Project
When this button is pressed, all three "Apply to Project" actions described above are executed.

8.3   ISOBUS Application

For the devices OPUS A6 Generation 2, OPUS A8, B3, X18 and X40, a special ISOBUS application can be created.
This type of application can only be installed on devices with an OPUS ISOBUS solution license.
For the OPUS devices, the article number needs to have the text UTS, UTM or UTL included.
Please note that the device software for these devices is not included in the OPUS Projektor.
Contact your vendor for information where to get the newest software for these devices.

To create an ISOBUS application, choose the according Panel type and activate the check box **Use ISOBUS Application**.
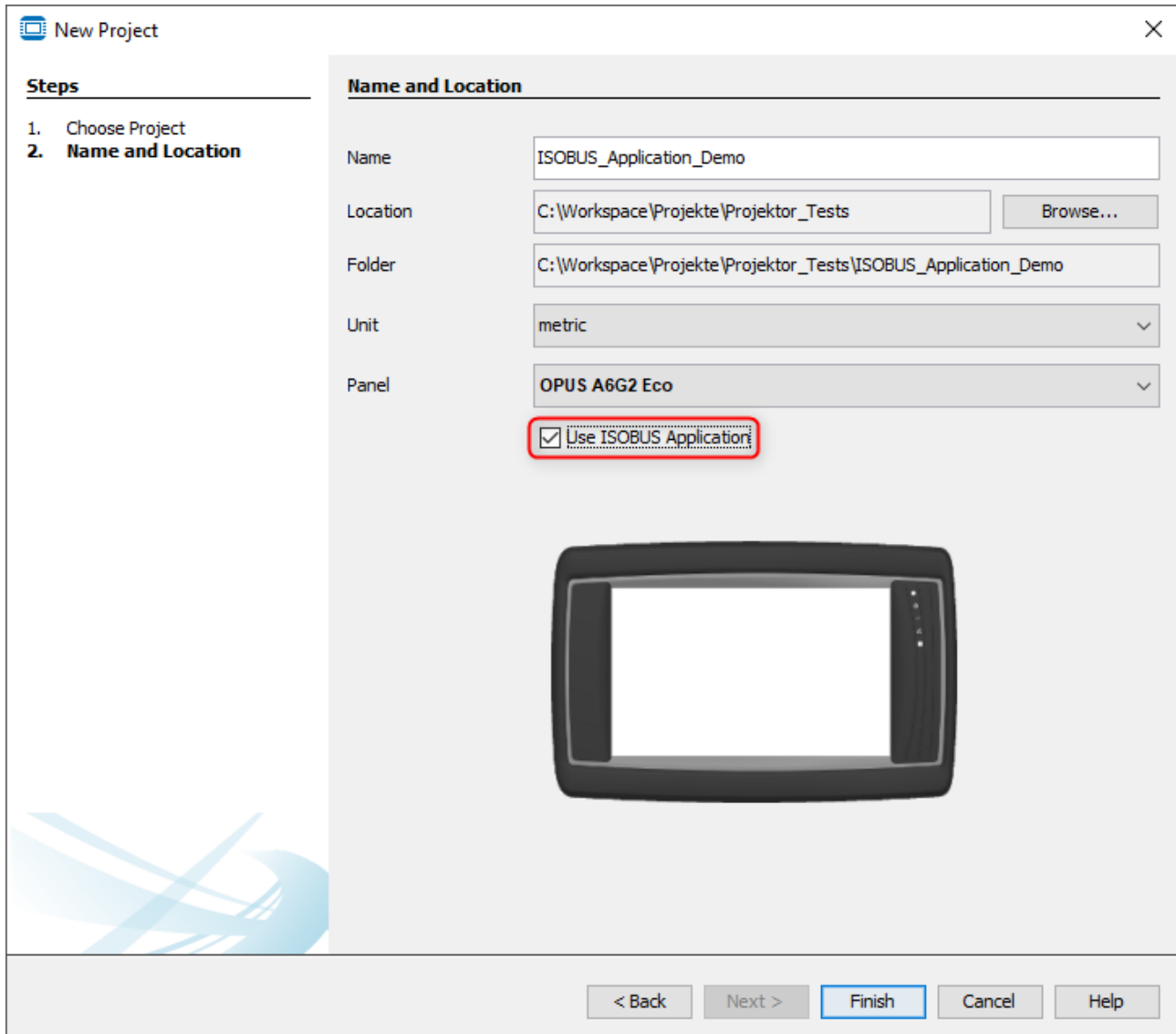
Please note that only Landscape versions can be used as ISOBUS applications. Portrait projects cannot run as an ISOBUS application.
For the X18 and X40, the ISOBUS checkbox is checked and cannot be un-checked.

For ISOBUS projects, a MiniView Window of the page is created automatically.
For OPUS A8 and X40, these MiniView pages are shown next to the actual page in the editor.

The large portion of the screen will be the normal page, as it will be integrated as a main screen in the OPUS ISOBUS solution.
The MiniView Window will be used as the small version of the page.
By default, the MiniView will be scaled automatically from the big page.

It is also possible to create a custom MiniView page. This is done by right-clicking the Miniview Pages entry in the project tree and clicking New Miniview Page.
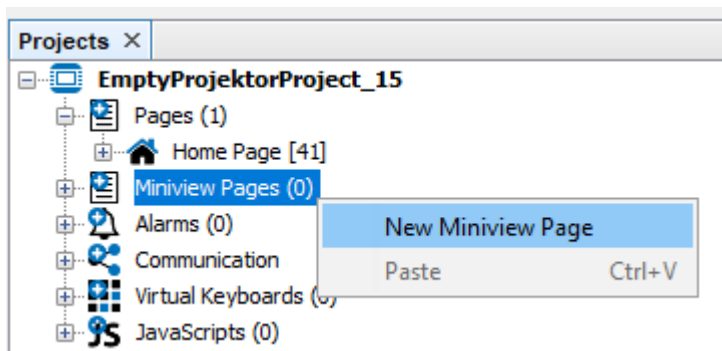
The resulting page only has the small MiniView size.

The mini page can be designed in any way like a big page.
Every normal page in the project has a Property Miniview Page ID.

This property can be set to the ID number of any MiniView page.
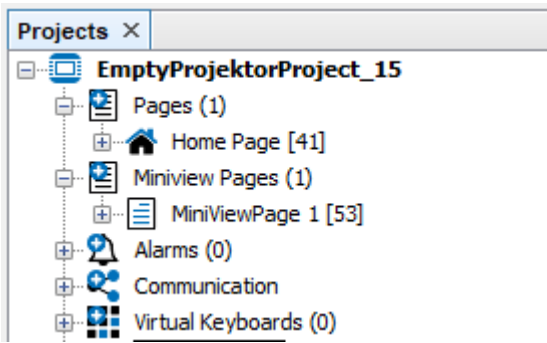
Then the MiniView Page will be shown in the normal page view.

For B3, X18 and A6 Generation 2 all of this is the same, only the MiniView pages are not shown in the page editor of the main pages.

8.4   Object Themes

General

Object themes are pre-defined or user-defined sets of properties for palette objects.
Five object themes are pre-defined:

Stunning Bright & Stunning Dark
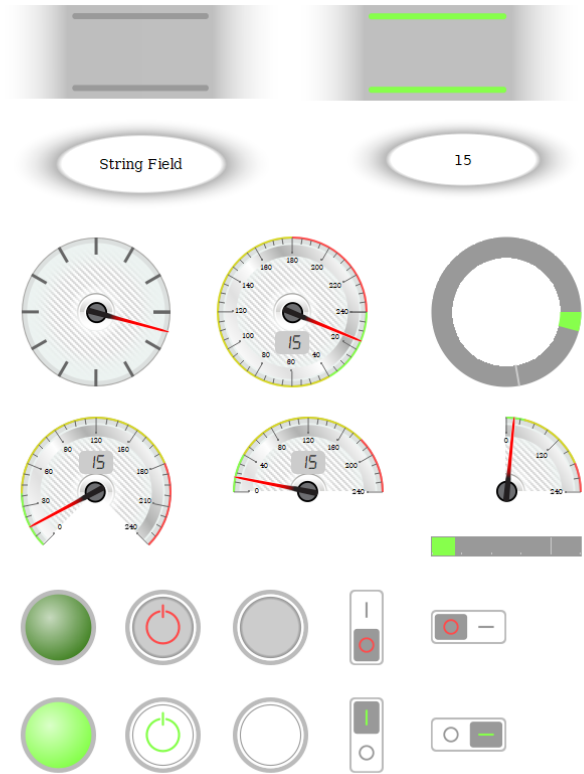
Modern bright
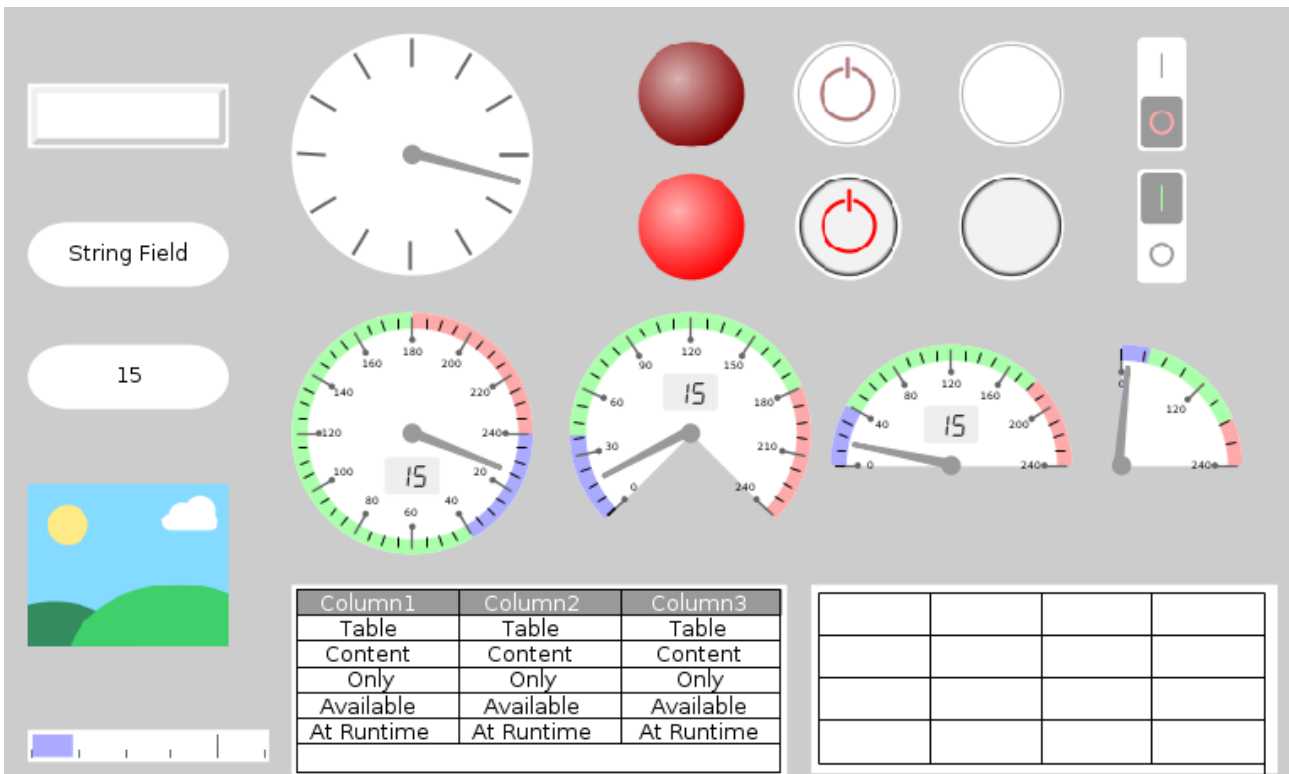
Modern dark
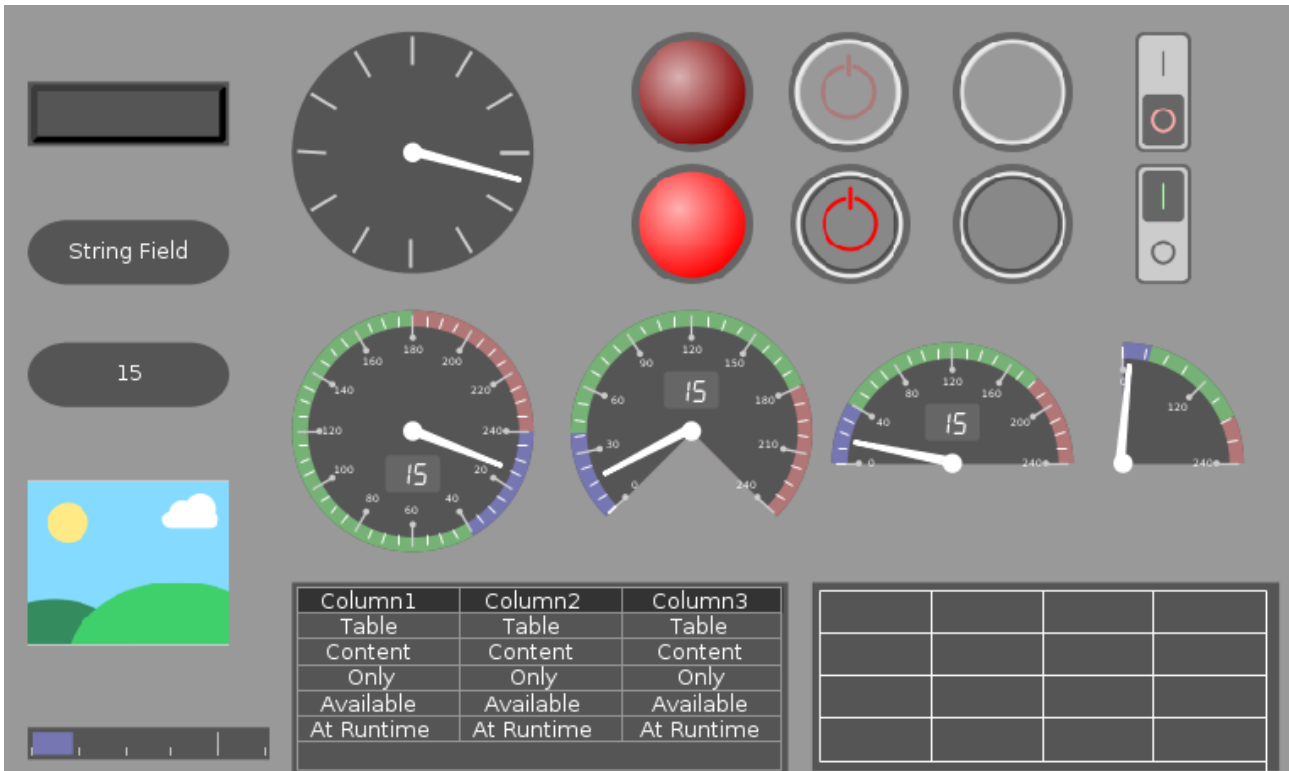
Classic
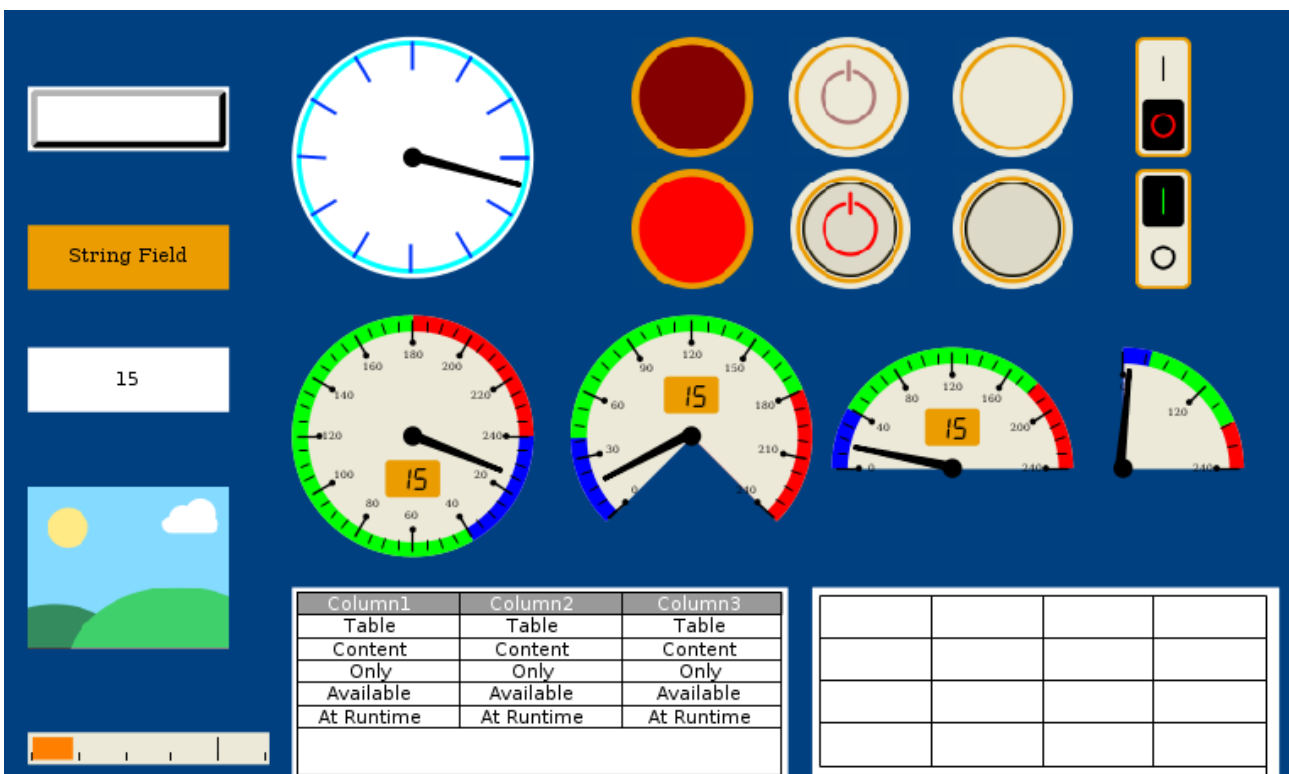
For each palette object, certain properties are "theme-dependent", i.e. they will be used when a new object with that theme is created or when the theme is applied to an object or to a whole project.

To see which properties are changed by a theme, go to the dialog reached over the menu *Options -> Scene -> Object Themes*, choose a theme and select one of the objects. In the "Properties" window on the right side of the dialog, the affected properties and their settings for the selected theme are displayed.
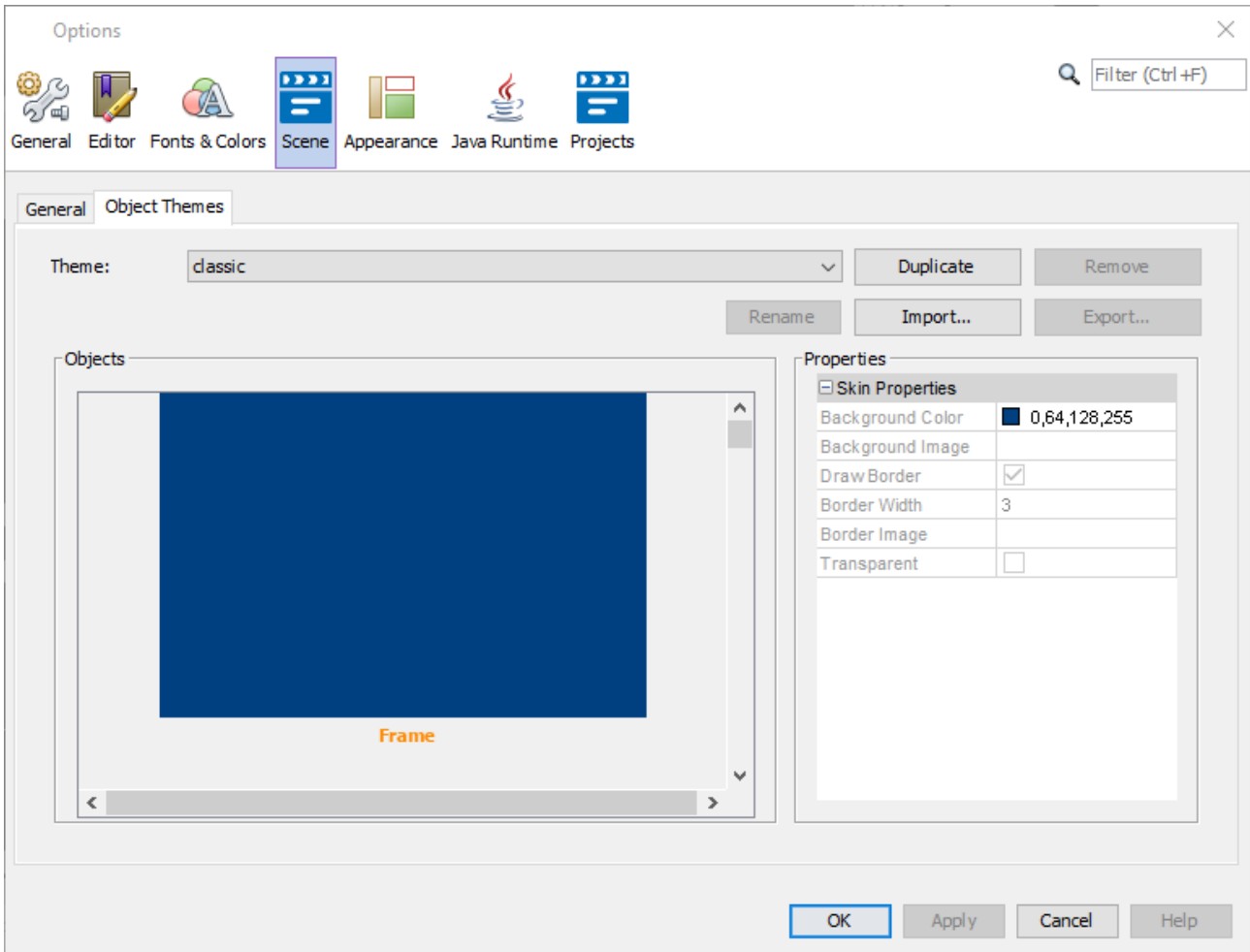
Choosing themes

Themes can be chosen

- for the currently selected object in the scene, by changing the Day or Night Theme property of the object, depending on the setting of the Day/Night Theme[571] Switch

- as the standard for new objects, by changing the "Default Day Theme" "Default Day Theme" setting in the project properties.
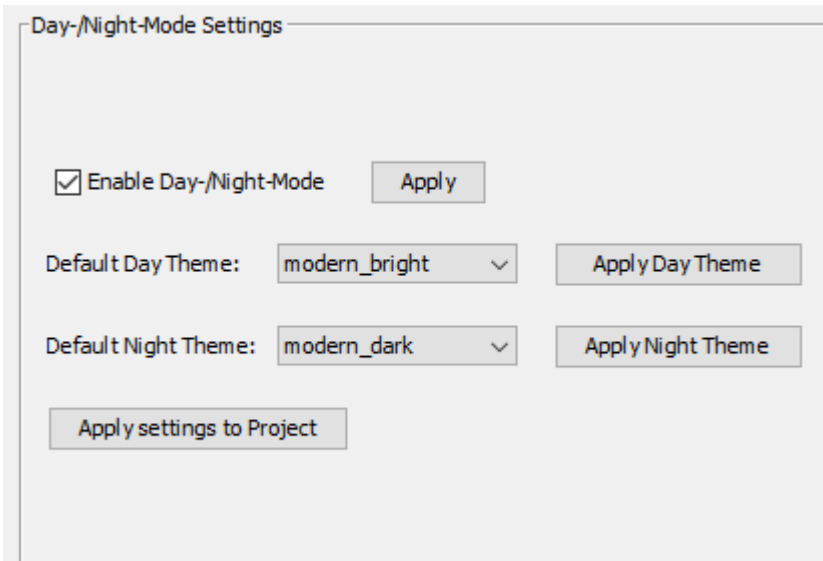
- for an existing project, by choosing a theme in the menu Tools -> Options and then Scene -> Object Themes in the "Theme:" dropdown menu, clicking *Apply* and selecting *Yes* in the confirmation dialog.
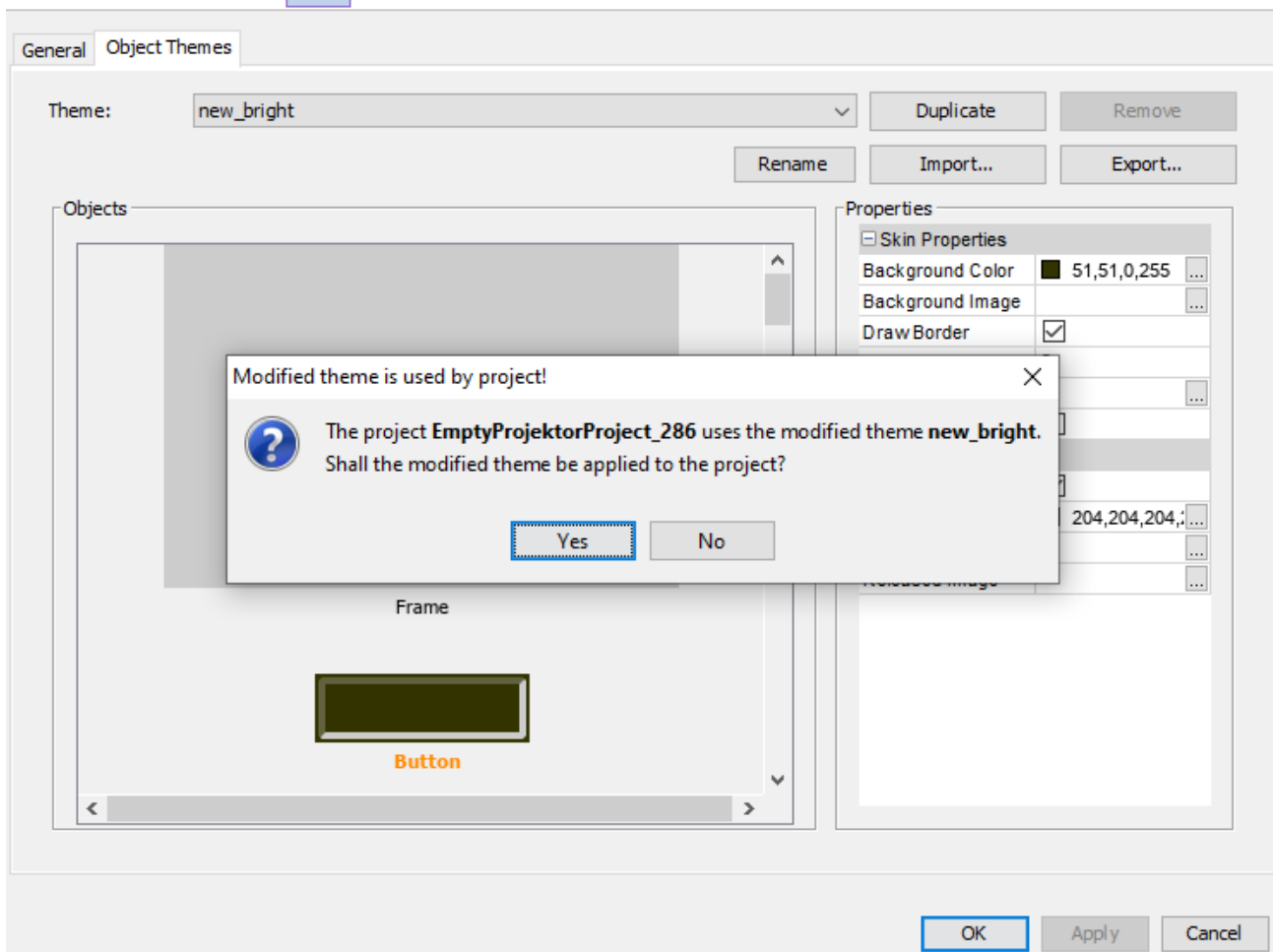
Editing themes

The pre-defined themes cannot be changed.

However, duplicates of these themes can be made. These duplicates can then be modified by the user.
In the dialog reached over the menu *Tools -> Options and then Scene -> Object Themes* an existing theme can be duplicated by selecting the theme and clicking "Duplicate".
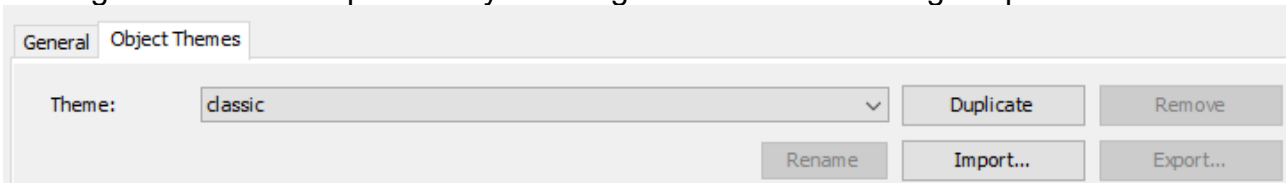
A new name needs to be given to the new theme. By clicking OK, the new theme will be created. The name needs to be unique compared to existing theme names.
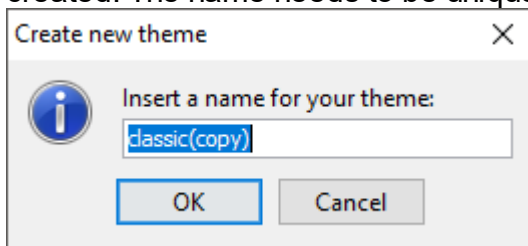
Theme-dependent properties of each object can then be edited in the "Properties" window on the right side of the dialog. To apply the changes, click the "Apply" button.
The theme dependent properties are marked as such in the object reference[412].

Now it gets a bit complicated if a custom theme, in our example "new_dark", which is already used on objects, is changed in the Options dialog.
After you changed any of the theme settings and press "Apply" or "OK", the following dialog appears.
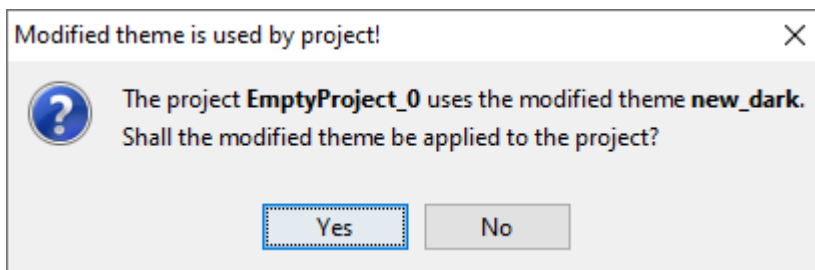
No - The changes will still be saved to the theme, but they won't be applied to the project.
This means that the project theme "new_dark" is now different than the "global" theme "new_dark".
The consequence of this is that now the global "new_dark" theme is not available anymore in this project.
Instead a theme called "new_dark_<project_name>" will be displayed. The theme will still have the same name, the attached project name is just to differentiate it from the global theme.
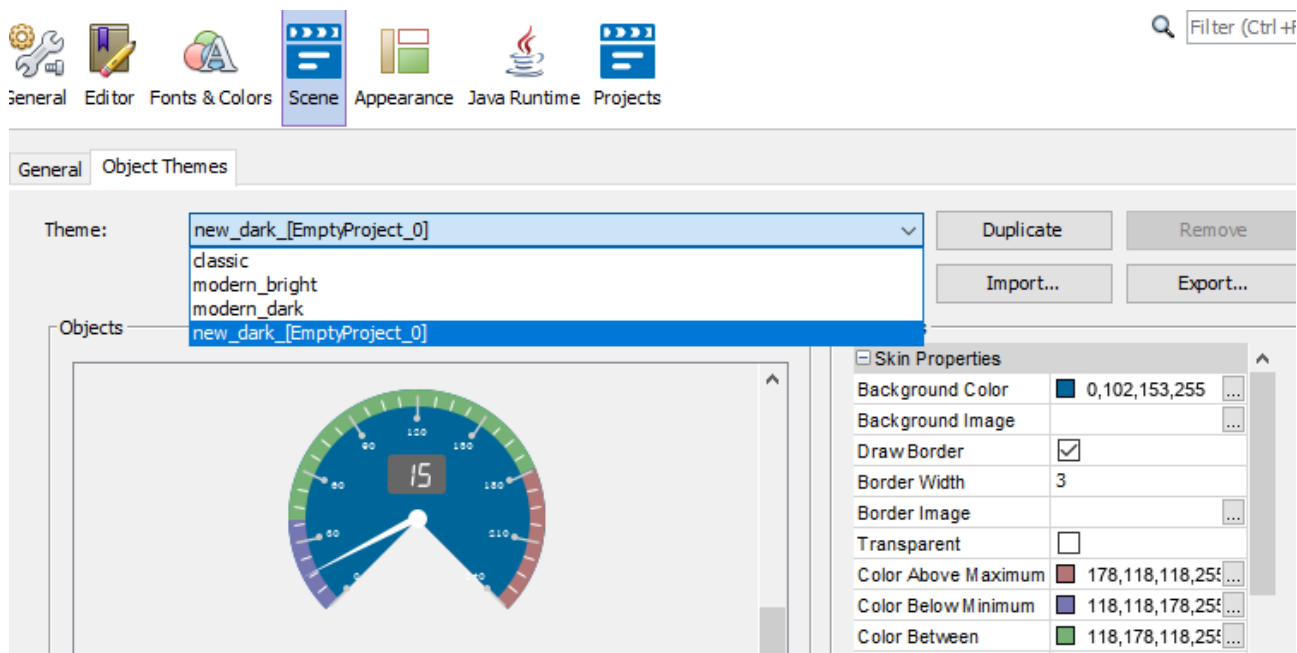
This theme is only visible with the project selected. Changes to this theme will only be applied to the project, and all changes to the project theme will directly be applied to the project when clicking OK or Apply.
To use this theme globally it can be duplicated and the duplicate becomes a global theme and can be used in other projects.

Yes - The modified theme will be copied to your project and the changes you made will directly be applied to your project.
In this case the project still uses the global project.

Note: The updated theme will only be applied to all objects which already use the theme. Object properties that were changed will not be overwritten.

It is also possible to put the settings from an existing object into a theme. To do this, right-click any graphical object with theme-dependent properties, select Save Object To Theme and select the custom theme you want to save the properties into.
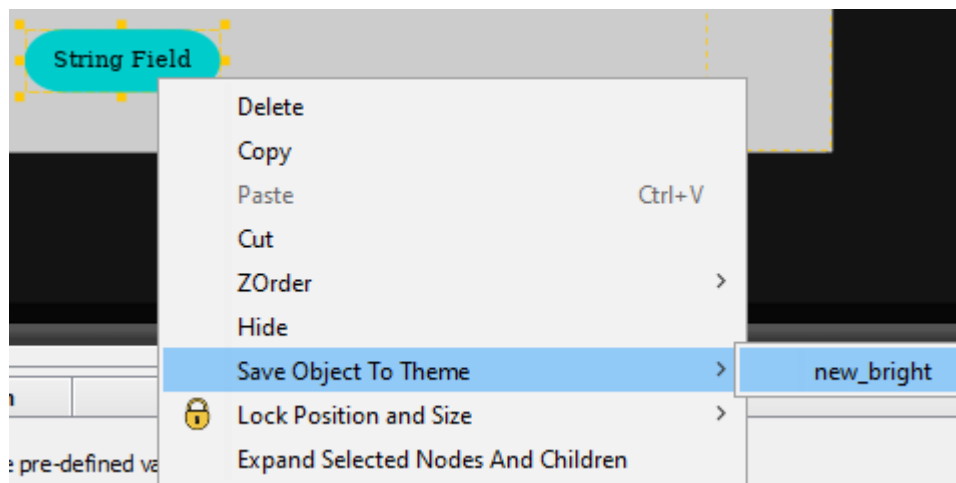
Please note that only the theme-dependent properties will be copied into the theme.

FAQ

I can't see the global theme <name> anymore, only <name>_<project_name>.

When a project uses a theme that has the same name as a global theme, and they are different, the global theme is not visible anymore when the project with the custom theme is selected. Instead, the project theme is displayed.

How can I get rid of this theme <name>_<project_name>?

This project theme is displayed if a theme with the same name, but with different settings, is already existing in the global theme folder.
This theme is only displayed when the project containing it is selected. If you select a different project or close that one, the global theme will be displayed.
If the project theme is modified in such a way that it becomes the same as the global project, it will also be displayed as the global theme again.

Where are the themes stored?

The global themes are stored in the application folder, in \application\extracted_components\object_themes.
Any theme in a project will be copied to this folder when it is loaded in the program, unless there is a theme existing already with the same name. In this case the global theme cannot be edited or selected, instead the project theme will be shown with the name <name>_<project_name>.

Themes used by projects are copied to the project folder, into the sub-folder \themes.
Additionally, the themes are also saved in the terminal_files sub folder for the use on the device

There is a themes folder in my project folder even though I don't use day / night mode.

When a project uses a theme, a copy of the theme is put in the subfolder \themes in the project folder, so it will be available on the device.
Even if no theme is actively used and day / night mode is not enabled, one theme will always be in that folder, the default day theme (by default, modern_bright).

Can I delete themes I don't need (anymore)?

Except for the pre-defined themes classic, modern_dark and modern_bright, themes can be deleted in the Options dialog.
If a project is opened and it contains a theme that was deleted, it will be re-created.

Custom themes can also be exported and imported.
To export a theme, select a custom theme and click "Export theme...". Then choose a location where the theme should be saved.
The exported file is a zip which contains the theme. The file name can be changed, but only alphanumerical characters and _ can be used.
To import a theme, click "Import theme..." and choose a valid theme file.
The theme name will be derived from the theme file name.

To remove a custom theme, click the Remove button.

Themes are not project-dependent, they can be used in all projects. They are saved in the user data folder.

8.5 Visibility

Tutorial video about visibility
https://www.youtube.com/watch?v=77RNxko1HvQ

Example projects: Download this example to see how Visibility is used.

There are several reasons for using Visibility.
The Visibility can be used...

- to hide symbols or information on the device, until for example an error occurs.
- in case you would like to use the same project for different systems (for example, for machines with different equipment).
- to define different work flows depending on the access rights. For example a normal user view and an admin user view. So you can hide/show buttons or pages for specific user groups.

New simplification of the Visibility system

The Visibility system has been extended to make things easier. Now every variable can be used as a visibility variable for objects.

To do this, drag & drop[43] a variable from the Variable View[146] onto the object. In the context menu, select the object first, and in the second level, select "Use as Visibility variable".
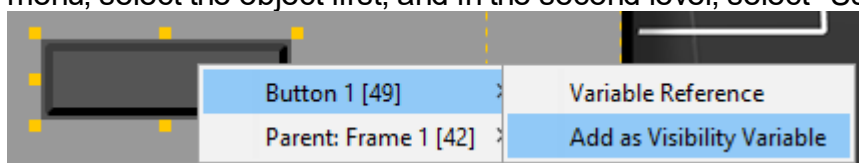
This activates the property Visibility for the object and sets the variable as the Visibility variable for the object.

Alternatively, the Visibility variable can be set through the properties.
Select the object and activate Visibility in the properties.
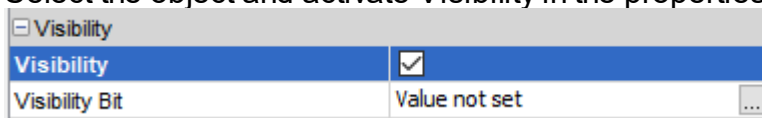
Then open the Visibility dialog by pressing the three-dot button in the Visibility Bit property.
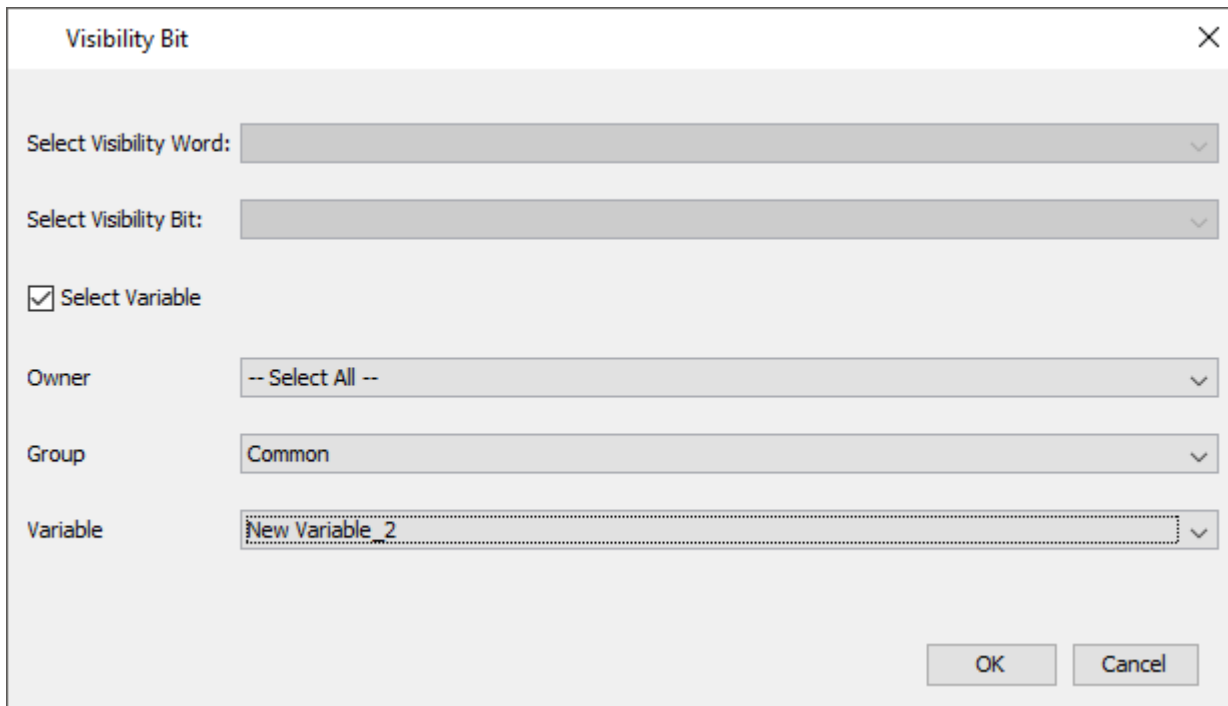
First, check the check box next to "Select Variable". Then choose a variable below by filtering by owner and/or group. After choosing the variable you want to use, click OK to save the setting. The variable will be displayed in the Visibility Bit property.
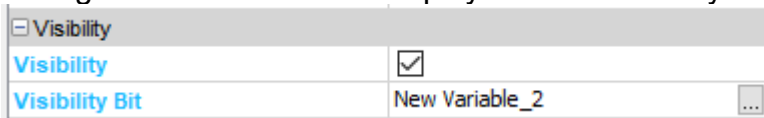
Setting the object visible / invisible is very easy:

If the variable is 0, the object is invisible
If the variable has ANY other value than that, the object is visible.

If variables are initialized as invalid, their default value will be used to determine the visibility status, i.e. if a variable is configured to be initialized as invalid and with default value 5, the according object will still be visible, even when the variable has not been set to a valid value.

Note that a variable can be used for several objects, i.e. one variable can make any number of objects visible / invisible at the same time.
Note that String variables cannot be used as visibility variables. Any other data type that is offered can be used.
Also note that the pre-defined Visibility variables (see below) can't be used for the new visibility system.

Alternatively to the new system described above, Visibility variables are provided by the program. Every project contains a set of 64 Visibility variables. All of them have 16 Visibility bits, which can be assigned to any graphical object(s). Each bit is a flag to hide or show graphical objects[412] by switching between the value **0** or **1**.
The value range of the Visibility variables is from **0x0000** to **0xFFFF**. The default value is set to **0xFFFF**, which means that all corresponding object will be visible.

How to configure Visibility

Before the Visibility can be used, you have to define the Visibility variables and its bits. Therefore please navigate to main menu tab **[File][Project Properties**[76]**]** and select

category **Visibility** ① .
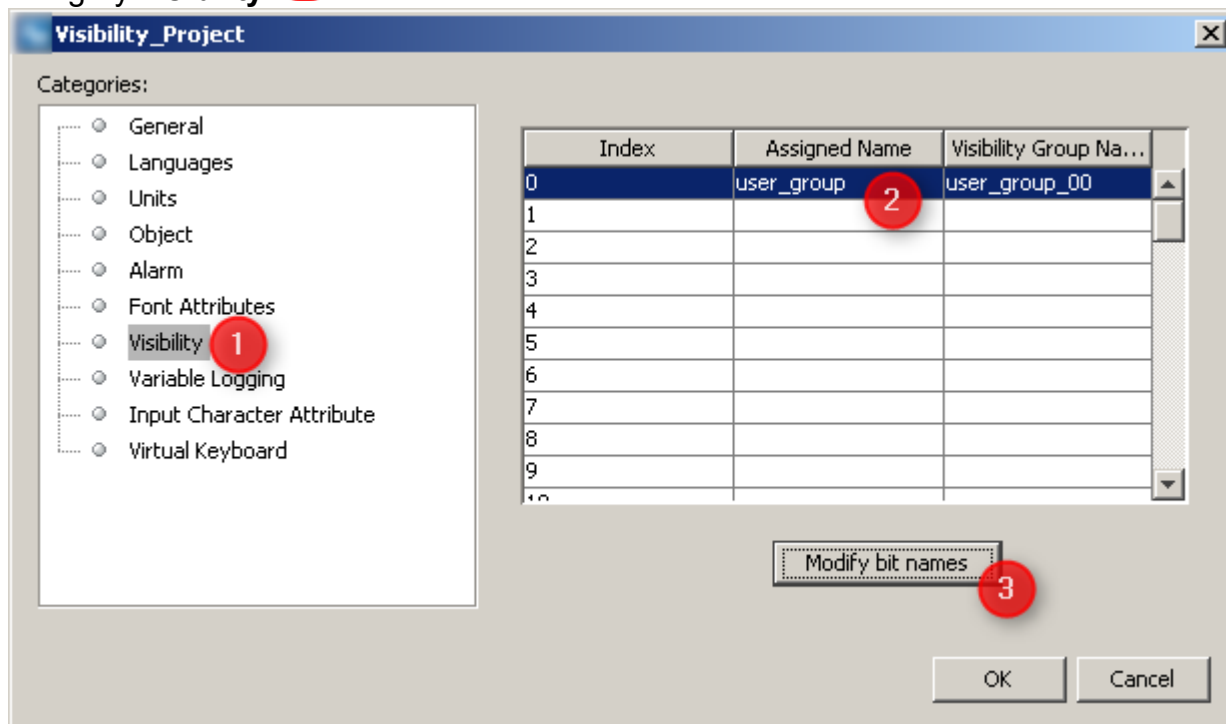
This dialog provides the fields to add new Visibility variables. The 'Index' of the variables defines what internal Visibility variable should be used. So **@Visibility00** variable could be

set with Index 0 by double press of the column **'Assigned Name'** ② .
Afterwards you can configure the bits for each Visibility variable by clicking on **'Modify bit**

names' ③ . Therefore a new dialog called **'Modify bit names'** will be opened:
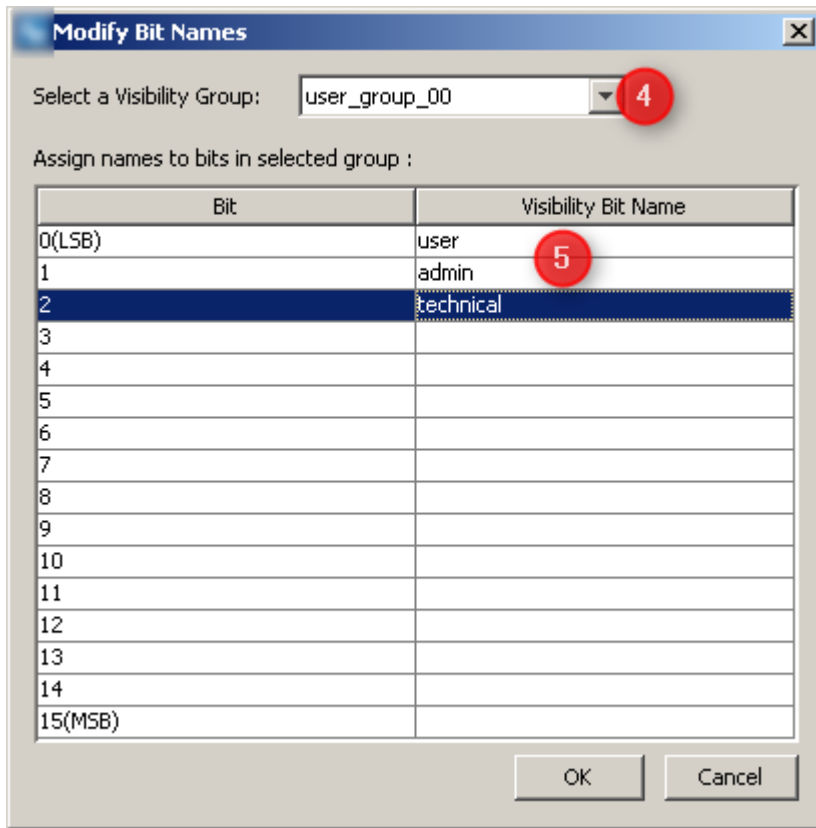
Please select the Visibility variable which you want to configure ④. Now you can define

for each of the 16 bis (Bit 0 - Bit 15) the correspondent visibility name ⑤.
After defining names for the Visibility bits please confirm both dialogs with 'OK'.

Now the Visibility variables is configured properly and it can be used in the project.

NOTE:
Each entry in both tables should be committed with **'Enter'** key.

Structure of Visibility Variables

To show the object corresponding to the specific bit of the Visibility variable, this bit should be set.

@Visibility00

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name														technical	admin	user
Value																
Decimal	32768	16384	8192	4096	2048	1024	512	256	128	64	32	16	8	4	2	1

If you want to show all objects which are corresponding with the first bit of the Visibility variable *(user)*, the value of the variable *@Visibility00* should be set to *'1'* (value in decimal)

@Visibility00

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name																user
Value																X
Decimal																1

If you want to show all objects which are corresponding with the 3rd bit of the Visibility variable (*technical*), the value of the variable *@Visibility00* should be set to *'4'* (value in decimal)

@Visibility00

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name														technical		
Value														X		
Decimal														4		

If you want to show all objects which are corresponding from the 1st - 3rd bit of the Visibility variable(*user*, *admin* and *technical*), the value of the variable *@Visibility00* should be set to *'7'* (value in decimal)

@Visibility00

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name														technical	admin	user
Value														X	X	X
Decimal														4	2	1

Recommendation: To make handling of visibility variables easier, it is recommended to start by using only the Bit 0 of the visibility variables. 64 visibility switches are enough for most usecases. This avoids difficult bit operations and makes things very easy: 0 - invisible, 1 - visible.

Use Visibility Variables in the Project

To make an graphical object Visibility depended please do following steps:

- Click on a graphical object e.g. **'Text Field**[504]**'** ①

- Navigate to **'Properties'**[415] window ②
  - Enable checkbox at property **'Visibility'**
  - Click on 'Value not set' at property **'Visibility Bit'**

- Select Visibility variable and Visibility bit at the dialog **'Visibility Bit'** ③
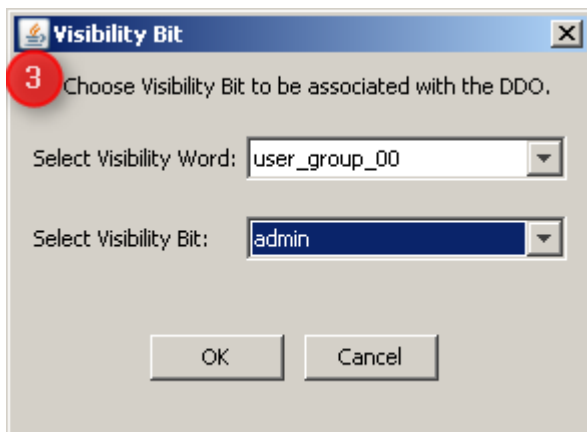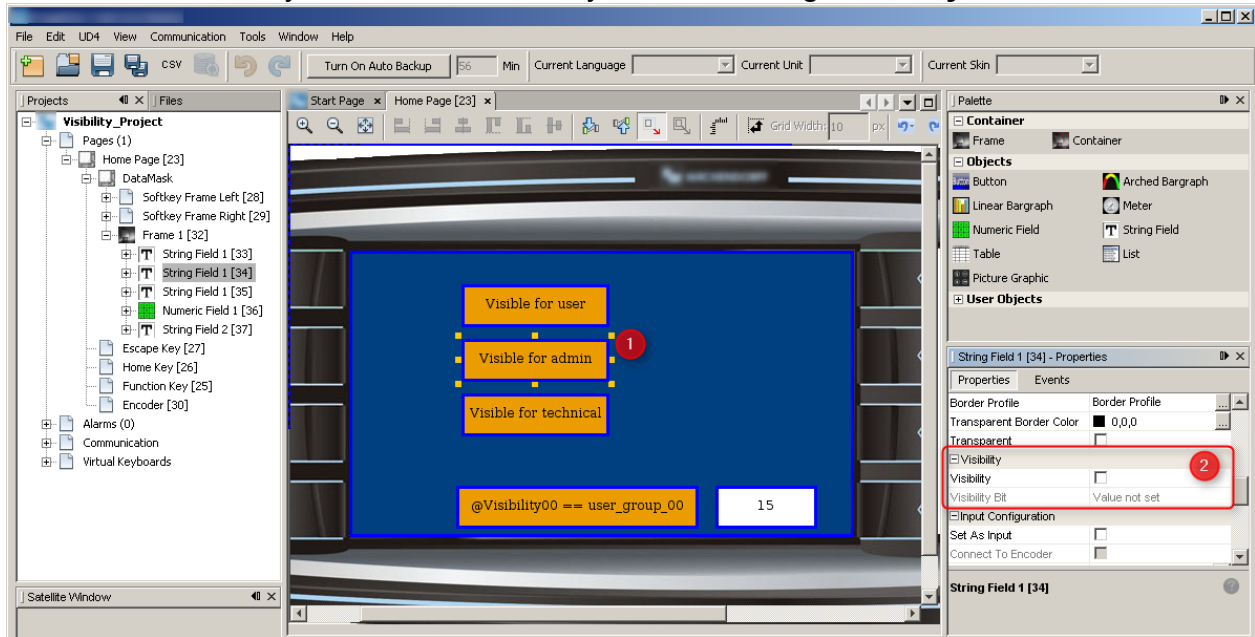
At the end following settings should be set:
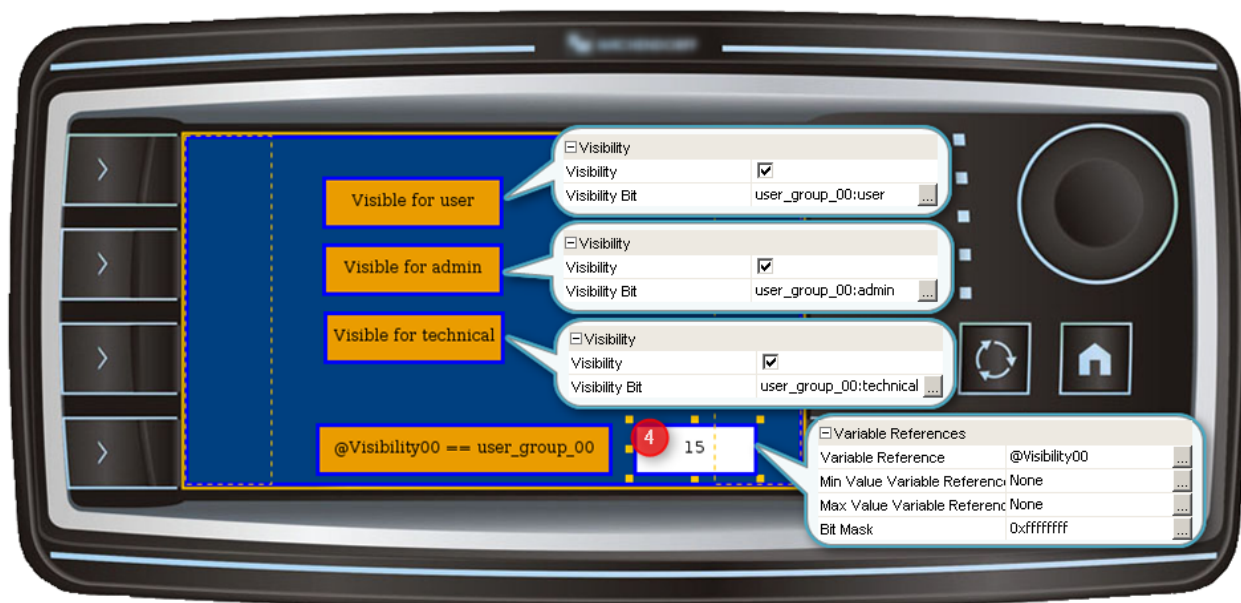
Now the Visibility variable **@Visibility00** can be updated by setting the value of the

'Numeric Field' [495] (4).

NOTE for J939 PGNs:
Be careful with adding visibility variables to receive PGNs.
To make sure all values are forwarded to the Visibility variable, enable the option Forward Don't Care for the variable in the J1939 mapping dialog, otherwise a value of all 1's will be ignored.

Simulations

9 Simulations

Two types of simulations are available (click the links for details):
- [PClient Simulation](#)[598]
- [Variable Simulation](#)[601]

Additionally [Simulation Settings](#)[596] can be made

 Tutorial video about the PClient Simulation

https://www.youtube.com/watch?v=rzgrpzZZ9tI

9.1 Simulation Settings

General

The settings dialog can be opened via the menu Simulation => Simulation Settings
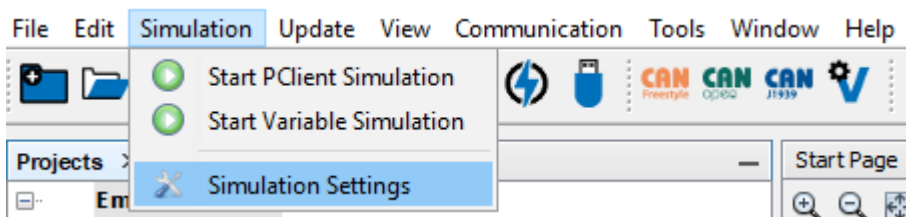
The following simulation settings can be made:
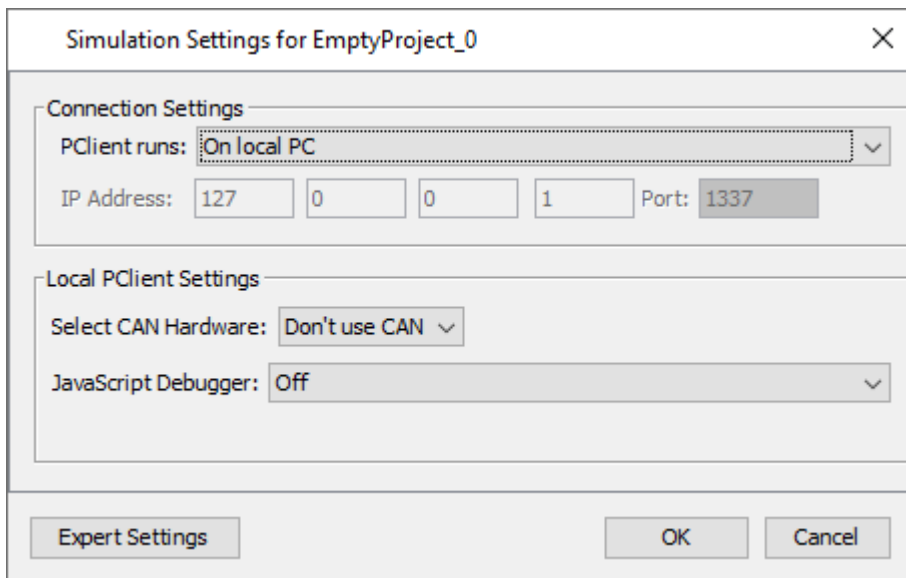
Connection Settings

Connection settings only need to be adjusted if the PClient shall not be run as a simulation on the local PC. If it is run as simulation don't change anything.

If a connection to a running PClient on a device via ethernet shall be used, the IP Address and Port for that connection can be entered. See [here](#)[602] for details.

CAN Hardware Type

Currently two options for the CAN hardware are available:
Peak CAN USB
    Peak CAN USB dongle with one CAN port per dongle.
Don't use CAN
    No CAN bus will be used for the simulations. Navigation, scripts, etc. can still be
        simulated.
    If you are not 100% sure that your CAN settings are correct and your PC is connected to a
        CAN bus with other devices on the bus or even in a vehicle, you should use this setting.
The CAN baudrate that the simulation will use is taken from the current project's settings.

JavaScript Debugging

It is possible to use the QtScript Debugger to debug JavaScript execution in the Windows
simulation. For a short description see [here](#)[514] .
In the dialog three basic debugger settings can be configured:
- **Off**
  Do not use the debugger.
- **On (show debugger on error or trigger)**
  Open the debugger window as soon as an exception is thrown or a *debugger;*
  statement is evaluated in a script.
- **On (show debugger on startup)**
  The debugger window will be opened as soon as the simulation starts.

Expert Settings

The Expert Simulation Settings dialog can be opened by clicking on the **Expert Settings**
button on the bottom of the Simulation Settings dialog.
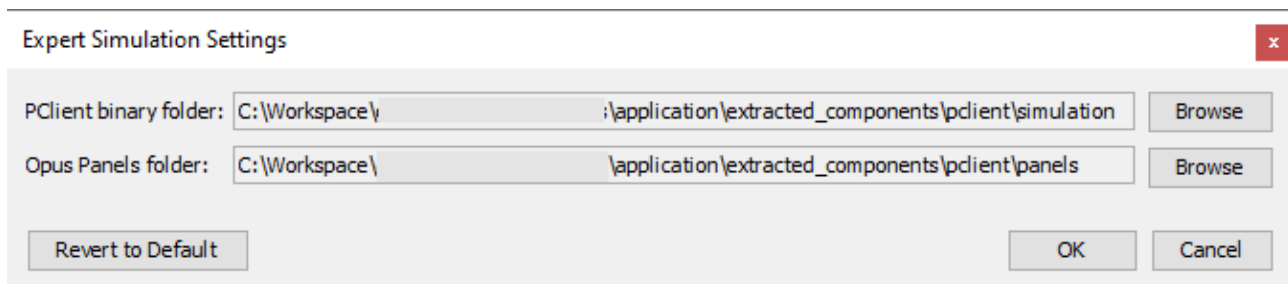
This dialog can be opened via Simulation Menu => Simulation Settings and then clicking the
Expert Settings button
The following settings can be made:

PClient binary folder

This is the path to the directory where PClient is installed.

Under normal conditions it is not necessary to change this path. It is set by the tool installer.
If another PClient version shall be run with the current tool, the path of the PClient can be
chosen via the **Browse** button.

Panel Folder

This is the path to the directory where the hardware panel templates are installed

Under normal conditions it is not neccessary to change this path. It is set by the tool installer. Via the **Browse** button another path can be chosen.
If the PClient simulation does not show the housing of the device, make sure that this path is correct.

Revert to Default

With the "Revert to Default" button at the lower left of the dialog all settings made in this dialog can be set to the default setting.

9.2   PClient Simulation

What is the PClient Simulation?

The PClient is the application that reads and runs the saved project.
The PClient Simulation was developed to make it easy to test small changes in a project without having to transfer the project to the device. It allows to run a version of the PClient directly on the Windows PC with the currently active project from the tool.
The PClient Simulation shares a lot of it's code with the version installed on the devices but since the simulation is run on a very different hardware and operating system it can not run 100% the same as on the device. Please keep this in mind and don't forget to test everything on the final hardware. Especially file operations that contain absolute paths or relative paths outside the project directory will not work on a Windows PC (e.g. changing images in JavaScript with the setProperty() functions).
Additionally it is possible to start a simulation mode on the PClient running on the device. In this mode, it is possible for applications like the Variable Simulation[601] to connect to a PClient running on the device. See Simulation Settings[596] for details about how to enable that mode.

Starting the PClient Simulation

It is possible to start a simulation of the currently active project using PClient:
- via Simulation Toolbar[598]
- via Simulation Menu[599]

Before the simulation is started, it should be checked if the simulation settings are correct. To do that open the **Simulation Menu** on **Simulation Settings**[596] :
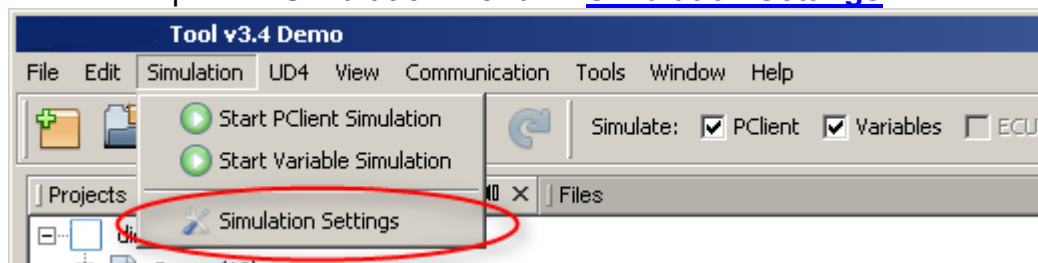
Important Notes:
- The PClient simulation will always use the last saved version of the active project. So make sure that the project you want to simulate contains no unsaved changes.
- Starting PClient Simulation on PC is not possible if the connection settings are not set to "Local PC"
- Starting PClient the first time may take several seconds. Please be patient until it shows up. All following starts will be significantly faster.

Starting from Simulation Toolbar

- Select the project you want to simulate from the project tree.
- Make sure that the **PClient** checkbox is set.
- After that just hit the **(Re)Start** button.
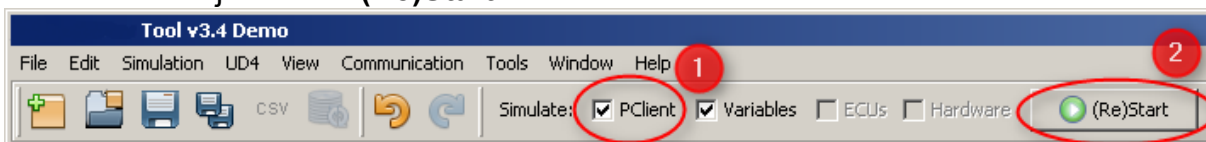
Starting from Menu

- Select the project you want to simulate from the project tree.
- Open the **Simulation menu** and click on **Start PClient Simulation**.
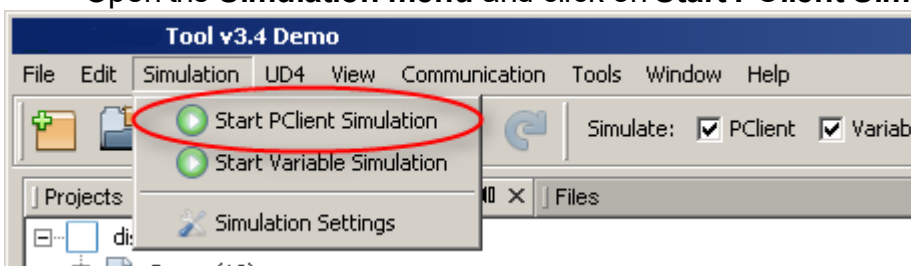
Note: If the connection mode of the PClient was configured to connect via ethernet to a running PClient on a device, the local PClient simulation cannot be started (checkmark in the simulation toolbar is disabled).

The running PClient Simulation

The running PClient should behave like the "real" PClient on the device with the exception of some hardware and operating system depending functions that can not be simulated properly (yet). This includes for example that all hardware variables will not work with the simulated PClient when running on local PC.

Navigation

The running PClient can be operated via keyboard or mouse.

Mouse / Touchscreen Navigation

All painted elements will react to mouse clicks like they would on touches on the real hardware. Input objects and Softkeys can be clicked and the assigned actions will be executed.
Additionally the keys directly on the housing will react like the real hardware keys: The softkeys on the sides, ESC, 3-Arrow and Home key on the housing can be clicked.

Above: A running PClient simulation on the device with the click areas on the housing marked with red rectangles and the associated keyboard keys.

Navigation via keys

The following keys simulate the hardware keys:

Hardware Key	Key on PC Keyboard
Softkeys	The F-Keys (F1 to FX, depending on device type, starting from Top-Left)
ESC	Esc
3 Arrow Key	Tab
Home Key	Home Key
Encoder Press	Enter
Encoder Turn	Up/Down or mouse wheel

The output of the PClient console will be printed into the Output Window of the program to make simulation debugging easier.
The output will be in a separate tab in the Output Window.

Stopping the PClient simulation

Simply close the window like any other window or hit ALT+F4.

9.3 Variable Simulation Overview

What can be achieved with the variable simulation?

The variable simulation is an application that connects to a PClient (simulated or on a device) and provides ways to view and change variable values. With the variable simulation the following can be done:

- Monitor the values of a running PClient project
- Manually change the value of each variable
- Create automatic value generators that can automatically change the value of variables in different ways
- Create lists that can set the values of different variables to defined values

How to start the variable simulation

The variable simulation can be started individually or together with a PClient simulation. To start the variable simulation set the "Variables" checkbox in the tool's simulation toolbar and then press the (Re)Start button.
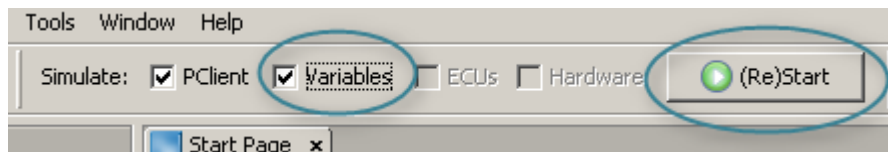
Or go to the simulation menu and then select "Start Variable Simulation".
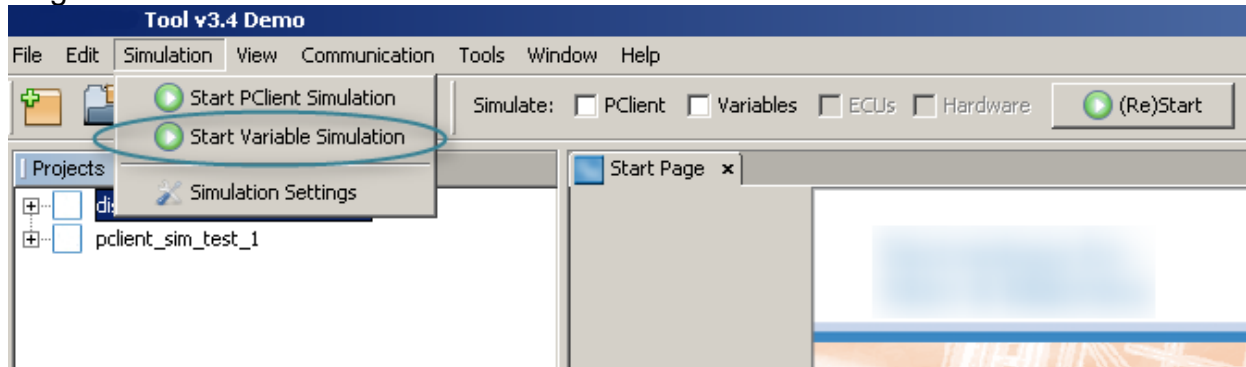
Please Note:
- A variable simulation without a running PClient simulation is useless.
- A project needs to be open and active in the tool before the simulations can be started.

Connecting to local PClient simulation

To connect to a local PClient simulation the following settings have to be made in the <u>Simulation Settings</u>◻⁵⁹⁶ dialog (these are the default settings):
- Connection Settings
  - PClient runs: On local PC
  - IP Address and Port: will be set automatically
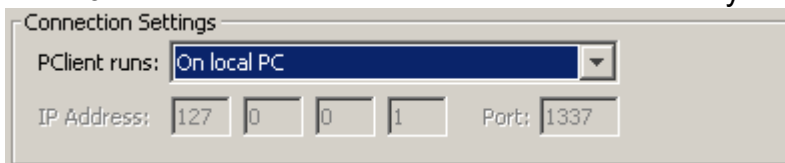
Connecting to a PClient running on a device

To connect to a PClient running on a device, the settings in the Simulation Settings dialog and some settings in PClient itself have to be adjusted.

Enable Ethernet and Simulation Mode in PClient

1. IP Address
   The IP address that PClient uses is stored in the variable **@Ethernet0_IPAddress**. This address can be changed to fit to the Network of the connecting PC. Also check the other variables in the Ethernet variable group if they need adjusting (like netmask **@Ethernet0_Netmask**). Contact the local network admin if there is a doubt about the correct IP settings.
2. Port
   The port that is used for simulation can be changed if it collides with other applications in the network. Use the variable **@PClientSimulationPort** in the Projekt variable group to change the port.
3. Enable Ethernet

To start the ethernet interface the value 1 needs to be written to the variable **@Ethernet0_applynstart**. After that the network interface should be running. This only has to be done if the value of variable **@Ethernet0_UpAtStart** is not set.

To enable ethernet automatically at each cold boot use the variable **@Ethernet0_UpAtStart**. Set value 1 to this variable if the ethernet interface shall be automatically started. If value 0 is set, the variable **@Ethernet0_applynstart** has to be set (see above).

4. Enable or disable simulation mode

To enable simulation mode write the value 1 to the variable **@PClientSimulationMode** in the Projekt variable group. After that the PClient will listen to connection attempts from variable simulation on ethernet.

Steps 1 to 3 only have to be done only once since PClient remembers those settings. A common way to enable simulation mode is:

- Set the variable values from step 1 to 3 in a script (e.g. project init)
- Create a kind of service page were the simulation mode can be enabled (e.g. with a latching button)

Troubleshooting

If the connection via ethernet fails, the following might help:

- Make sure that **@PClientSimulationMode** is set to 0 before changing any network variable or port. When setting the network variables is finished enable simulation as last step (set 1 to **@PClientSimulationMode**)
- Make sure that all network variables were correctly written to the hardware interface: Log in via serial console and type the command **ifconfig eth0**. Make sure that "*inet addr*" contains the correct IP address and "*Mask*" contains the correct netmask for your network.
- Another way to check if the connection settings are OK, the device can be "pinged". Open a DOS command prompt and type **ping <IP_ADDRESS>**. Were <IP_ADDRESS> is the address set in step 1 above (e.g. ping 192.168.135.2). If the output of that command shows timeouts, something is wrong with the settings.
- Restart the the complete device after setting the network variables. Enable network interface and simulation mode after the restart.
- Contact the local network admin to get help about the correct IP address and port that should be used in the network.

Ethernet settings for the variable simulation

To connect to a PClient running on a device with enabled simulation mode, the following settings have to be made in the <u>Simulation Settings</u>[596] dialog:

- Connection Settings
  - PClient runs: On panel
  - IP Address: The IP address used by PClient on the device (see @Ethernet0_IPAddress and step 1 above)

o Port: The simulation port used by PClient on the device (see @PClientSimulationPort and step 2 above)

```
┌─ Connection Settings ─────────────────────────────────┐
│  PClient runs:  [On      panel            ▼]          │
│                                                        │
│  IP Address:  [192] [168] [0] [2]   Port: [1337]      │
└────────────────────────────────────────────────────────┘
```
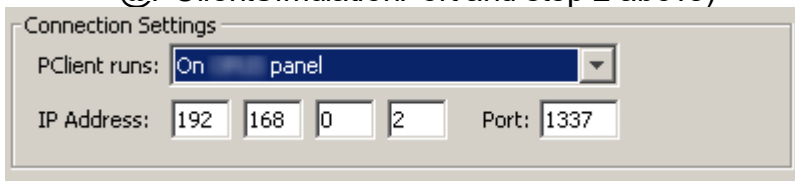
How to work with the variable simulation
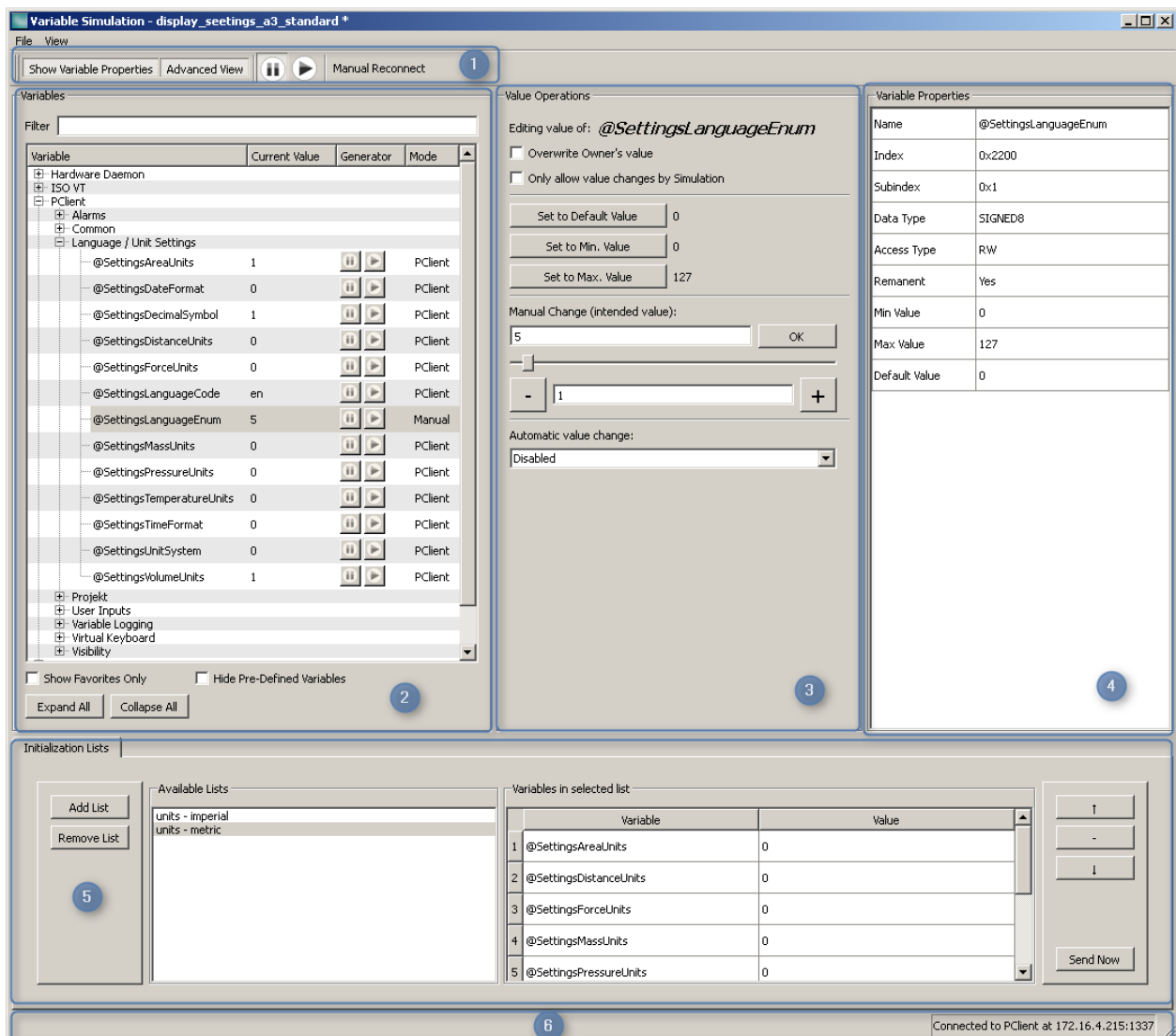
The variable simulation basically consists of the following parts (for detailed descriptions click the respective link):

1. The tool bar[606]
2. The variable view[607]
3. The value operations view[609]
4. The variable properties view[614]
5. The initialization lists[614]
6. The status bar

The left side of the status bar shows some importand status messages, e.g. when the connection to PClient is lost.
On the right side the connection status to PClient is displayed.

Saving and loading settings

Default settings

The default settings for a project are loaded automatically when starting the variable simulation and a connection to PClient was successful. The default settings will be saved individually for each project.
Each time a setting was changed and the variable simulation is closed, the user will be asked if the changed settings shall be saved. This close operation can then be cancelled, the settings discarded or saved.
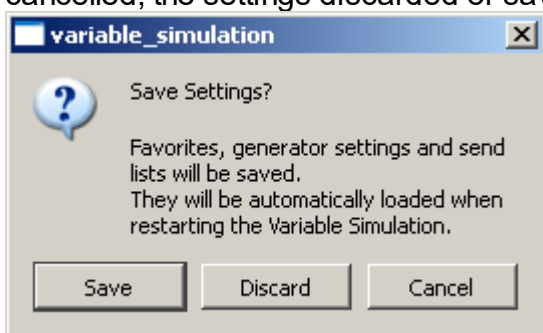
The current settings can also be stored as default settings via **File** -> **Save as Default**.
The default settings can also be loaded manually via **File** -> **Load Last Default**.
Note: When starting the variable simulation again from within the tool, it will be restarted and any changed setting will be lost!

Saving settings as a file and loading from a file

It is possible to save the current settings in a file via **File -> Save Settings As**. A file dialog will appear where the save directory can be chosen and a file name for the settings file can be specified. Name and suffix of the file are not important.
A saved settings file can be loaded by **File** -> **Load Settings**. A file dialog appears in which the settings file that shall be loaded can be selected.
If a settings file is loaded that was saved from another project, a warning will be displayed.
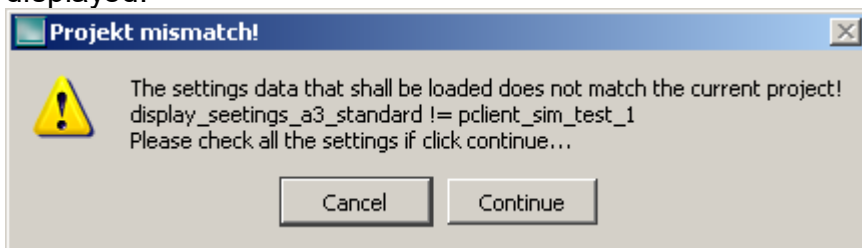
FAQ

For frequently asked questions click [here](#)[616].

9.3.1 Variable Simulation Toolbar

The tool bar of the variable simulation contains the following elements:
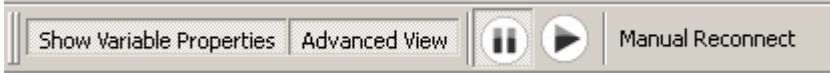
- Button "Show Variable Properties"
  With this button the <u>Variable Properties</u>[614] view can be shown or hidden.
- Button "Advanced View"
  With this button the Advanced view can be shown or hidden.
- Button
  Use this button to pause all automatic value generators
- Button
  Use this button to continue all automatic value generators
- Button "Manual Reconnect"
  This button can be used to manually reconnect to a running PClient. If any settings were changed, a popup asking to save the settings will pop up before the reconnect is done.

9.3.2   Variable Simulation Variable View
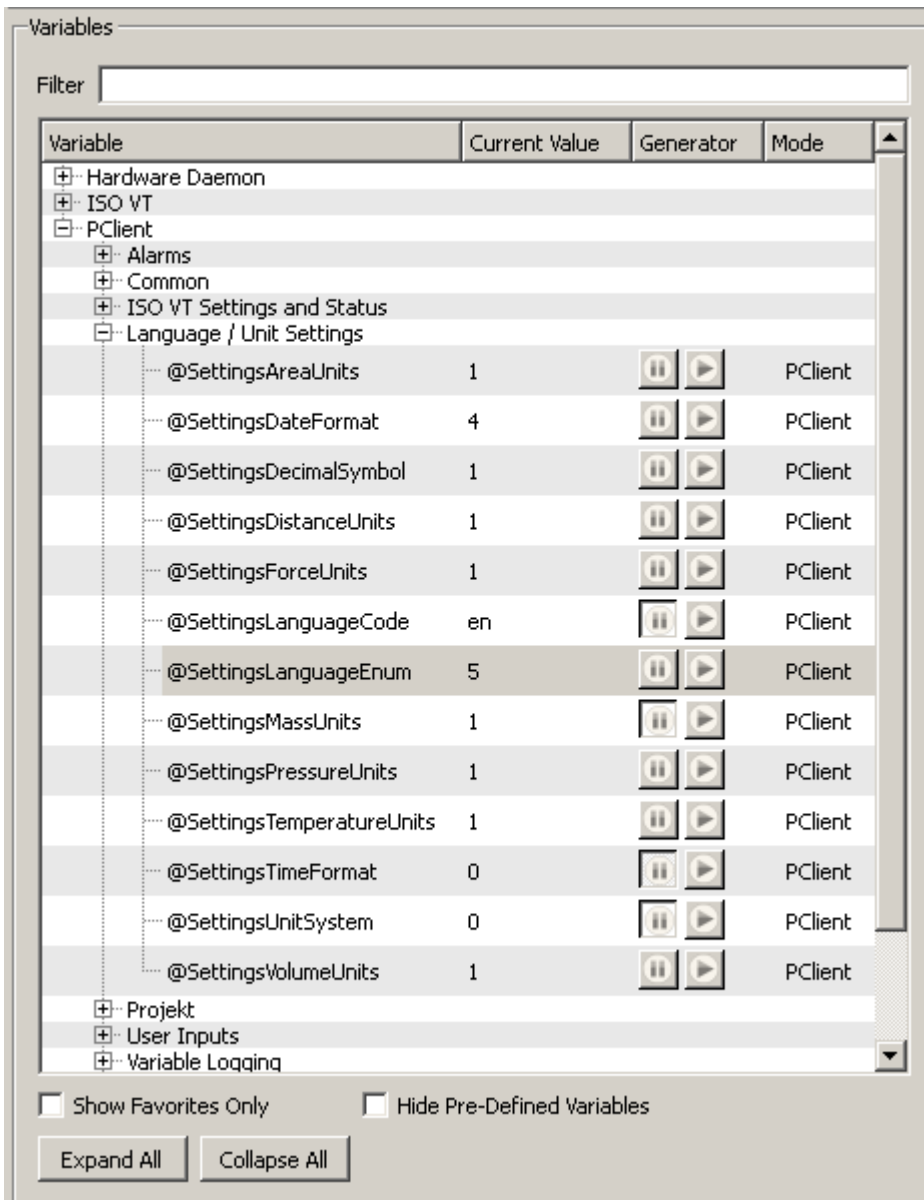
The variable view shows all variables that are configured for the currently simulated project.
The data is obtained by the running PClient simulation.

The variable tree

The variables are shown in a tree. The top level node of that tree are the variable owners.
Below the owners are the variable groups as configured in the project. The groups again
contain the variables.
The tree is sorted alphabetically.
To go to a deeper level of the tree simply click the ⊞ symbol of a node in the tree or double-
click the node itself. To collapse a node click the ⊟ symbol or double-click the node.
The whole tree can be expanded or collapsed with the "Expand All" and "Collapse All"
buttons at the bottom of the variable view.

The variable node

The variable node in the tree shows the following information:

- The variable name
- The current variable value as reported by the PClient
  - If the value of the variable is grey, the value of the variable is invalid
  - If the value of the variable is black, the value of the variable is valid
- Automatic Value Generator[609] Play and Resume buttons
  - If a generator was configured for this variable and the generator settings are in the valid range for that variable, the buttons are enabled and can be clicked
  - If no generator is configured for the variable or the generator settings contain invalid values, the buttons are disabled
- The "Mode" column show how the current value was set:
  - **PClient**: The value was retreived by PClient. The simulation did not change the value (yet).
  - **Manual**: The value was manually changed via the manual value change[609] by the simulation.
  - **Auto**: The value was changed by an automatic value generator[609] .
  - **Stopped**: An automatic value generator [609] was just stopped

By clicking on a variable node, the value operations view[609] and the variable properties view[614] will be populated with the data of the selected variable.

Favorites

Add and remove favorites

To add a variable as a favorite simply right-click on the variable. In the context menu select "**Add to Favorites**".
To remove a variable from the favorites simply click on the variable with the **right** mouse button. In the popup menu select "**Remove from Favorites**".
When a variable was added to the favorites, it's name will appear **bold** in the variable tree.
It is also possible to select multiple variables and add or remove them from the favorites list. Simply hold the CTRL key while selecting multiple variables with the left mouse button. When the desired variables are selected, do a right-click on one of the selected variables and follow the description above to add or remove all selected variables to/from the favorites.
To add or remove all variables of one group to or from the favorites, right-click on the group and follow the instructions above (Note that only the variables inside the group will be displayed bold when they are favorites, not the group itself).

Show only favorite variables

To only show favorites in the variable tree, enable the checkbox "**Show Favorites Only**" below the variable tree.
To show all variables, remove the check mark from the "Show Favorites Only" checkbox.

Hide pre-defined variables

A project contains a lot of pre-defined variables that might render the variable tree a little confusing. To hide the pre-defined variables and only show user created variables set the checkmark in the "**Hide Pre-Defined Variables**" checkbox.

Filter / Search

To filter the variable tree or to search for a certain variable, a part of the variable name can be entered in the filter field above the variable tree.

The search is case insensitive: It makes no difference if you search e.g. for CAN, Can, CaN, cAn, etc.

Wildcards (like *) are not supported in the search.

The following screenshots shows all variables that contain "can" in their name:
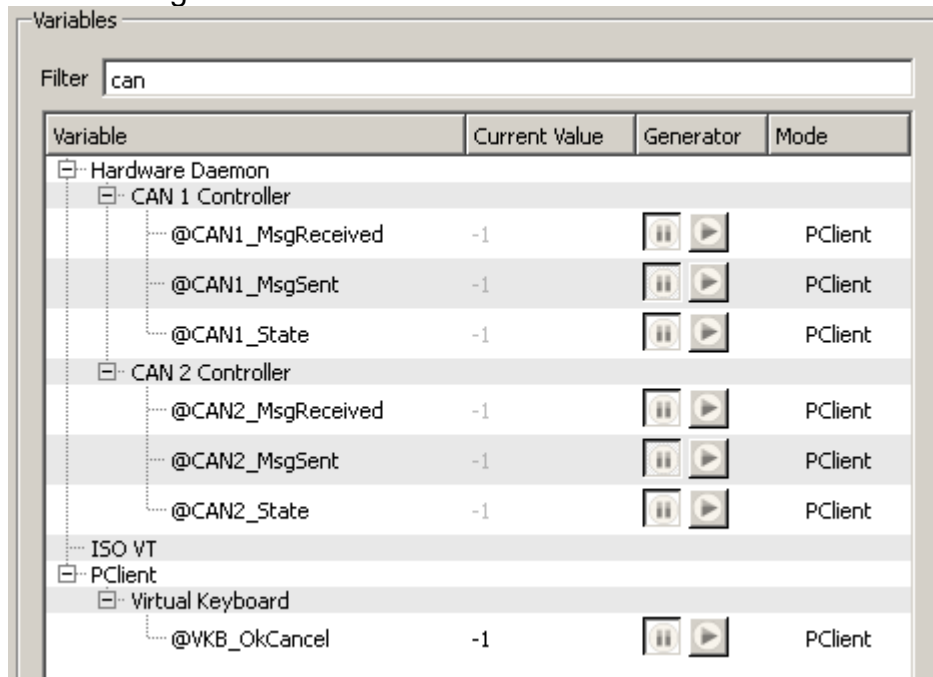
To show all variables again just remove any text from the filter field.

Note about filtering and filter checkboxes

The text filter and the two checkboxes **"Show Favorites Only"** and "**Hide Pre-Defined Variables**" work in combination. It is possible to filter for a certain text while one or both checkboxes are active, the result will then only contain favorites and/or user defined variables.

If you can't see any results, check the checkboxes and maybe remove one or both of the checkmarks.

9.3.3   Variable Simulation - Value Operations

Common

In the value operations view the value of a selected variable can be changed in various ways.
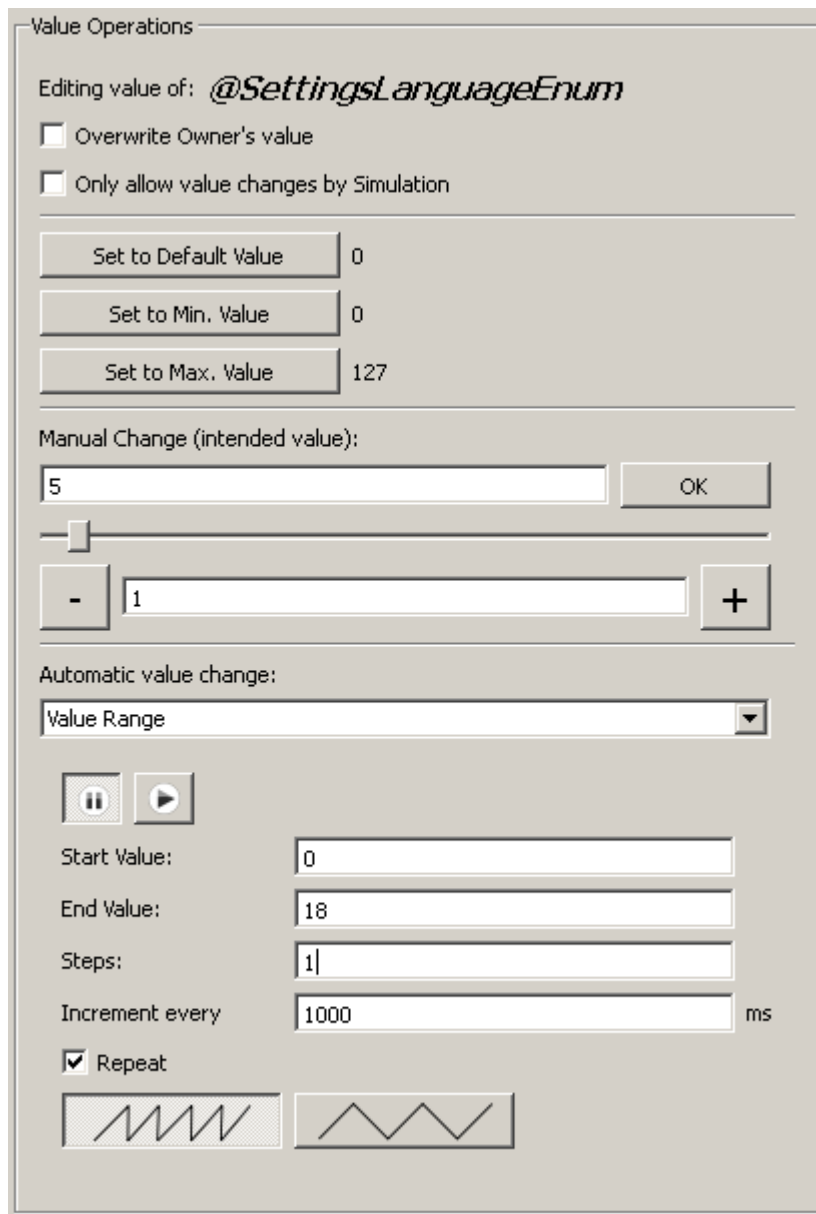
On top of the view the name of the selected variable is shown (right besides "Editing value of:").

Below that the checkbox "**Overwrite Owner's value**" is located. This is an important setting for each variable whose owner is not the PClient:

- If the owner of that variable is running and connected to the PClient
  - checkmark is not set:
    A request to update the value of the variable is sent to the owner and the owner decides if the value is valid and accepts or declines it
  - checkmark is set
    No request will be sent to the owner but internal variable value will be overwritten. E.g. the value of a variable owned by a CAN ECU will be overwritten with the configured value in the simulation regardles of what was received over CAN bus.
- If the owner of the variable is not running
  - checkmark is not set:
    Changing the value in the simulation will have no effect
  - checkmark is set:

The value set in the simulation will be accepted by PClient

Please note that all value operations (manual and automatic) use the setting of this checkbox when sending requests to the PClient.

Important note regarding the "**Overwrite Owner's value**" check box in combination with **internal variables owned by PClient**: Predefined internal PClient variables have a special action assigned (e.g. jumping to a page via @ProjectCurrentPage or managing visibility with one of the visibility variables). If the check box for **Overwrite Owners Value** for one of this variables is set, **the action for this variable will not be executed**. The value is directly written to the variable bypassing the action.

If the PClient is configured to run using a CAN interface (see), this checkbox is disabled (security reasons).

The checkbox "**Only allow value changes by Simulation**" will cause the PClient to decline any variable value change requests that were not originated by the variable simulation.

Use this checkbox for variables that are updated automatically and frequently by PClient so that the value can be simulated and no conflicts arise.

Manual Value Change

There are several ways to change the value of a variable manually:

Min., Max. and default value

To quickly set minimum, maximum or the default value of the variable, the three buttons
- **Set to Default Value**
- **Set to Min Value**
- **Set to Max Value**

can be used. The value that will be sent to the PClient is shown right beside the buttons. Please note that Min. and Max. value operations are **not available for string variables**.

Manual value changes and incrementing

Below the "Manual Change (intended value)" label there are three ways to change the value of the variable manually:
- Just enter the value that shall be set in the **manual value field** and press **enter** or the **OK** button right besides that field.
  Note: The value in this field always shows the intended value and not neccessarily the current value of the variable. Check the current value shown in the variable tree to check if setting the value was successful.
- Use the **slider** below the manual field to change the value between minimum and maximum value of the selected variable
  (not available for string variables)
- Use the **-** or **+** button to **decrement** or **increment** the variable value. The amount of decrement/increment can be set in the field between the **-/+** button.
  (not available for string variables)

Automatic Value Change

If the value of a variable shall be changed automatically, different value generators can be configured for each variable.

To chose a generator use the drop down box below "Automatic Value Change".

The default for this is "**Disabled**".

Note: The settings made for automatic variable change can be saved and loaded along with other settings.
Note: Automatic value change generators are not available for string variables.

Value Range Generator

The value range generator changes the value of a variable within a given range. The value is incremented or decremented within a configured period.
Use the following elements to configure the value range generator:

- buttons
  Pause or resume the generator (make sure that the rest of the configuration is correct)
- **Start Value** field
  The starting value of the value range. This value needs to be less than End Value
- **End Value** field
  The end value of the value range. This needs to be greater than Start Value
- **Steps** field
  The current value of the variable will be incremented or decremented by this value. Use positive values to increment and negative values to decrement.
- **Increment every** field
  This is the period after which every increment/decrement happens. The period is in milliseconds and may not be below 10ms.
- **Repeat** checkbox
  When the range was completed, repeat the sequence. See also the two buttons below that checkbox
- button
  When the end of the range is reached, the generator will start over with the start of the range
- button
  When the end of the range is reached, the generator will run backwards to the start of the range

Value Switch Generator

The value switch generator is a simple generator that allows switching between two values in a given period.
Use the following elements to configure the value switch generator:

- buttons
  Pause or resume the generator (make sure that the rest of the configuration is correct)
- **Switch between** fields
  Enter the two values that shall be switched in these two fields
- **every** field
  Enter the period with which the values shall be switched. The period is in milliseconds and may not be below 10ms.

Value List Generator
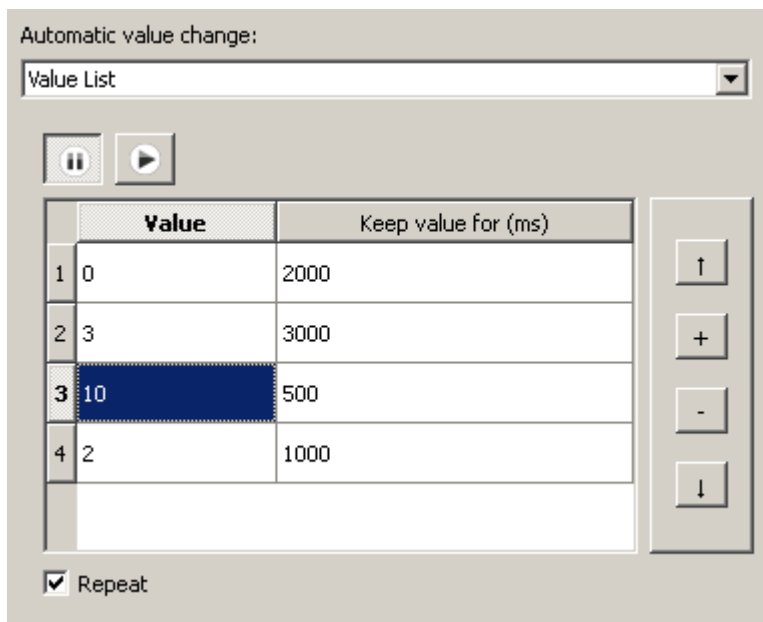
The value list generator can be used to set certain distinctive values to a variable in a defined sequence and at defined times.

Use the following elements to configure the value list generator:

- buttons

  Pause or resume the generator (make sure that the rest of the configuration is correct)

- button

  Move the selected row up in the value list

- button

  Add a new row to the value list

- button

  Remove the selected row from the value list

- button

  Move the selected row down in the value list

- The **value list** itself

  To change the value in the list just double click a cell in the left column of the list.
  To change the amount of time that a value stays, just double click a cell in the right column of the list.

- **Repeat** checkbox

  If the end of the list is reached, the generator will begin again at the top of the list

Random Generator
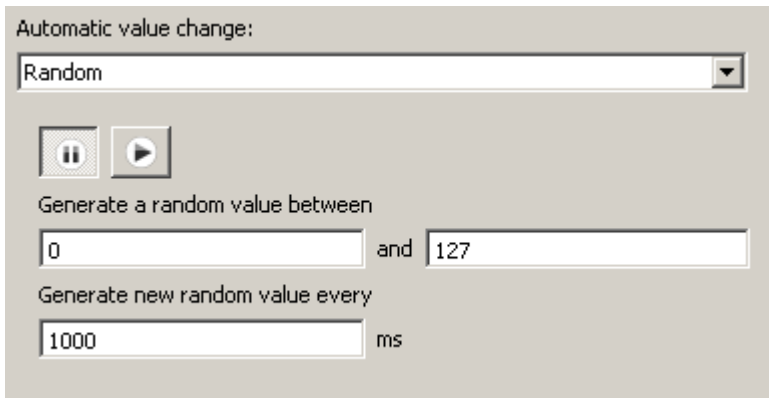
The random value generator can be used to set random values within a defined range to a variable.
Use the following elements to configure the random value generator:

- buttons
  Pause or resume the generator (make sure that the rest of the configuration is correct)
- **Generate a random value between** fields
  Use these two fields to define a range for the random values.
- **Generate new random value every** field
  Enter the period in milliseconds in which the random value shall be generated (may not be below 10ms)

9.3.4   Variable Simulation - Variable Properties

The most un-spectacular but sometimes useful view in the variable simulation is the variable properties view.
The view can be shown or hidden via the [tool bar](#)[606] .

The property table shows the variable properties as they were configured in the tool.

9.3.5   Variable Simulation - Initialization Lists

The initialization lists are part of the advanced view. The advanced view can be shown or hidden via the [tool bar](#)[606] .

Initialization lists are meant to provide an easy way to set values of several variables at once. This can be helpful for initialization purposes, end of line programming or just to test certain scenarios with different value sets.
Note: All initialization lists and their settings will be saved when the Variable Simulation settings are saved.
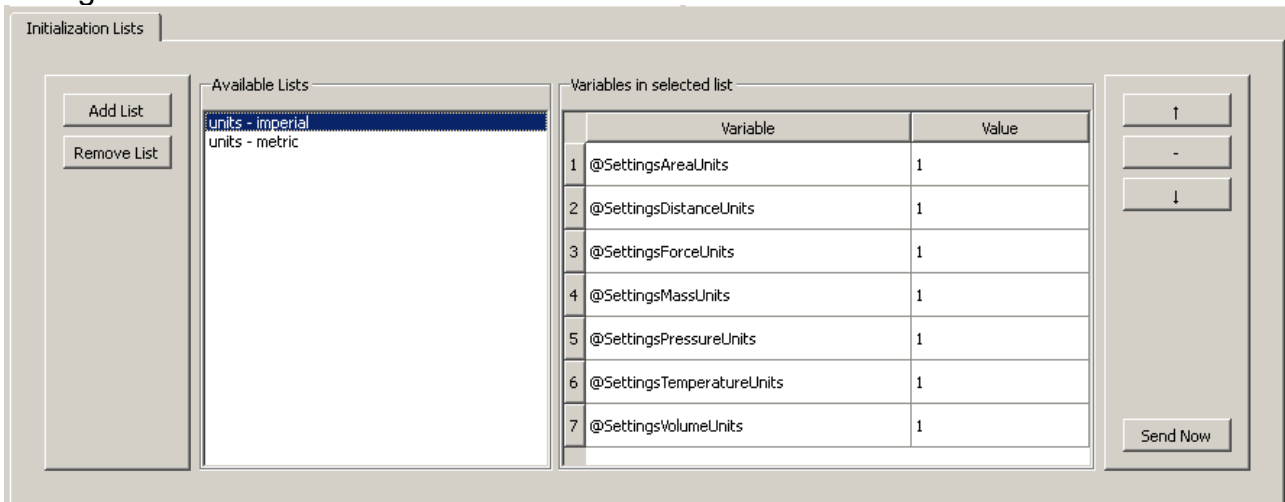
Create and remove initialization lists

To create a list click the **Add List** button on the left of the view. A new list will be generated, the name of the list can be entered directly.
To remove a initialization list select a list and click the **Remove List** button.

Working with variables in the initialization lists

Add variables to a initialization list

Make sure that a initialization list is selected. A variable can be dragged from the variable tree[607] into the initialization list table (labeled *Variables in selected list*).
After dropping the variable in the table a new row will be created with the variable's name and the current value of that variable.
This also works when multiple variables were selected and dragged & dropped to a list (select multiple variables by holding the CTRL key when clicking variables).

Sorting and removing variables

Three buttons are available with which the list can be sorted or variables can be removed:

- ↑ butoon:
  Select a variable in the initialization list table and press the button to move the selected variable one row **up**

- - button
  Select a variable in the initialization list table and press the button to **remove** the variable from the selected initialization list

- ↓ button
  Select a variable in the initialization list table and press the button to move the selected variable one row **down**

Changing the value of a variable in the initialization list

Just double-click the value in the table that shall be changed and enter the new value.

Sending an initialization list

Select an initialization list then click the **Send Now** button on the lower right of the view. The variable simulation will send requests for all variables in the list with the configured values to the PClient.

9.3.6   Variable Simulation FAQ

Why is the current value of a variable not updated after changing the value in the simulation?

The reason is a possible wrong setting of the "Overwrite Owner's Value" checkbox in the value operations view. See [here](#) [610] for a detailed description.

Why can't I see anything in the variable tree?

If the variable tree is completely empty, the variable simulation has probably lost connection to PClient (see status bar messages).
Make sure that PClient runs. The variable simulation will automatically try to reconnect to the PClient.

The variable tree shows only the owners but no variables

This can be caused by:
- A **filter text** was set but no matching variable was found
- **Show Favorites Only** was selected but no variable was marked as favorite
- **Hide Pre-Defined Variables** was selected but no user defined variables are present in the project
- A combination of the three points above

Why is the checkbox to start the PClient simulation disabled?

This checkbox is disabled if the connection settings were configured to connect to a running PClient via ethernet in the [Simulation Settings](#) [596] .

Device Software Update

10   Device Software Update

10.1   Project Update

 Tutorial video about the Updater

https://www.youtube.com/watch?v=eSW_P0Z3imU

The *Updater* is a small GUI application which allows the user to transfer a project to the device via **CAN bus** or to a **USB drive**.

The Updater has two parts.

One part is just called *Updater* or *HostUpdater*. This is the GUI application which runs on the **PC** and is described here.

The second part is running on the **device** and is called *ClientUpdater*.

How to start the Updater?

The *Updater* can only be started if a project is selected in the project tree. If the condition is achieved, the *tool bar icon* is enabled.

The *tool bar icon* can be found here:
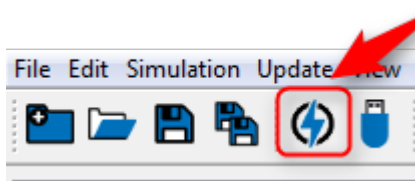
Updater Wizard

When the *Updater icon* in the toolbar is pressed, first the project will be saved, then the *Updater* will be opened, which helps you to transfer the project. It guides you step by step. First the **Start Page** (Wizard 1/3) is shown*,* where you can choose the transfer mode.

Currently you can choose between 3 update modes:

- **USB update** [619]   and
- **CAN update** [625]
- **ETH update** [618]

Start Page
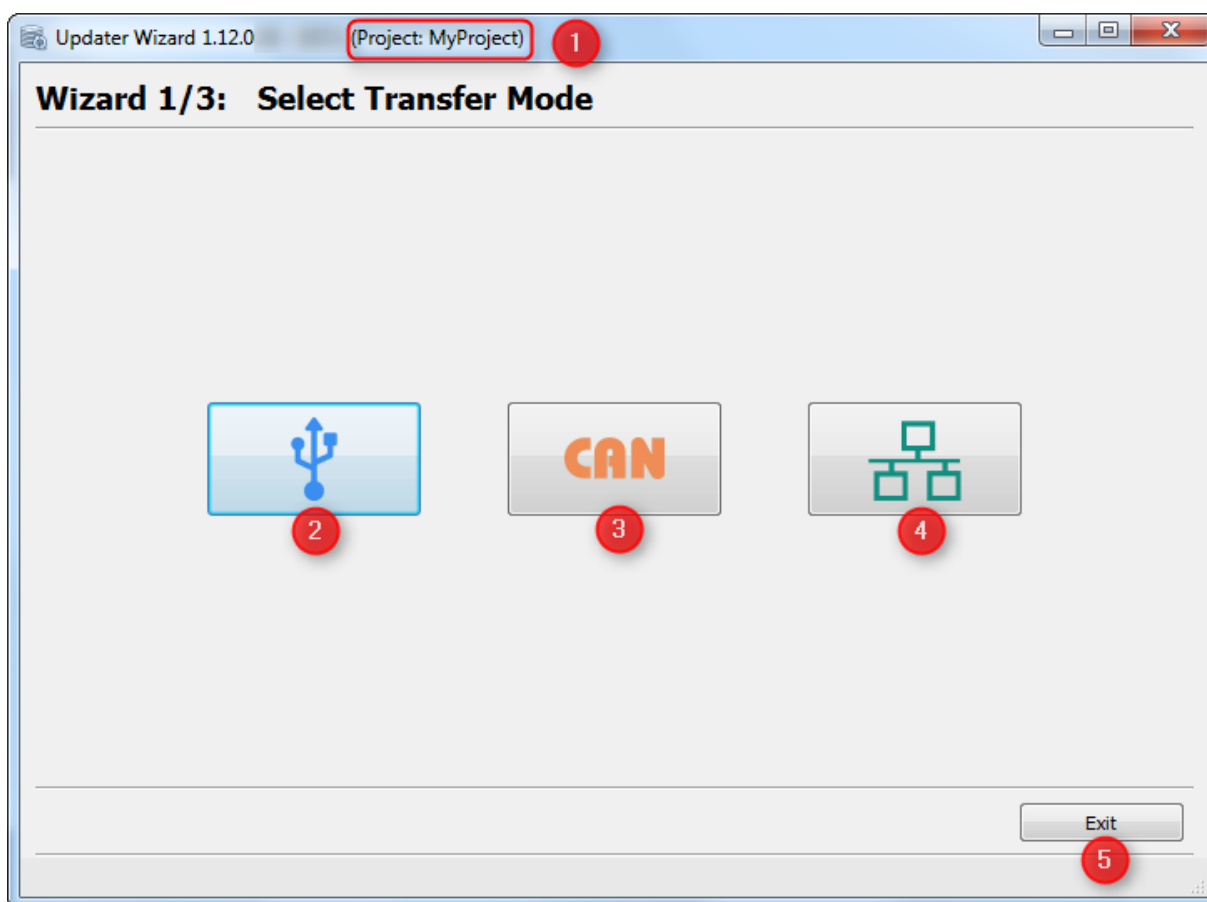
Legend:

(1) The application title shows you the name of the project which will be transferred
(Please do always check if the correct project is chosen)
(2) Start USB update
(3) Start CAN update
(4) Start Ethernet update
(5) Exit application

First we will start with the USB update mode.

⚠ **NOTE:**
The **Start Page** (Wizard 1/3) will only be shown the first time you opened the **Updater**. Once a transfer via **CAN/Ethernet** or **USB** has successfully finished, the **Start Page** will be skipped the next time you open the *Updater* and it will start with the mode you have selected the last time. You can always go back in the dialog to change to a different mode.

USB Update

If the **USB** button is clicked on the **Start Page**, the **USB Settings Page** is shown. Here you can set the USB drive to transfer the **Project Package**[644] to.

USB Settings Page
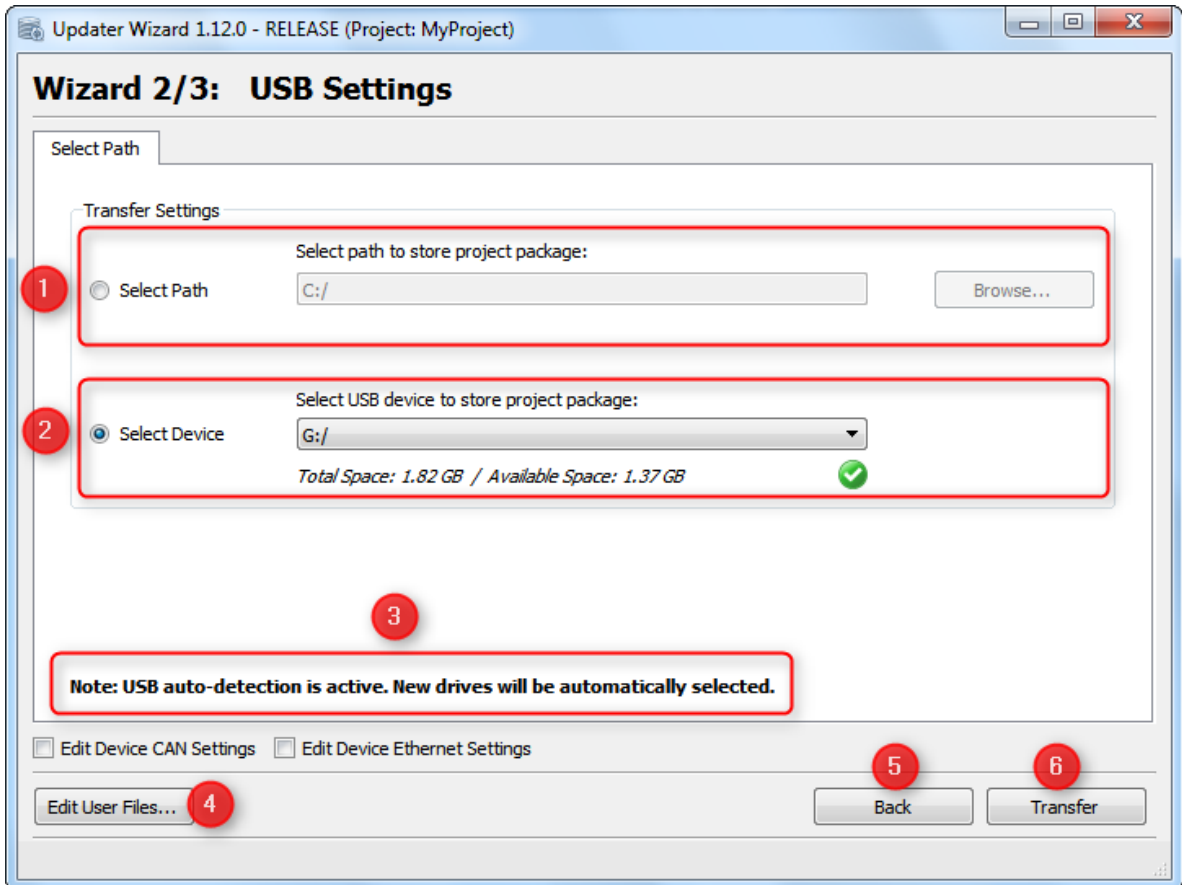
Legend:

(1) The text box shows the path or USB drive name, the **Project Package** will transferred to. If the application is started for the first time, the standard path **"C:\"** is shown. You can edit the text box or choose a path via 'Browse...' button.

⚠	**NOTE:**
	Please always select the **root directory (5, 6)** of the USB drive (e.g. **"E:\"**) otherwise the ClientUpdater application on the device **won't** be able to **install** the project package. Do **NOT** create or select any folder on the USB drive. Afterwards accept the dialog with the button "Choose Folder" **(7)**.

(2) The combo box contains all connected devices of the computer and is automatically selected. So you can easily selected the one you want transfer the project. (The selected device will be saved in the 'MainSettings.ini' file of the updater. For all other projects the device will be chosen, too)
When the device is available at program start the 'green check' is visible, otherwise a red cross is shown.
(3) The dialog has an USB auto-detection, so whenever a new USB drive is plugged-in the new device will be automatically chosen in the device combo box.
(4) Opens the '**Add User Files**' page. The page helps to add separate files which shall be transfered to the device.
(5) Go back to the **Start Page**
(6) Start **USB Transfer**.

Edit Device CAN/ETH Settings

If the checkbox is selected a new tab is shown, where the **CAN/ETH Device Settings** can be set. **(=> this is only important if you want to prepare to transfer projects via CAN or Ethernet, otherwise you can ignore it)**

These settings are important if the CAN or Ethernet update shall be enabled. Please have a look at the **CAN Checklist**[645] in this case.
Please have also a detailed look at the chapter **How to configure CAN parameters**[630]
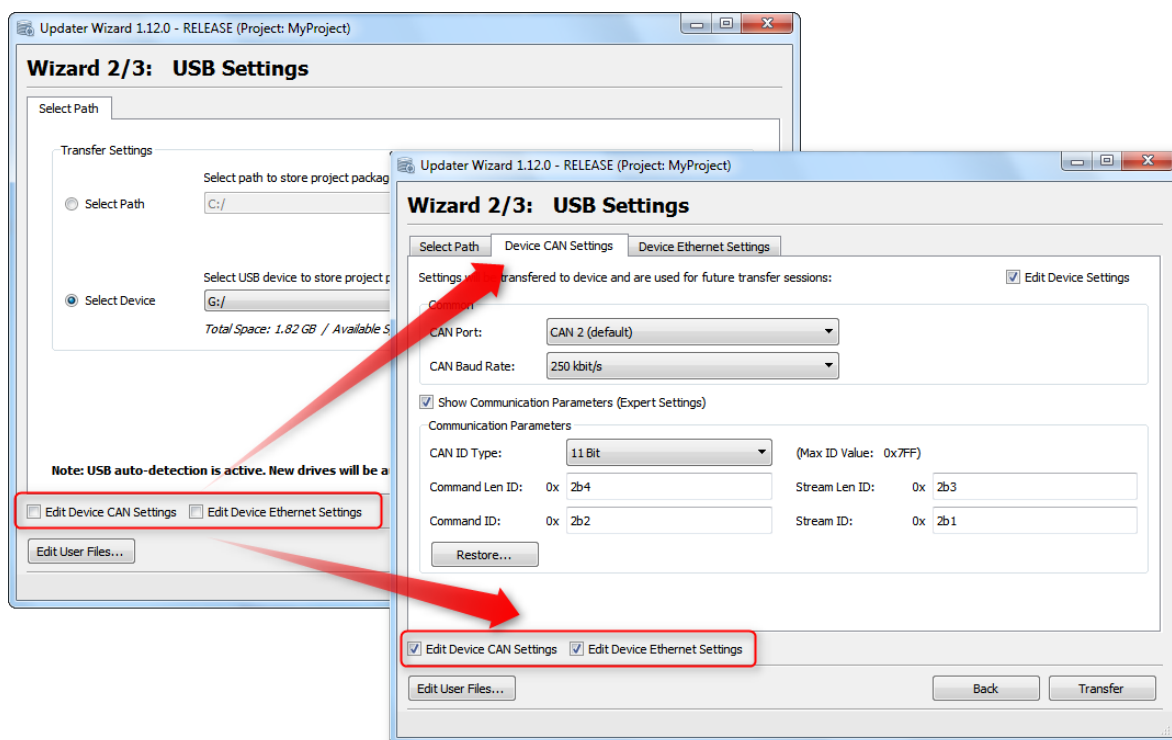
Legend:

(1) If the checkboxes are checked, the tab "Device CAN Settings" and/or "Device Ethernet Settings" are opened.
(2) In these tabs it is possible to edit the CAN/Ethernet settings for the device.

USB Transfer Page

When the **Transfer** button is pressed, this page appears and shows the current USB transfer progress and its result.
The USB Transfer is divided in 3 steps.
1. Creating Project Package,
2. Connecting to the USB drive and
3. Transferring the project.
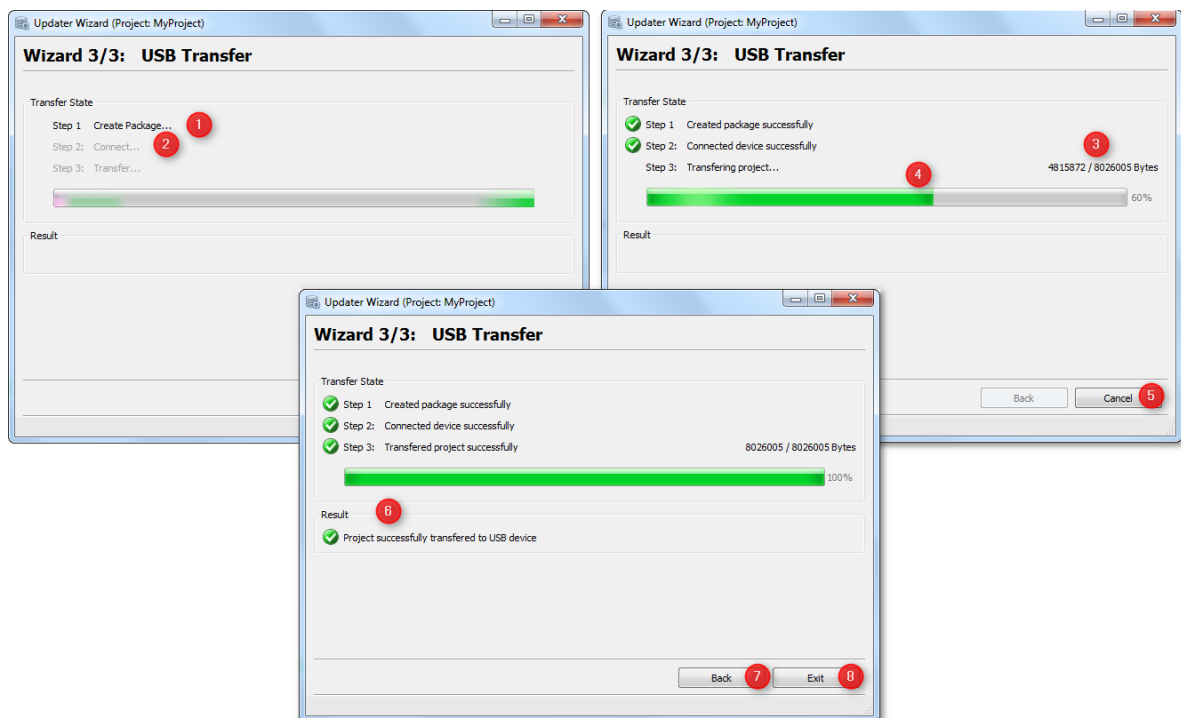
Legend:

(1) Shows status of create package.
(2) Shows if USB drive is successfully connected.
(3) (4) Shows current transfer progress in percentage and bytes.
(5) When Transfer is started, it can be aborted with the **"Cancel"** button.
(6) Shows if the project package has been successfully created on the USB drive.
(7) Button to go back to the USB Settings page. (During transfer, **"Back"** button is **disabeld**)
(8) Exit application. (During transfer **"Exit"** button is renamed to **"Cancel"**)
    If the application is closed via **"X"** button during the active transfer, a dialog is shown with a warning that current transfer will be aborted.
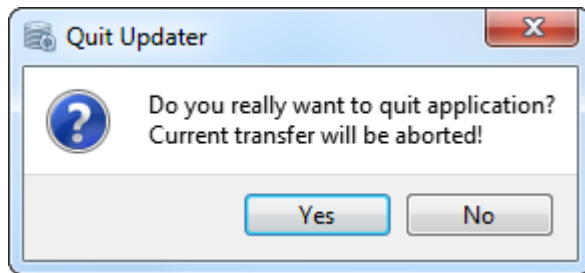
Step 1: Creating Project Package
This step shows if the Project Package was created successfully.

Step 2: Connecting device
This step shows if the USB device is connected successfully.

Step3: Transferring project
This step shows the progress of the project transfer to the USB drive.
If the transfer fails an error code and error text is shown under the progress bar.
Otherwise a message **(6)** is shown that the project was succsessfully transferred to the USB drive.

⚠ **NOTE 1:**
If the project package was successfully transferred to the **USB drive**, a new directory **"updater"** has been created. It contains **2 files**, a file called **"mmscript.xml"** and the project package file, here called **"MyProject.ud4"**. Once the **Project Package** was transferred to the **USB drive** successfuly, the USB drive has to be plugged into the USB port of the **device** to update the project on the device. The project will only be installed on the device if the variable **@enableUpdater**[644] is set to **0x1** or **0x3**. **The default value is 0x1.**

⚠ **NOTE 2:**
The next time you open the *Updater* for the same project, the last used path or USB drive will be set again. The path or USB drive is saved in **HostSettings.ini**[643] file. The USB drive is **not unmounted** when the project is transferred to the USB drive.

Error Codes during Transfer

If the **USB Transfer** has not finished properly, an *Error Code* with *Error Message* is shown on the **USB Transfer Page**. This shall help to identify the issue quickly.

In these examples the **Step 2: Transferring project...** shows an error:

1. Unable to create folder on USB drive
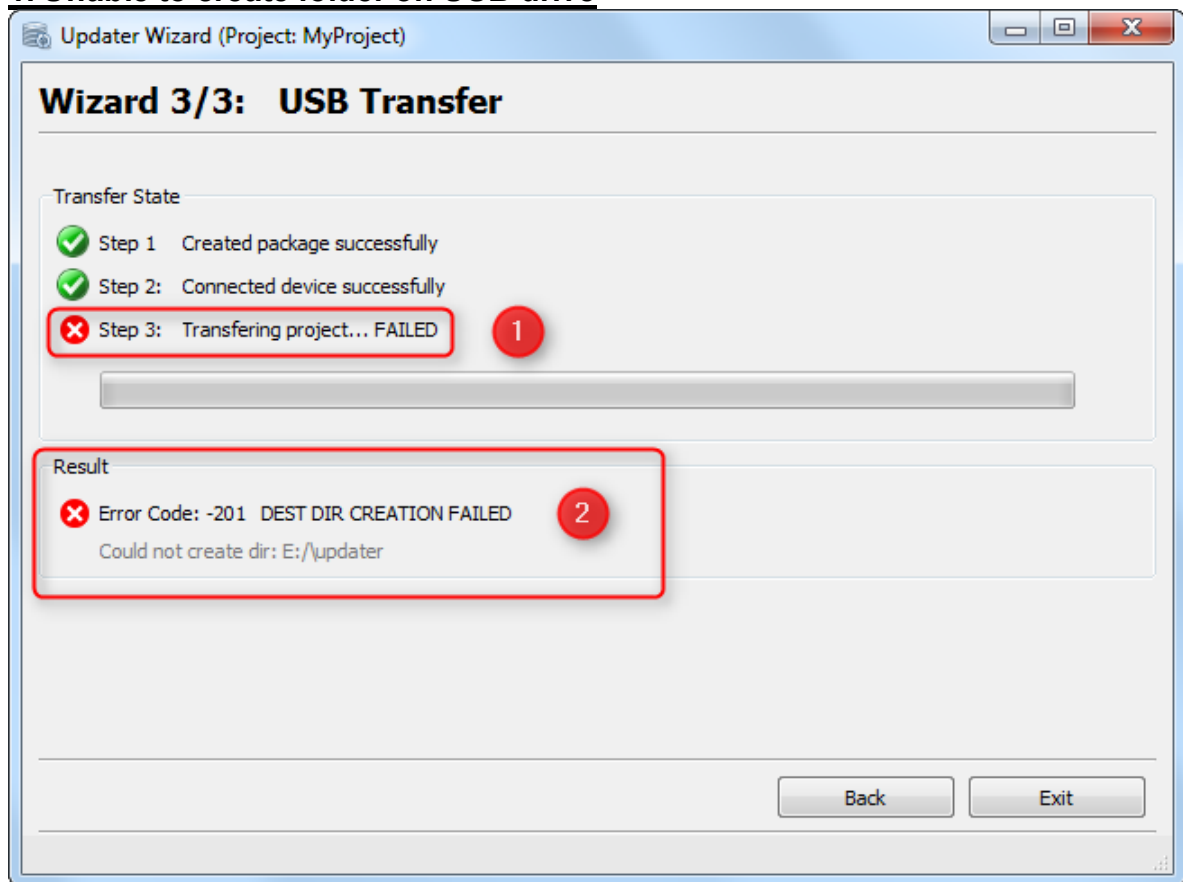
Legend:

(1) Step 2: The Transfer to USB drive failed.

(2) The *Error Code* (here: *-201*) is shown under the progress bar. After the code the appropriate *Error Message* is shown (here: *DEST DIR CREATION FAILED*).

In this case the *Updater* could not write the **Project Package** to **USB drive**. Possible reason for this error is:

- USB drive was disconnected during transfer.

2. Unable to write file on USB drive

Legend:
(1) Step 2: The Transfer to USB drive failed.
(2) The **Error Code** (here: **-207**) is shown under the progress bar. After the code the appropriate **Error Message** is shown (here:
ERR_MMFT_DEST_FILE_CLOSE_FAILED).
    Additionally an detailed message is shown what exaclty went wrong: **Could not write file: E:\updater\MyProject.ud4**
In this case the **Updater** could not write the **Project Package** to **USB drive**. Possible reason for this error is:
- USB drive has not enough disk space left.

CAN Update

If the **CAN** [CAN] button is clicked on the **Start Page**, the **CAN Settings Page** is shown. Here you can modify the CAN settings for the project transfer. Afterwards you can start the transfer.

Disclaimer about using CAN

The **CAN Update mode** is a little more complex than the **USB Update mode**. Many things have to be considered. Otherwise the transfer won't work or the **CAN network** can be **damaged**.

1. The *ClientUpdater* uses as default **CAN Port 2** on the device. If you want to use instead **CAN Port 1** you can change the port **here**□618 .
If the *ClientUpdater* uses the same CAN Port as a *CAN Protocol* in the project, please always check that the **Baud rate** for the *ClientUpdater* is the same as the one in the used **CAN Protocol** in the project.
2. To transfer the project between **HostUpdater** and **ClientUpdater**, **4 CAN IDs** will be used. These **4 CAN IDs** are neccesary and will all be used.
The IDs can be changed **here**□618. Please check, in case another protocol is running on the same CAN Port, that the CAN IDs won't be used by any other entity connected to the port.
3. The *ClientUpdater* only enables the **CAN Interface** if the value of the pre-defined variable *@EnableUpdater*□644 has been set to **0x2** or **0x3**. So check if the variable is set properly in the project (e.g. in a script for the event OnProjektInit).
4. The first project update has to be done via the **USB drive**□619. With this update the new CAN Settings for the device will be set. Only then the CAN interface can be enabled and will not damage your CAN network.

Please **follow** these steps **carefully**.

CAN Disclaimer Page

The **CAN Disclaimer Page** is only shown once in a Tool session when the **CAN** button on **Start Page** is clicked.
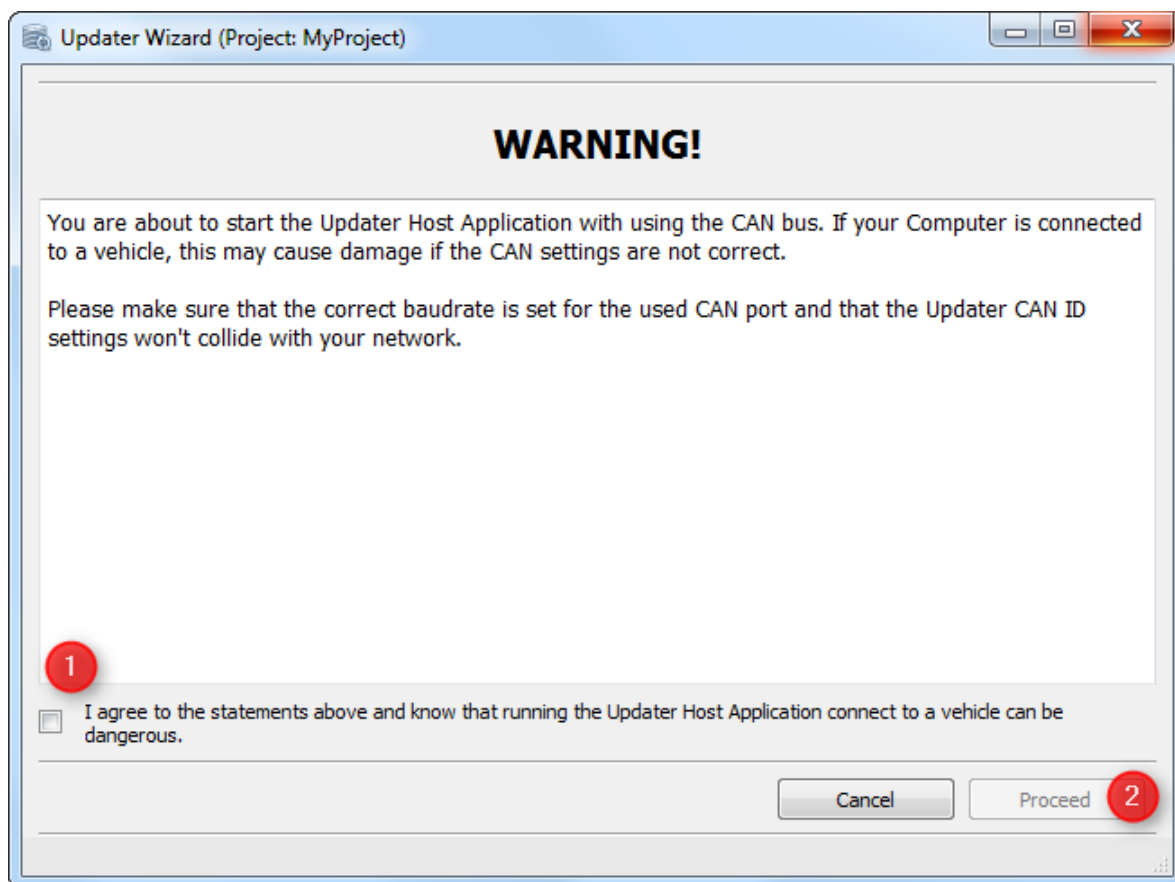
Legend:

(1) When the **agree checkbox** is checked, the **Proceed** button will be enabled.
(2) If **Proceed** button is pressed you will see the **CAN Settings Page**.
(3) With **Cancel** you get back to **Start Page**.

⚠ **NOTE:**
When the Tool is started and you want to transfer a project via CAN bus you have to accept the **CAN Disclaimer**☐ 618 once. The next time in this Tool session the **CAN Disclaimer** will not be visible again.

PC CAN Settings Page

These settings will be used for the **current project transfer**. The settings shown here have to be same as they were configured for the device, otherwise the transfer will not work.
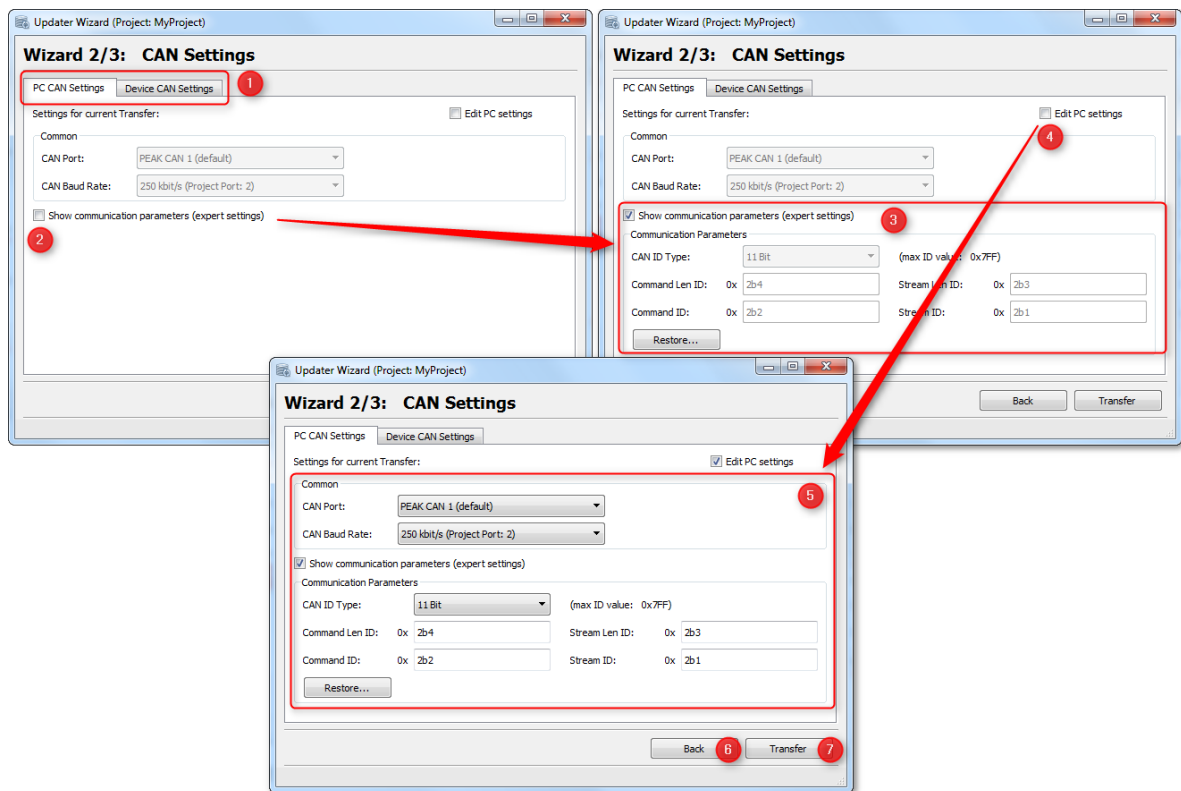
Legend:

(1) With the **Tabs** you can choose between **PC** and **Device CAN Settings**.
(2) Show communication parameters (expert settings) shows the communication parameters **(3)**.
(4) Edit PC settings enables the settings mask. If it is checked, all parameters can now be edited **(5)**.
(6) Back button to go back to **Start Page**.
(7) Starts **CAN Transfer** with current settings.

PC CAN Parameter
1. **Common Settings**
   - ☐ *CAN Port*
     - ▪ Port Number of the CAN dongle. Currently **only PEAK CAN dongle**s are supported on **Port 1/2**.
   - ☐ *CAN Baud Rate*
     - ▪ **Baud Rate** in **[kbit/s]**. The Project Package will be transmitted with this Baud Rate
     - ▪ Allowed Baud Rates are (20, 50 , 83.333, 100 , 111.11, 125, 250 ,500 ,1000)
1. **Expert Settings**
   - ☐ *CAN ID Type*
     - ▪ Setting **CAN ID Type** for **ID**s. Either **11 Bit** or **29 Bit**.
   - ☐ *Command Len ID*

☐ *Command ID*
☐ *Stream Len ID*
☐ *Stream ID*

- These for IDs will be used for CAN communication between **HostUpdater** on PC and **ClientUpdater** on device.
- It is important that these IDs are set properly, otherwise the connection or transfer can not be established.
- Each ID has to be in the valid range and must be unique.

⚠ <u>**NOTE (DANGER):**</u>
Please make sure that the **CAN ID**s which will be used for the **CAN Transfer** are **not used** in the **project** and **are not used** from **any other entity** on the CAN bus.

Device CAN Settings Page

These settings will be **transmitted to the device** in **this transfer session**. The transmitted settings are **available** after the **update has successfully finished**.
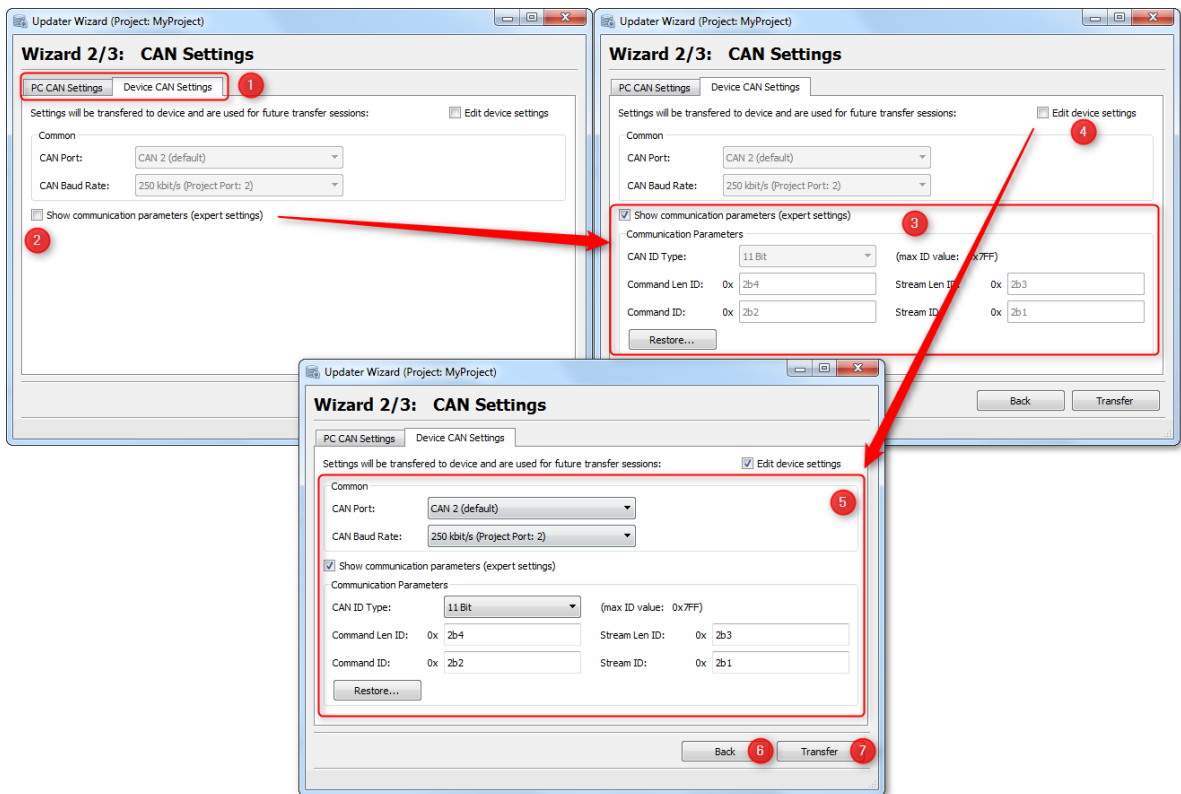
<u>**Legend:**</u>

(1) With the **Tabs** you can choose between **PC** and **Device CAN Settings**.
(2) Show communication parameters (expert settings) shows the communication parameters **(3)**.

(4) Edit device settings enables settings mask. If it is checked all parameters can now be edited **(5)**.
(6) Back button to go back to **Start Page**.
(7) Starts **CAN Transfer** with current settings.

Device CAN Parameter
  1. **Common Settings**
     ☐ *CAN Port*
       ▪ Choose the CAN Port for using project updates. Either **CAN 1** or **CAN 2** can be used.
     ☐ *CAN Baud Rate*
       ▪ **Baud Rate** in **[kbit/s]**. HostUpdater will use this Baud Rate for project updates.
       ▪ Allowed Baud Rates are (20, 50 ,83.333, 100 111.11,,125, 250 ,500 ,1000)
  1. **Expert Settings** (same as **PC Expert Settings**[627])

⚠	**NOTE:**
	The **CAN Settings (PC and device)** are saved when the project transfer has successfully finished. Next time you open the *Updater* for the same project, the last used **CAN Settings** will set again. The CAN Settings are saved in **HostSettings.ini**[643] file.

How to configure CAN Parameters

When the Updater is started for a project the first time the **default values** will be set for the **CAN Parameters**:

Default CAN Parameters:

	PC	Device
CAN Port	PEAK CAN 1	CAN 2
CAN Baud Rate	250 kbit/s	[250 kbit/s] Project dependent
CAN ID Type	11 Bit	11 Bit
Command Len ID	0x2b4	0x2b4
Command ID	0x2b2	0x2b2
Stream Len ID	0x2b3	0x2b3
Stream ID	0x2b1	0x2b1

CAN Port settings
  1. **PC**
     Port of **PEAK CAN dongle (Port 1/2)**.
  2. **Device**
     As default the **CAN Port 2** is used for CAN updates on the device. But it is also possible to use **CAN Port 1**. Simply set the **CAN Port** combo box in **Device CAN Settings** appropriate.

CAN Baud Rate setting

During the start, the *Updater* reads the project file and checks if **CAN Protocols** are used and on which **CAN Ports** the protocols are running.

If protocols are used, the appropriate baud rate is read. If a protocol uses the same port as the *Updater,* the appropriate baud rate is set in the **Device CAN Settings** (also for the first start of the *Updater*).

The baud rate combo box shows whether a baud rate is configured for a CAN Port which uses a CAN Protocol. The appropriate baud rate then gets an extension like this "*250 kbit/s (Project Port: 2)*"

The *Updater* usually should always set the baud rate for **PC** and **device properly** and it is not necessary to set them manually. So please be careful if you change the baud rate.

If the *Updater* uses a **CAN Port** which is not used by other protocols, the baud rate is set to **250 kbit/s as default,** but in this case you can set a preferred Baud rate by yourself.

CAN ID Type setting

Here you can define whether **11 Bit** or **29 Bit CAN IDs** shall be used for the CAN communication.

For **11 Bit CAN IDs** all values between **0x0 - 0x7FF** are allowed.

For **29 Bit CAN IDs** all values between **0x0 - 0x1FFF FFFF** are allowed.

Please check the **CAN IDs** after updating the **CAN ID Type**.

CAN ID settings (Command Length ID, Command ID, Stream Length ID and Stream ID)

These **ID**s will be used for the **CAN communication** between the **HostUpdater** and **ClientUpdater**.

Each **ID** has to be **valid** (in the range of the **CAN ID Type**) and **unique**.

If the IDs are **not valid,** the **Transfer** button is **disabled**.

It is very important that the configured IDs won't be used by other entities which also use the same CAN bus, otherwise the project update process won't work properly and/or other entities won't work properly.

Restore CAN Settings

With the button **"Restore..."** you can restore the CAN parameters for PC or Device
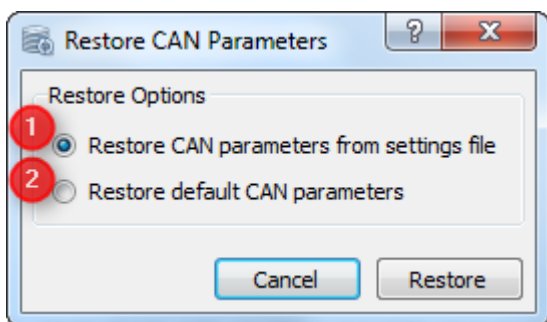
You can choose between **2** options:
(1) Restore CAN parameters from settings file. This option can be used to restore the last working CAN settings. The settings will be restored from the HostSettings.ini file.
(2) Restore default CAN parameters. This option can be used to restore the factory default CAN settings. Could be reasonable if the device was updated with the latest OS.

CAN Transfer Page

The CAN Transfer Page shows the current transfer progress and if the current transfer finished successfully or with an error.
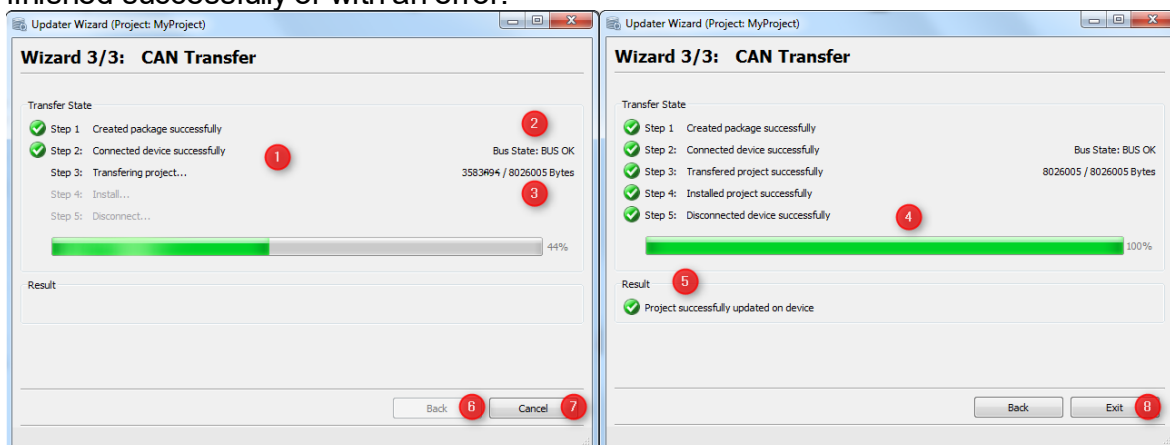
Legend:

(1) On the left side you can see the current step which is permformed. Last Step with black font color => current performed step (here Transferring Project), grey font color steps which have not be performed yet.
(2) Shows the current **CAN BUS State** (States are: *BUS_OK*, *BUS_LIGHT*, *BUS_HEAVY* and *BUS_OFF*)
(3) Transferred Bytes and Project Package size. **(4)** Progress bar with transferred bytes in percentage.
(5) Shows the result of the Transfer. If the transfer fails, an error code and error message are shown.
(6) Back button which is disable during active transfer. When transfer finishes, the button is enabled.
(7) Exit/Cancel button. During transfer button is renamed to **"Cancel"**. When the transfer finishes, the button is renamed to **"Exit"**.

If the application is closed via the **"X"** button during active transfer, a dialog is shown with a warning that the current transfer will be aborted.
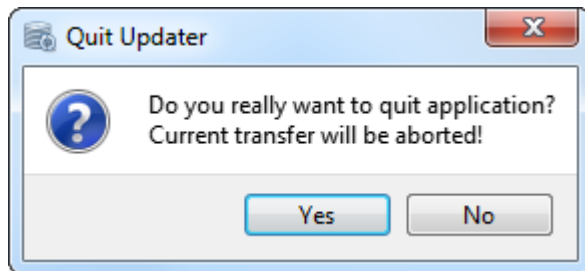
The Transfer is devided in **5 Steps**:

1. **Creating Project Package => Creates the Project Package which is transmitted.**
2. **Connecting   => Tries to connect the device via CAN bus.**
3. **Transferring   => Transfers the Project Package**
4. **Installing   => Installs the Project Package on the device**
5. **Disconnecting   => Disconnects device.**

If a step has finished successfully, a **green check mark** is shown and the next step is starting.

If as step fails, a **red x** is shown and the the update progress stops. In this case an **error code with error text** is shown.

The current step is always shown in bold font (highlighted).

Step 1: Creating Project Package:

It can take **several seconds** until the **Project Package**⌐644 is created.

If this step fails, an error code with error text is shown in the GUI.

Step 2: Connecting device

In this step the **HostUpdater** tries to establish a connection to the **ClientUpdater**.

The **HostUpdater** awaits a reply within **10 seconds**.

This step can be aborted by pressing the **Cancel** button.

Step 3: Transferring project

In this step the **HostUpdater** sends the **Project Package** to the **ClientUpdater**. If the connection is interrupted for more than **10 seconds**, the transfer is aborted.

This step can be aborted by pressing the **Cancel** button.

If the **CAN transfer** does not work properly, please have a look at the chapter **FAQ: Why do the CAN Transfers not work?**

Step 4: Installing project

The transferred **Project Package** will be installed in this step. This step will automatically start on the device when **Step 3** has successfully finished.

Step 4 continues on the device, even if the CAN connection is interrupted after **Step 3** or during **Step 4**.

This step **CAN NOT** be aborted by pressing the **Cancel** button. Once the install step has started on the device, it will keep running until it is finished.

Don't disconnect the device from power supply during Step 4, otherwise the project is not installed properly and will probably not start.

If the CAN Transfer does not work properly please have a look at the chapter **FAQ: Why does the CAN Transfer not work?**⌐652

<u>**Step 5: Disconnect device**</u>

This step disconnects the *HostUpdater* from *ClientUpdater*.

If the CAN connection is interrupted, the ClientUpdater will automatically disconnect itself after **3 seconds**.

Afterwards the project is starting on the device.

Error Codes during Transfer

If the CAN Transfer does not work properly, an *Error Code* with *Error Message* is shown on the **CAN Transfer Page**. This shall help to identify the issue quickly.

In this example, the **Step 2: Connecting** device runs into a **time out**:
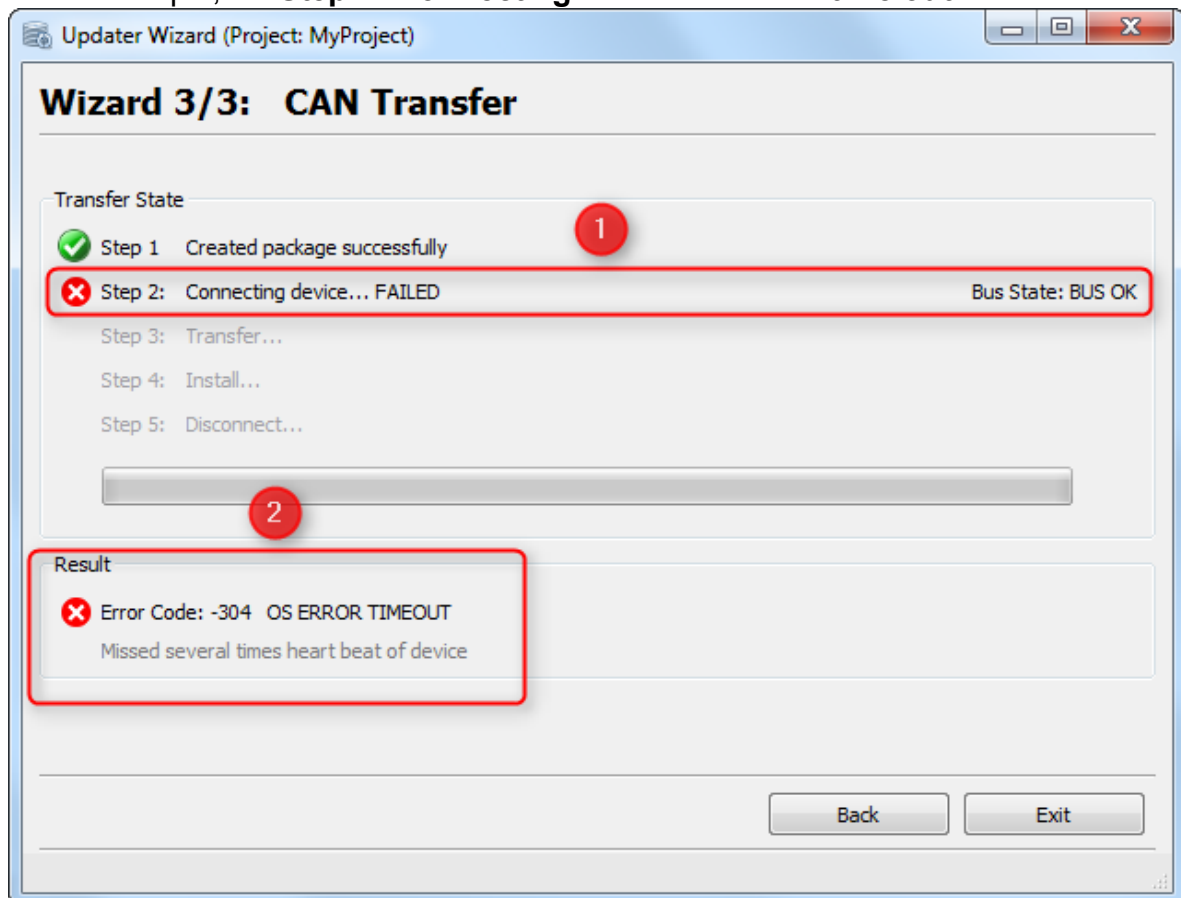
<u>**Legend:**</u>

(1) Step 1: The connection process failed and a *red x* icon is shown.

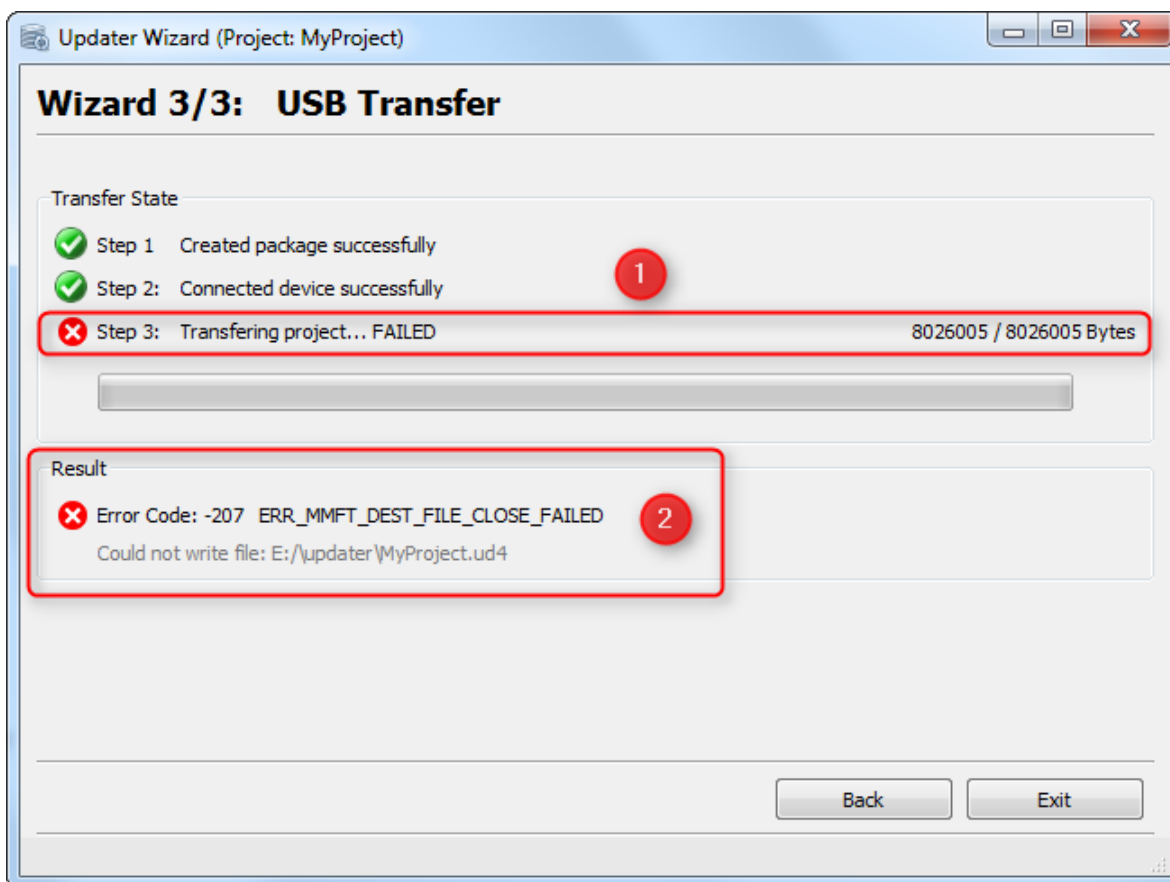**(2)** The *Error Code* (here: *-304*) is shown under the progress bar. After the code, the appropriate *Error Message* is shown (here: *OS ERROR TIMEOUT*).

In this case the *HostUpdater* gets no response from the *ClientUpdater*. Possible reasons for this error are:

- The **ClientUpdater** is not running on device
- Pre-defined variable **@EnableUpdater** is not set properly
- **CAN IDs** are not set properly.
- **NOTE:** Bus State shows *BUS OK* which indicates that the baud rate is set properly. If Bus State shows *BUS HEAVY, BUS_LIGHT* or *BUS OFF* it indicates that the baudrate is not set properly.

Other popular Error Codes are:

- **(- 1) UD_FAILURE**: It seems an unexpected error occurs. Just try again, and/or restart the HostUpdater and ClientUpdater (device reboot).
- **(-22) DEVICE MODEL MISSMATCH**: It seems the device model name of the project does not match with the device model name of the device connect on the CAN bus.
- **(-27) ACTIVE CAN INTERFACE:** ClientUpdater has been already connected via CAN. If the connection is broken during transfer, please wait a few seconds before you try again to transfer the project.
- **(-42) OPEN PORT FAILED:** The HostUpdater can not open the CAN Interface. (E.g. PEAK CAN Dongle is not plugged-in)
- **(-408) BUS ERROR:** It seems the CAN network is used with diffrent baud rates.

Ethernet Update

If the **Ethernet** button is clicked on the **Start Page**, the **Ethernet Settings Page** is shown. Here you can modify the Ethernet settings for the project transfer.
Afterwards you can start the transfer.

Disclaimer about using Ethernet

The **Ethernet Update mode** is a little more complex than the **USB Update mode**. Many things have to be considered. Otherwise the transfer won't work.

1. To transfer the project between **HostUpdater** and **ClientUpdater**, **2 Ports** will be used and must be configured properly.
2. The *ClientUpdater* only enables the **Ethernet Interface** if the value of the pre-defined variable **@EnableUpdater**[644] has been set to a value between **0x4 and 0x7**. So check if the variable is set properly in the project (e.g. in a script for the event OnProjektInit).
3. The first project update has to be done via the **USB drive**[619]. With this update the new Ethernet Settings for the device will be set.

Please **follow** these steps **carefully**.

PC Ethernet Settings Page

These settings will be used for the **current project transfer**. The settings shown here have to be same as for the connected device, otherwise the transfer will not work.

Scan Network (Broadcast):
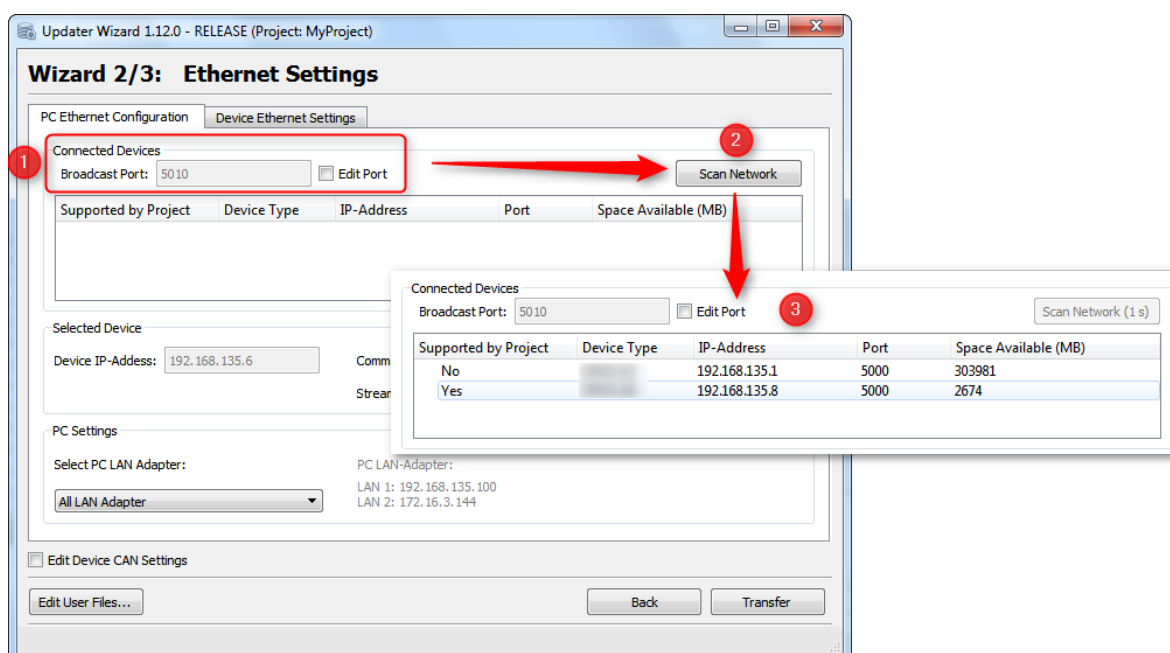
Legend:

(1) When the ethernet interface of an device is activated, it is possible to scan the network for that device. For the communication between Host and Client the **'Broadcast Port'** is used. When Host and Client are using diffrent ports the communication won't work.
(2) When the correct **'Broatcast Port'** is configured, it is possible to search after devices with the button **'Scan Network'**
(3) The list box shows all available devices in the network with some details.

PC Ethernet Parameter
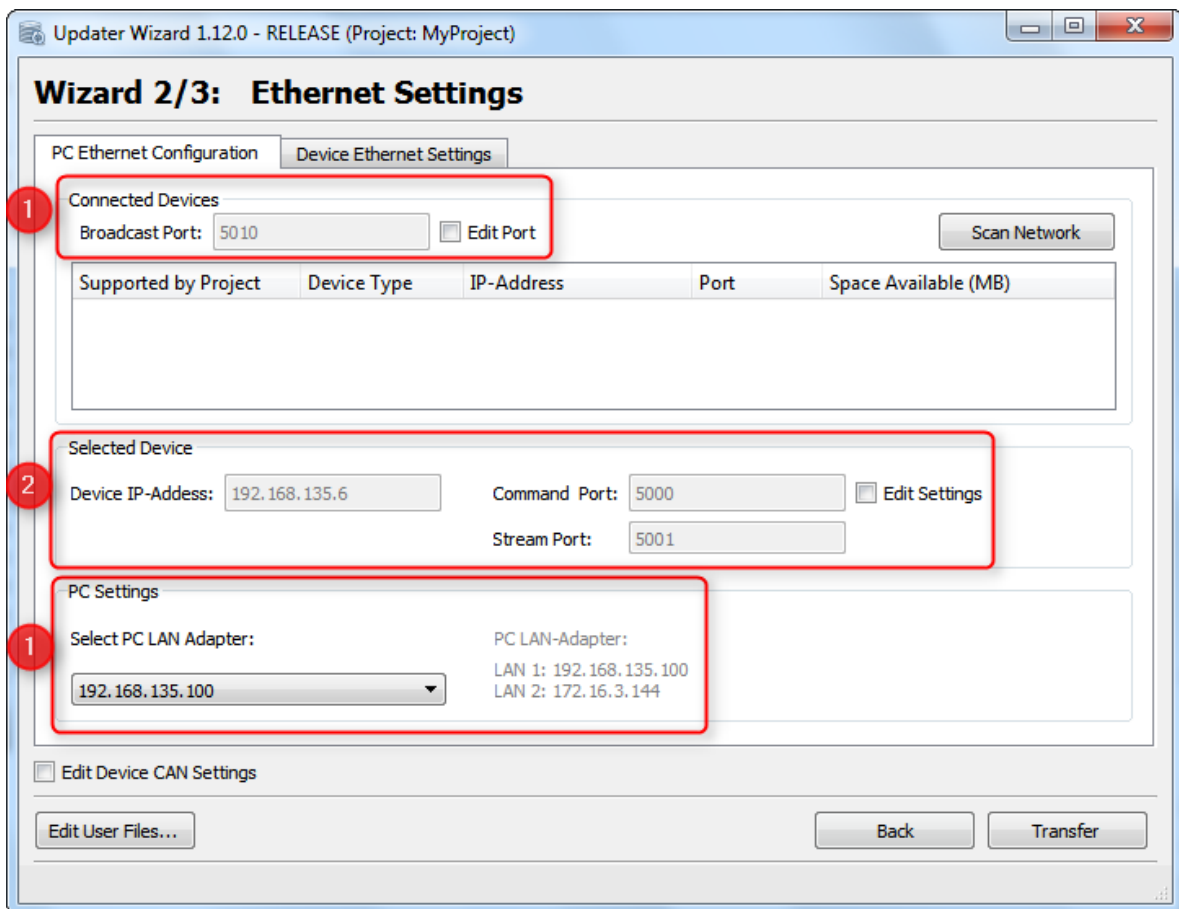
1. **Broadcast Settings:**
   - ☐ *Broadcast Port*
     - ▪ Port Number for Broadcast message to find connected devices.
   - ☐ *Select PC LAN Adapter*
     - ▪ When the PC has more than 1 LAN Adapter you can select the Adapter *(Device Net)*, where the Broadcast messages should be sent to.
2. **Selected Device Settings**
   - ☐ *Device IP-Address*
     - ▪ Here you can configure the IP-Address of the device, where the project package shall be sent to.
     - ▪ If a device from the **Connected Devices List** is chosen, the IP-Address will be set automatically.
   - ☐ *Command Port*
   - ☐ *Stream Port*
     - ▪ These Ports will be used for Ethernet communication between **HostUpdater** on PC and **ClientUpdater** on device.
     - ▪ If a device from the **Connected Devices List** is chosen, the Ports will be set automatically.
     - ▪ Only the **Command Port** can be edited, the **Stream Port** will be set automatically. **(Command Port + 1)**

Device Ethernet Parameter
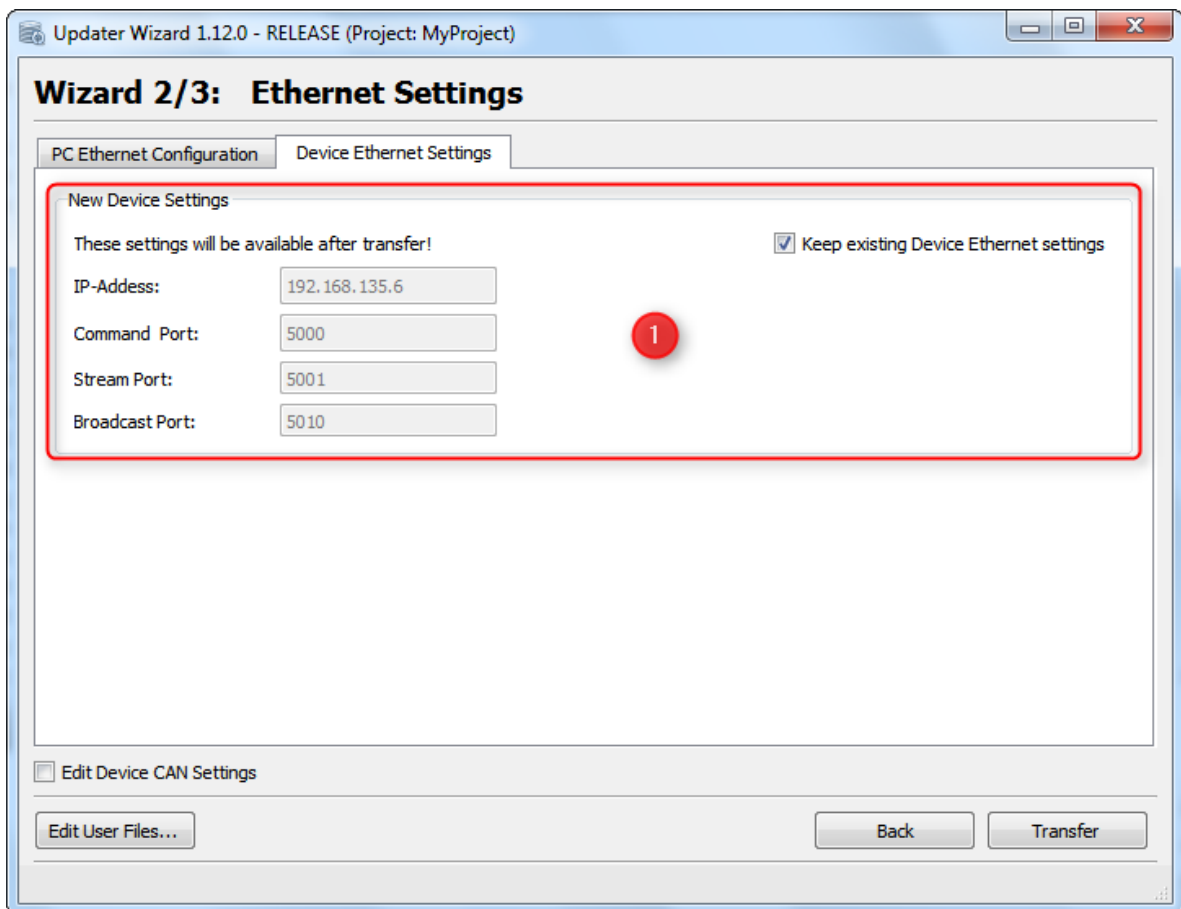
- **New Device Settings:**
  - ☐ *New Device IP-Address*
    - The new IP-Address for the device.
  - ☐ *Command Port*
  - ☐ *Stream Port*
    - New Port Numbers, which will be used for Ethernet communication between **HostUpdater** on PC and **ClientUpdater** on device.
    - Only the **Command Port** can be edited, the **Stream Port** will be set automatically. **(Command Port + 1)**
  - ☐ **Broadcast Port**
    - New Broadcast Port, which is used to scan the network for connected devices.
  - ☐ **Keep existing Device Ethernet settings**
    - Keep this checkbox enabled, when you don't want to change the ethernet settings for a device.

Ethernet Transfer Page

The Ethernet Transfer Page shows the current transfer progress and if the current transfer finished successfully or with an error.
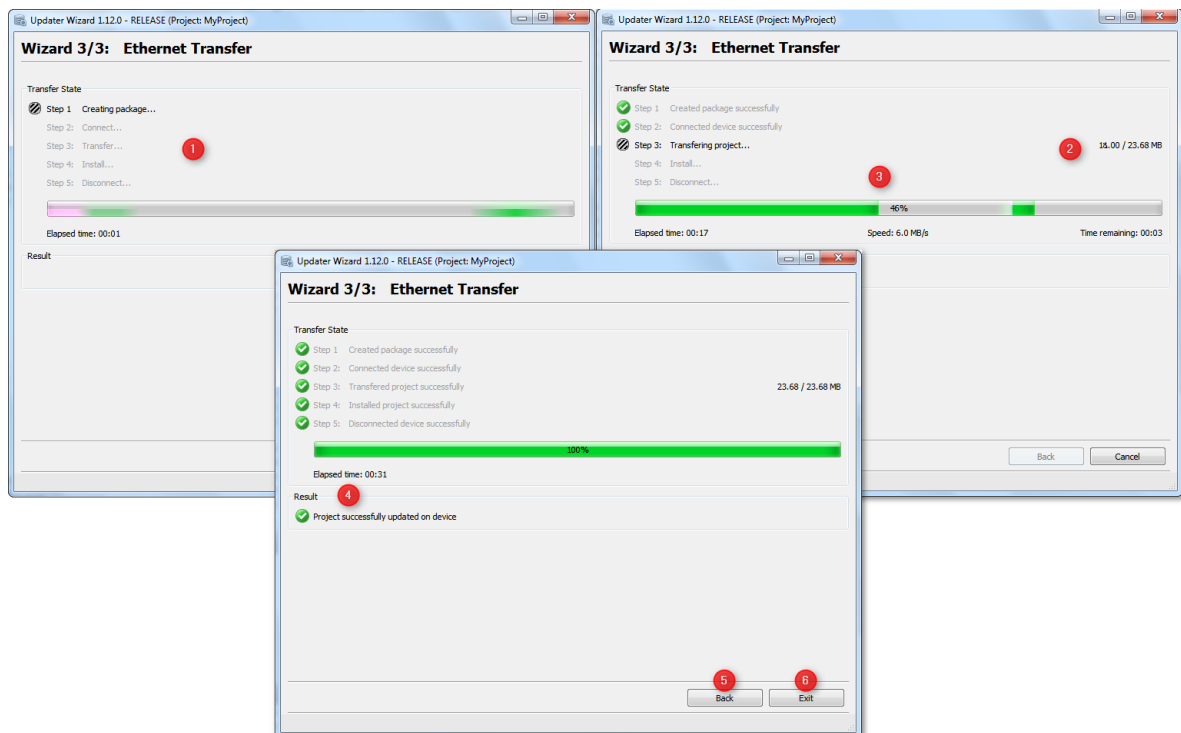
Legend:

(1) On the left side you can see the current step which is permformed. Last Step with black font color => current performed step (here Transferring Project), grey font color steps which have not be performed yet.
(2) Transferred Bytes and Project Package size. **(3)** Progress bar with transferred bytes in percentage, Elapsed Time, Transfer-Speed and Time Remaing.
(4) Shows the result of the Transfer. If the transfer fails, an error code and error message are shown.
(6) Back button which is disabled during active transfer. When transfer finishes, the button is enabled.
(7) Exit/Cancel button. During transfer button is renamed to **"Cancel"**. When the transfer finishes, the button is renamed to **"Exit"**.

If the application is closed via **"X"** button during active transfer, a dialog is shown with a warning that the current transfer will be aborted.

Adding User Files to Project Package

On the pages **USB Settings**, **CAN Settings** and **Ethernet Settings** it is possible to open the page **'Edit User Files'**. On this page you can add additional files, which shall be transfered to the device with the current project.
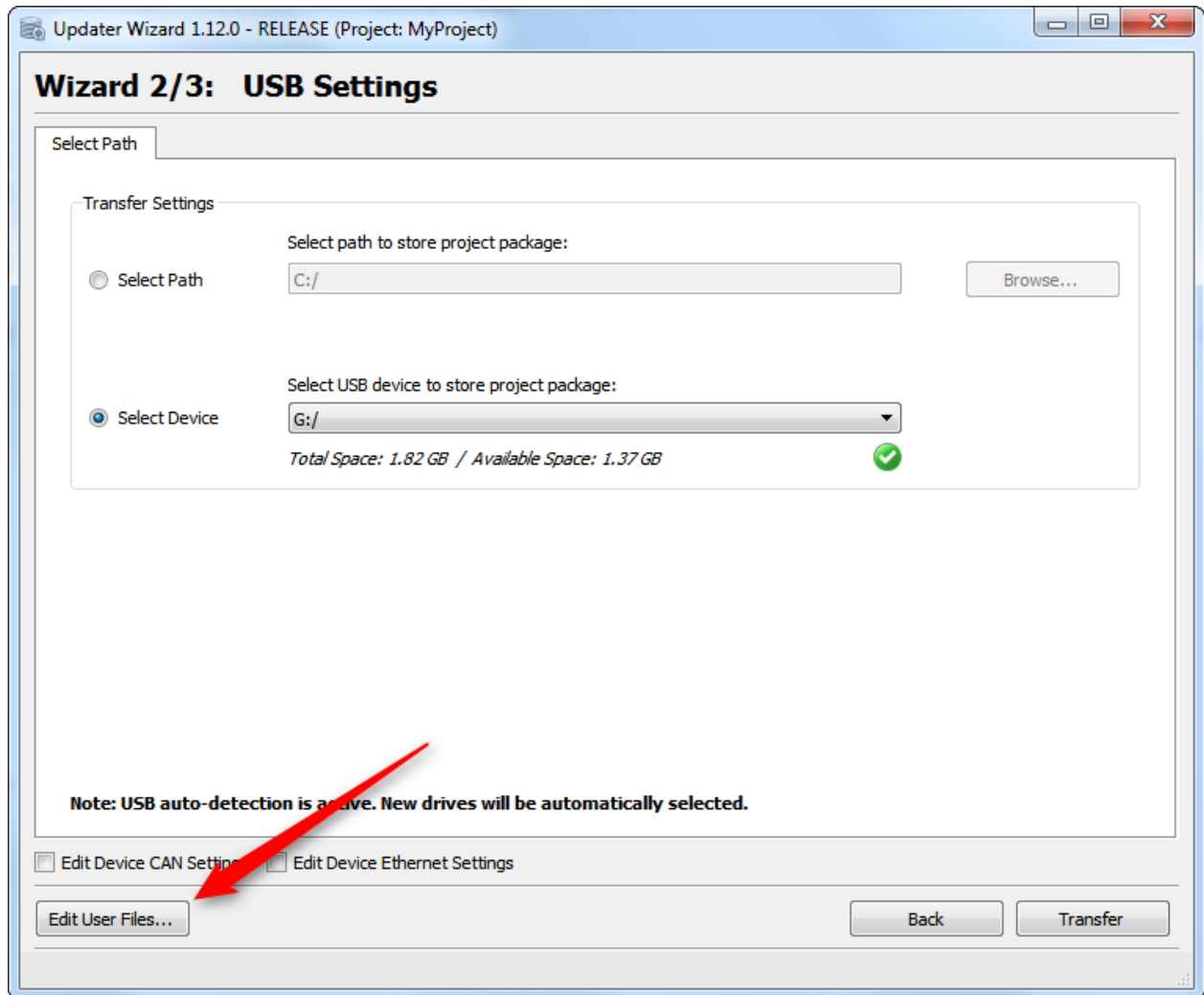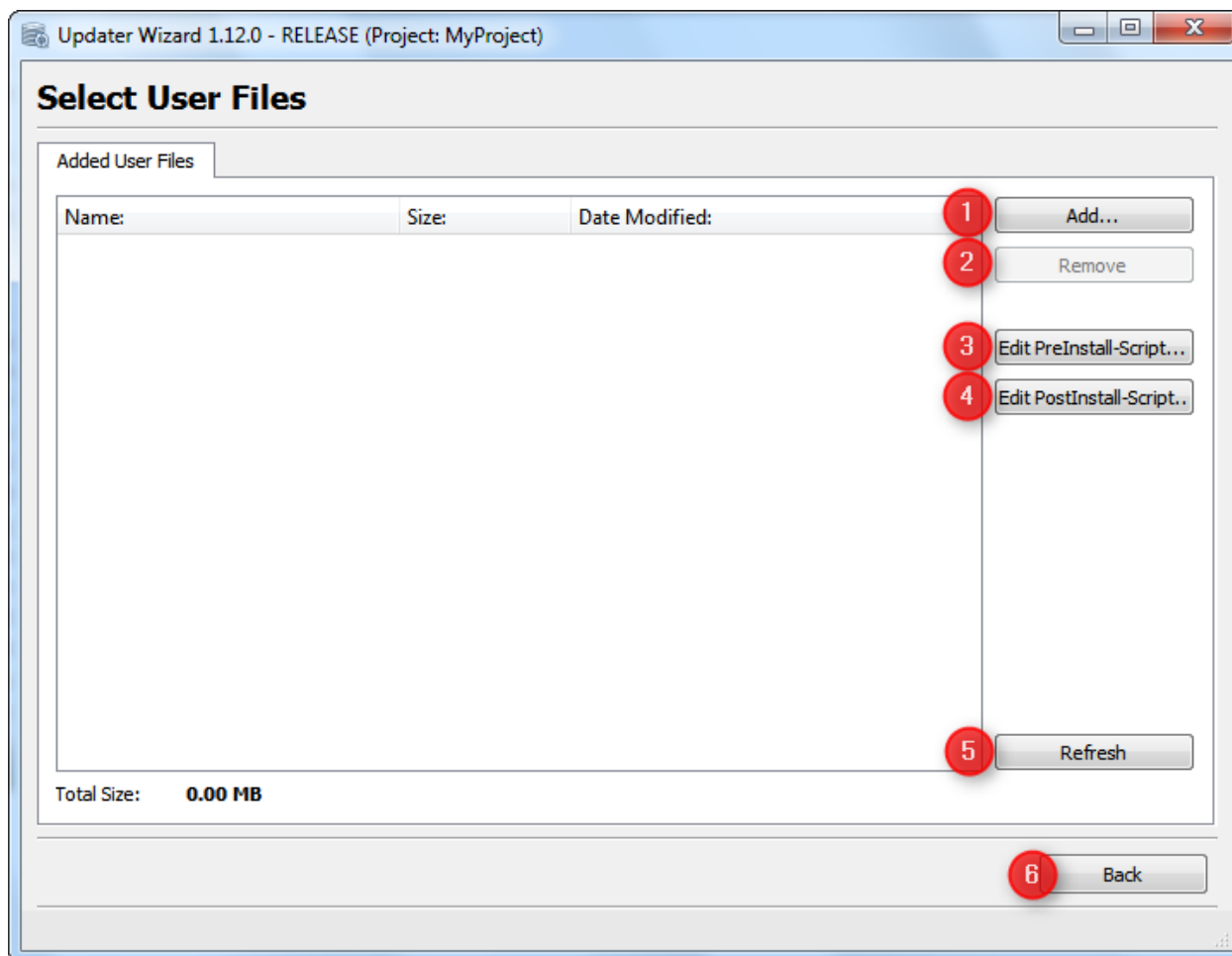
Legend:

(1) Add.. button opens a file browser to selecte a file which shall be added to the project package.

⚠	**NOTE:**
	The user is responsible to delete unused files manually (by console or by using the "post-/preinstallscript.sh")

(2) Remove button removes selected files from the list. (Only activated if at least one file is selected in the list).

(3/4) Opens the **Pre/Post-Install** scripts in the default application for shell scripts.

(5) If the file list is not updated properly, you can refresh the view manually with **Refresh** button.

(6) Back goes back to the last settings page.

Add Files

When the **Add...** button is clicked, a File Chooser dialog is opend where you can select a file which shall be added to the user files folder.
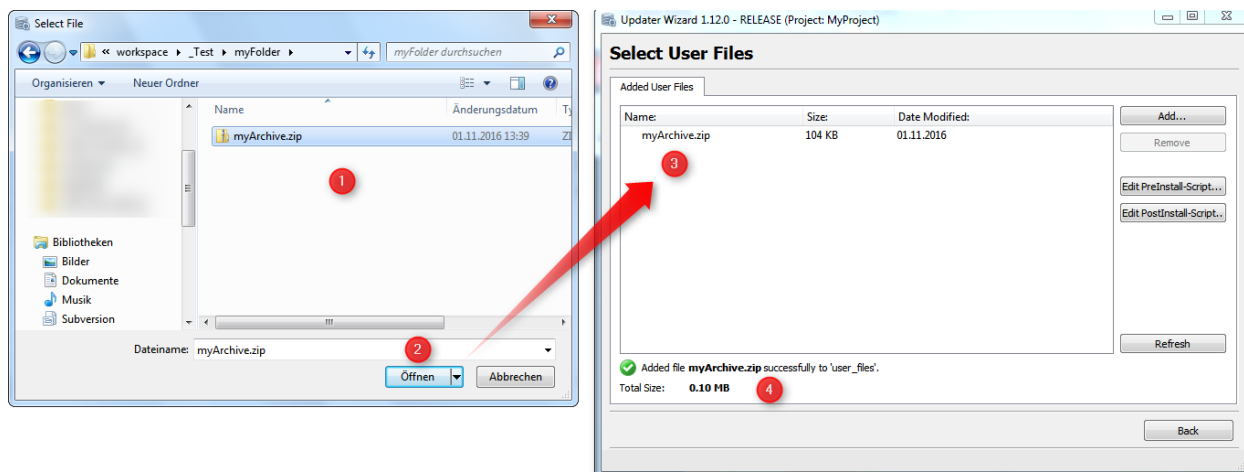
Legend:
(1) Select a file which shall be added to project package.
(2) Click on **Open** to add the file.
(3) The file list view shows all files which are added to the **user_files** folder.
(4) The green check shows that the file could be copied to the folder, and below the size of all files shown.

Remove Files
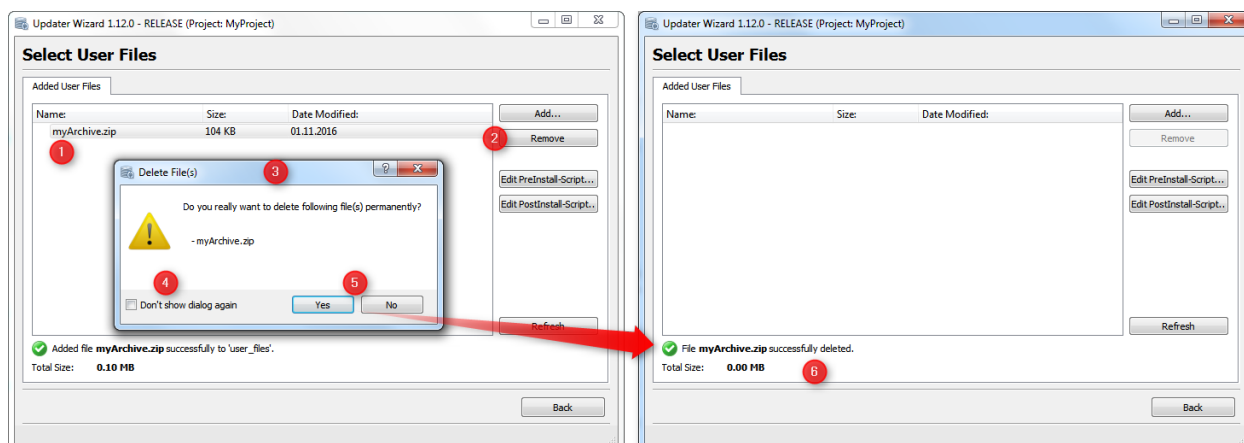
Legend:
(1) Select a file which shall be removed from **user_files** folder.
(2) Click on **Remove** button.
(3) A dialog **Delete File(s)** is shown, where you have to confirm the delete operation.
(4) If the dialog shall not be shown again you can check the checkbox **Don't show dialog again**.
(5) With click on **Yes** the file will be deleted, click on **No** closes the dialog without an action.
(6) After click on **Yes**, the file will be deleted and the files view is updated.

⚠️	**NOTE:**
	All files in the user_files folder will be copied to the directory **/op/pclient/projekte/default_prj** on the device.

⚠	NOTE:
	With the **postinstall_script.sh** you can, it is possible to edit the file after download process.

Setting Files

PC (HostSettings.ini)

The *HostSettings.ini* file saves the settings (**USB Settings, Device CAN Settings and Device Ethernet Settings**) which can be changed in the *Updater* application. **BUT** the settings will only be saved if the Transfer (to **USB drive**, or via **CAN/Ethernet** was successful).
The *HostSettings.ini* file is saved in the folder **updater**[643] of the **project directory**.

Do not delete or modify this file!

Device (ClientSetting.ini)

Whenever a project is transferred to a **USB drive** or to the device via **CAN bus**, the *ClientSetting.ini* will be created. The file is saved to the project package (e.g. *MyProject.ud4*) which will be transmitted.
The *ClientSetting.ini* file contains the **project name** and the **Device CAN/ETH Settings**[618].
The *ClientUpdater* on the device reads this file set up the **CAN Interface** appropriate.

Do not delete or modify this file on the device!

Updater Folder

The subfolder *updater* which is located in the **project directory**, contains the following files:
- *HostSettings.ini*
  - ☐ This file contains the **USB** and **Device CAN Settings**.
- *preinstall_script.js* (included in Project Package[644])
  - ☐ This script is executed during Install step on the device **before** the *project is installed.*
  - ☐ This file can be edited by the user.
- *postinstall_script.js* (included in Project Package[644])
  - ☐ This script is executed during the Install step on the device **after** the *project was installed.*
  - ☐ This file can be edited by the user.
- folder *user_files* (include in Project Package[644])
  - ☐ All files in this folder will be added to the *Project Package* and will be transmitted to the device.
  - ☐ The files will be copied during the project install step to the following location on the device:
    - **/opt/pclient/projekte/default_prj/**

☐ The files are already available in the <u>postinstall_script.sh</u>[☐]⁶⁴⁴ at the above location

> ⚠️ **NOTE:**
>
> The folder **user_files** may only contain **less than 8 files**. If the folder contains more than 8 files only the first 8 files will be transmitted. **File names may** not contain **empty spaces or UTF8 characters like ('ä', 'ö', 'ü', or 'é')**. **Subfolders in** **user_files won't be transmitted** to the device.
>
> Do **not delete** any of these files or folders, otherwise **no Project Package** can be created anymore:
> - **preinstall_script.js**
> - **postinstall_script.js**
> - folder **updater**

Project Package

For each transfer, a **Project Package** (*.ud4) file is created, which contains all the important files for a project update.
The **Project Package** contains:
- **Project archive** (terminal_files.zip)
- **Preinstall_script.js**
- **Postinstall_script.js**
- **ClientSettings.xml**
- files from **user_files** folder in **<u>Updater folder</u>**[☐]⁶⁴³ <u>.</u>[☐]⁶⁴³

Variable @EnableUpdater

With this variable the **ClientUpdater** on the device can be **enabled** or **disabled**.
Additionally, you can set the **active interfaces**, which allows you to update the project on the device.
The following values can be set:
- Updater is **disabled**
- **0x0**   => <u>**NO**</u> project update can be performed.
- Updater is **enabled (Single Inteface)**
- **0x1**   => Project update can <u>**ONLY**</u> be performed via <u>**USB drive.**</u>
- **0x2**   => Project update can <u>**ONLY**</u> be performed via <u>**CAN bus.**</u>
- **0x4**   => Project update can <u>**ONLY**</u> be performed via <u>**Ethernet.**</u>
- Updater is **enabled (Multiple Intefaces)**
- **0x3**   => Update possible via <u>**USB drive**</u> or <u>**CAN bus.**</u>
- **0x5**   => Update possible via <u>**USB drive**</u> or <u>**Ethernet.**</u>
- **0x6**   => Update possible via <u>**CAN bus**</u> or <u>**Ethernet.**</u>
- **0x7**   => Update possible via <u>**USB drive,**</u> <u>**CAN bus**</u> or <u>**Ethernet.**</u>

@EnableUpdater value	Update over USB	Update over CAN	Update over Ethernet
0	☒	☒	☒
1 (default)	☑	☒	☒
2	☒	☑	☒
3	☑	☑	☒
4	☒	☒	☑
5	☑	☒	☑
6	☒	☑	☑
7	☑	☑	☑

⚠️	**NOTE (DANGER):**
	Please be aware, if you set the pre-defined variable to **'0x0'**, the Updater on the device is disabled and **NO** project update is possible anymore. Please always add a possibility to change the value of the pre-defined variable in your project.

CAN Transfer Checklist

The CAN update is disabled by default on the device. This is important to protect the CAN network and to avoid undesired CAN traffic.

If you want to activate update over CAN please follow these steps:
1. **Common:**
   a. Please first read the chapter: **Disclaimer about using CAN**[618]
2. **Project settings:**
   a. If CAN shall be activated, the variable **@EnableUpdater**[644] must be set properly in the project. The value must be set to *0x2* or *0x3 (Please have a look here*[644] *)*
3. **Updater settings:**
   You need to enable the update by changing the @EnableUpdater variable. If the existing project on the device allows such a setting, you can make it on the device. Otherwise:
   a. The first project update has to be done via **USB update**[619] . **Here**[620] you have to check or edit the CAN settings you want to use.
   b. Transfer project to USB drive
   c. Update device via USB.

Now the *ClientUpdater* on the device should be **enabled for CAN** and the **CAN Update**[625] mode can be used for future project updates.

Errors and Warnings

If an error occurs in the *Updater* application, the warning is shown in the status bar.
By clicking the icon or label in the status bar, a dialog with detailed information is shown.
All **error messages** will **disable** the **Transfer** button until the error is resolved.
Warnings do not disable the **Transfer** button, but it is recommended to check why the warning is shown.

Common

Warning reading HostSettings.ini file
If the *HostSettings.ini* file can not be read by the *Updater*, a warning is shown. The warning shows which parameter could not be read properly.
In this example, the "**Baud Rate**" was removed in the settings file*.*
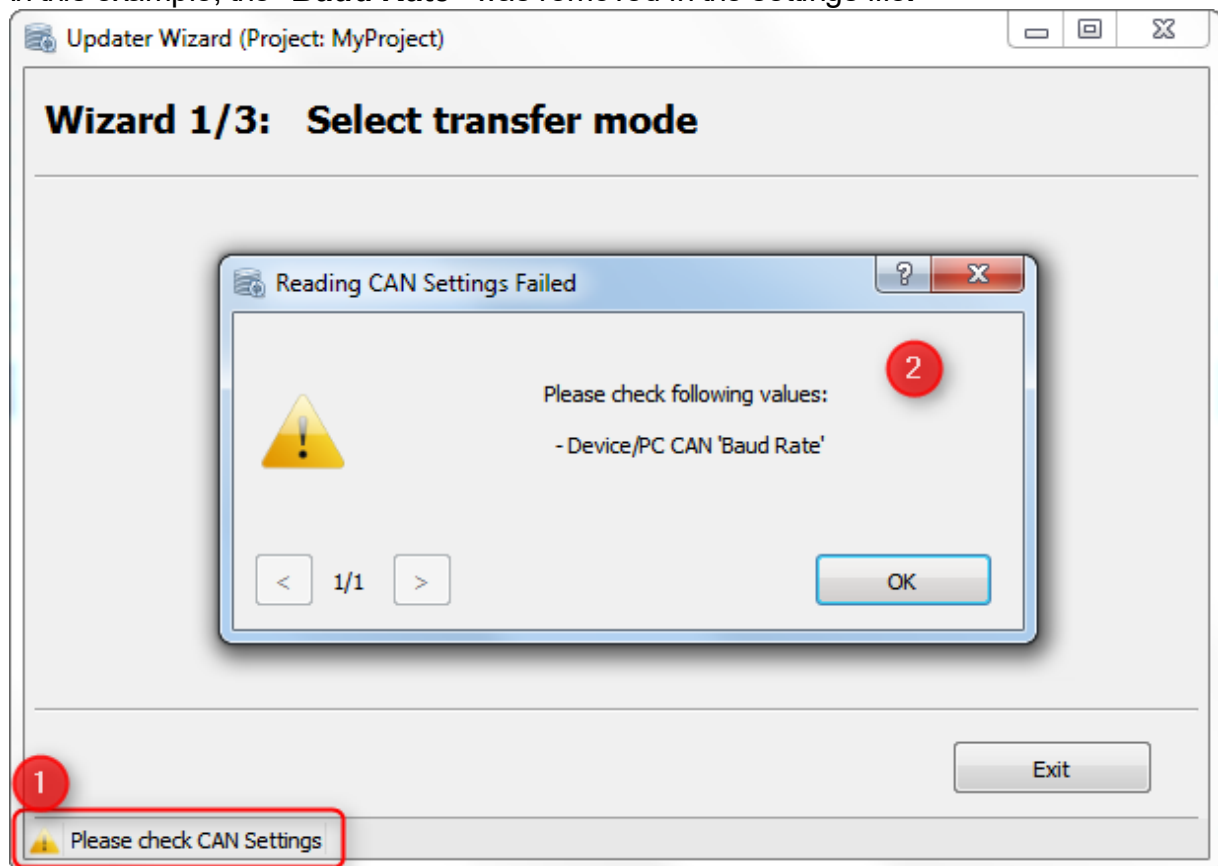
(1) By pressing on the warning icon in bottom left corner a dialog appears with detailed information.
(2) A dialog is shown that the user shall check the **Baud Rate** in the **CAN Settings** for Device and PC.

Missing Files (postinstall_script.sh, preinstall_script.sh, terminal_files.zip)
If one of these files is missing, an error is shown, and the **Transfer** button will be **disabled**. These files are mandatory for the updater to install the **Project Package** on the device. If the file(s) are available again, the message will disappear and **Transfer** button gets **enabled** again.
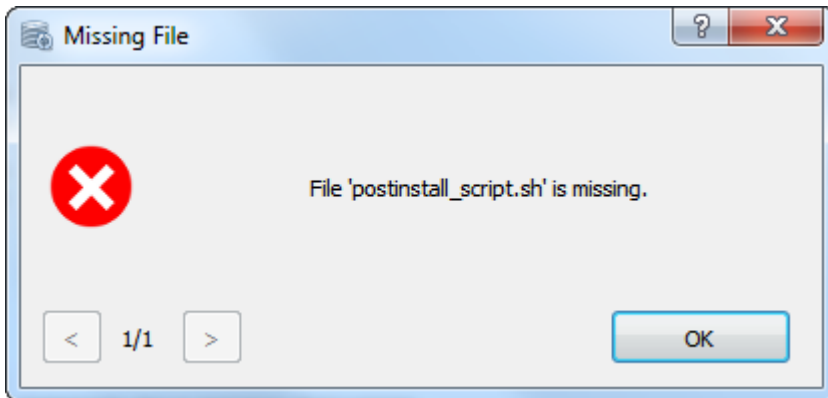
Too many files in 'user_files' folder

If more than **8 files** are copied to the *user_files* folder, this message will appear.
In this case only the first **8 files** will be copied to the **Project Package**[644] .
Please also have a look **here**[643] .
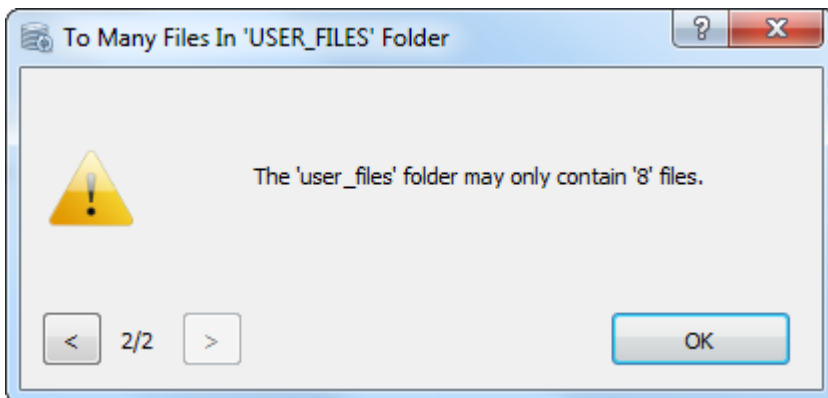
Subdirectories in 'user_files' folder

If subdirectories are copied to the *user_files* folder, this message will appear.
In this case the subdirectories won't be copied to **Project Package**[644] .
Please also have a look **here**[643] .
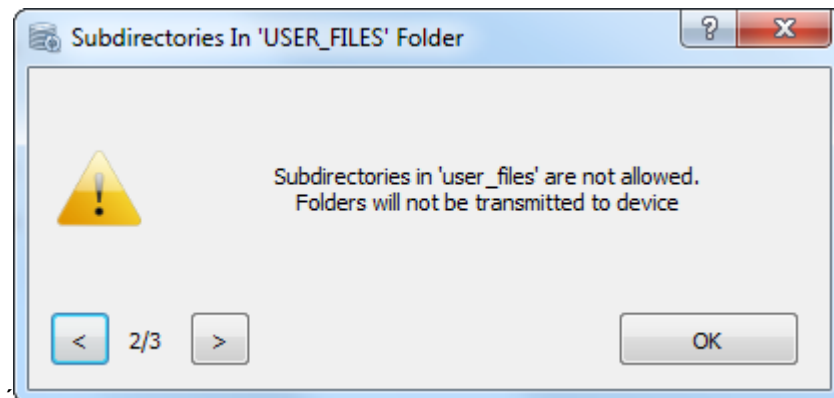
No terminal_files.zip

If no terminal_files.zip file is located in the project folder, this error appears.

To create the file, close the wizard and save the project. This will create the file.
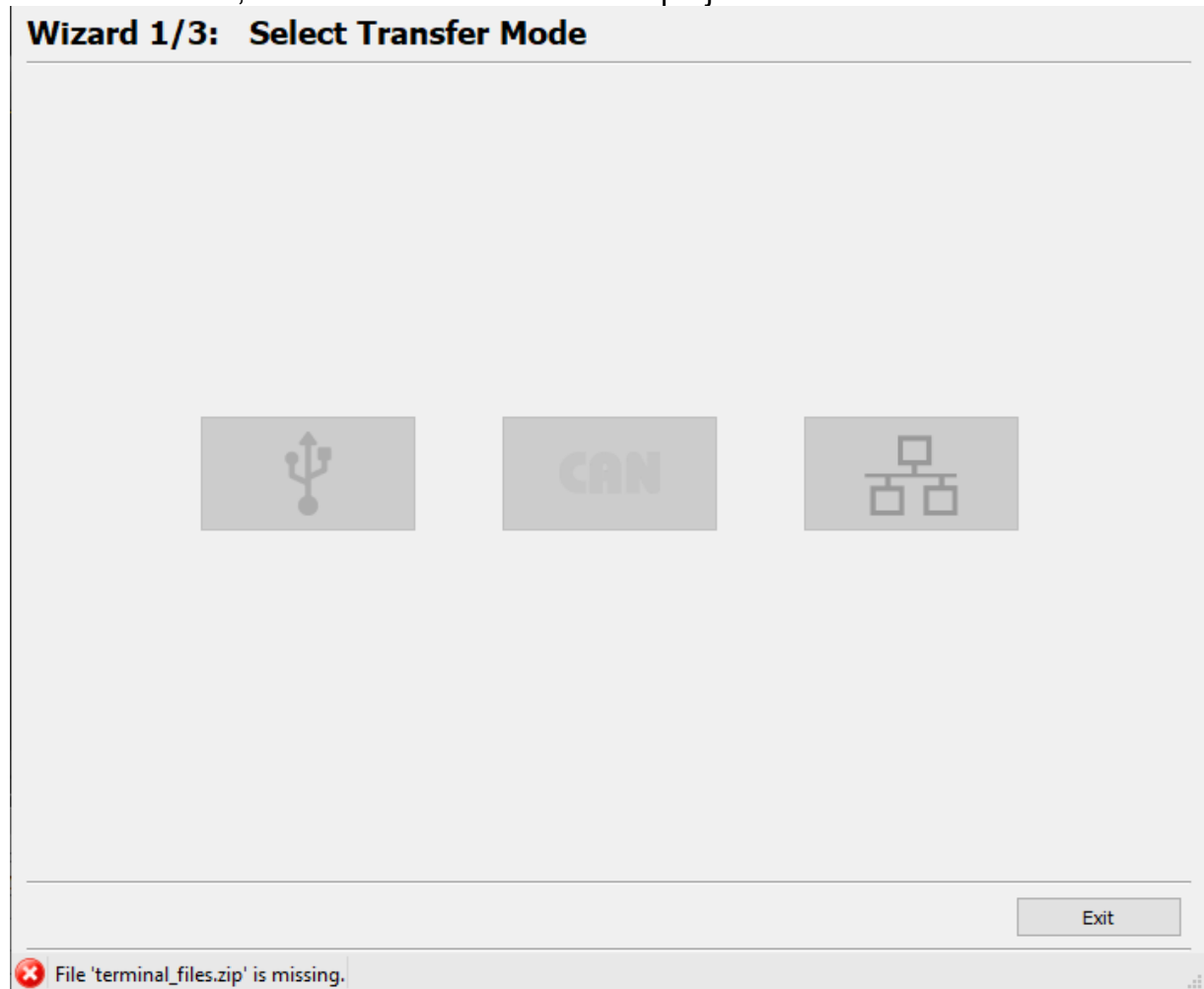
CAN Settings

PC Duplicate CAN IDs
CAN IDs have to be unique, otherwise the CAN communication does not work properly.
If an error occurs, the **Transfer** button is **disabled**.
This error is only shown on the **CAN Settings Page**.

Device Duplicate CAN IDs
CAN IDs have to be unique, otherwise the CAN communication does not work properly.
If an error occurs, the **Transfer** and **Back** buttons **are disabled**.
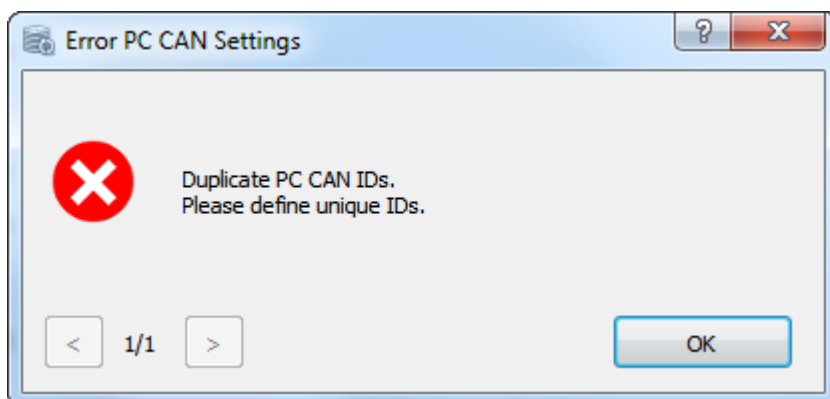
PC Empty CAN ID

CAN IDs must have valid values, otherwise the CAN communication does not work properly. If an error occurs, the **Transfer** button is **disabled**.
This error is only shown on **CAN Settings Page**.

Device Empty CAN ID

CAN IDs must have a valid value, otherwise the CAN communication does not work properly. If an error occurs, the **Transfer** and **Back** buttons are **disabled**.
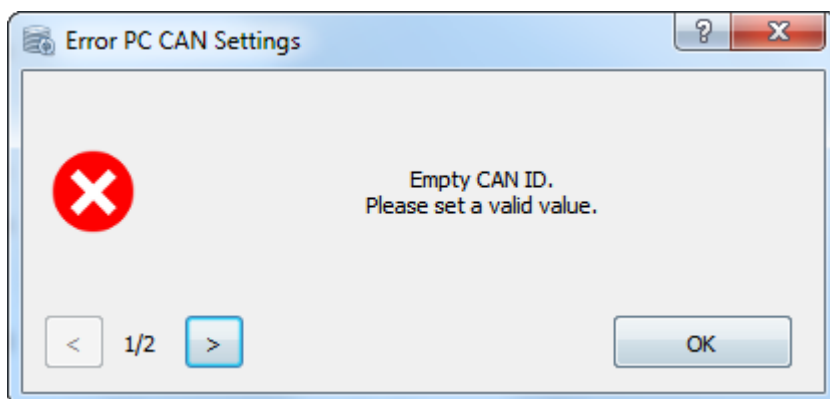
USB Settings

Invalid drive

If the **USB drive** is not available anymore, this error message will appear and the **Transfer** button gets **disabled**.
The currently chosen drive is checked continuously, which means if the drive is available again the message will disappear and the **Transfer** button gets **enabled**.
This error is only shown on the USB Settings Page.
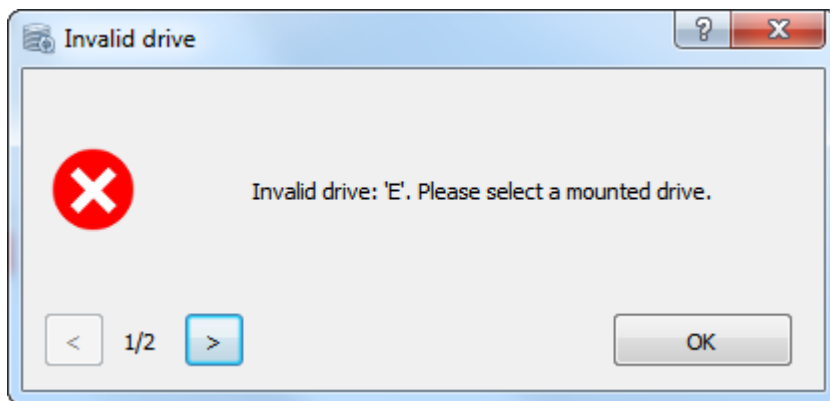
Ethernet Settings

Error -93
If error -93 is displayed during an Ethernet project transfer:
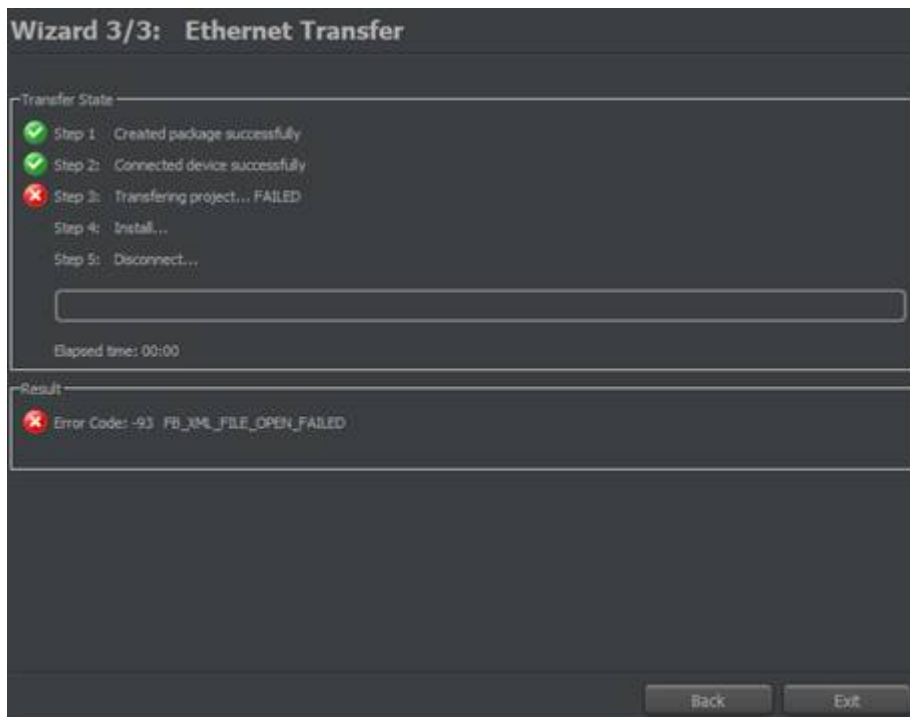
The solution might be to allow the updater wizard executable file, HostUD.exe in your firewall. Upon the first use, a popup like this should be displayed (from Windows Defender):
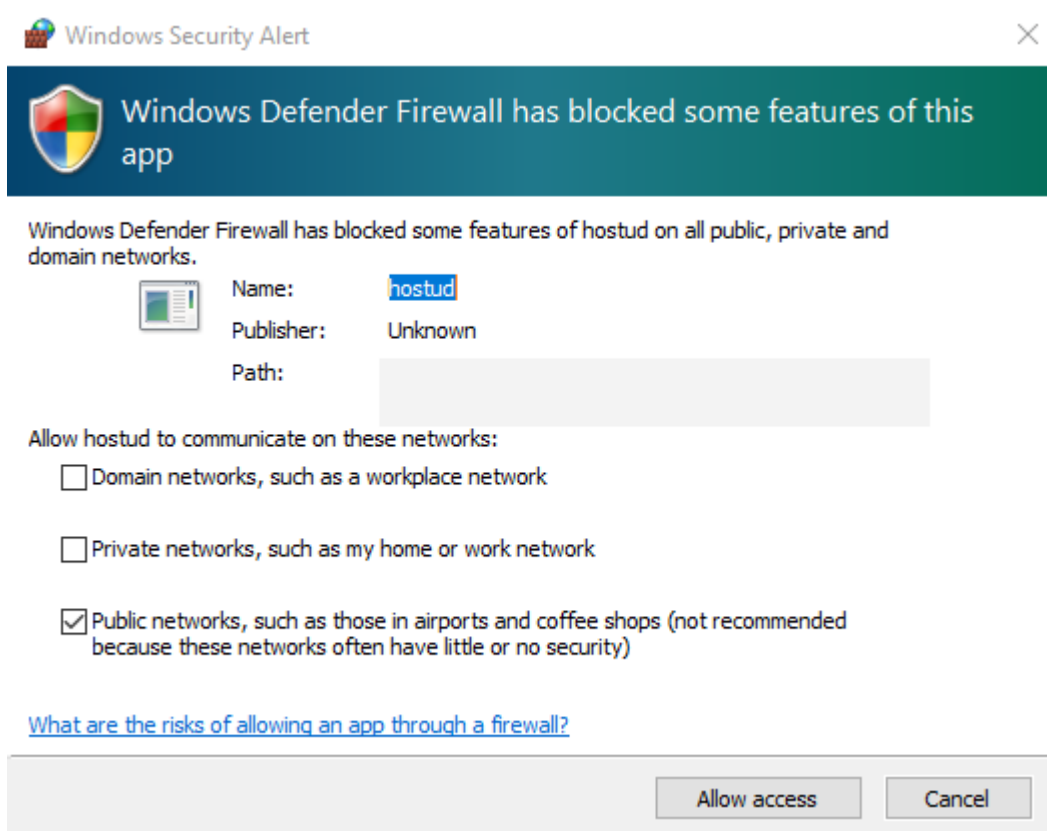
Please check all network types and allow access. Similar settings should be made for other firewall solutions.

Command Line Interface

It is possible to start the updater program through a command line.

The updater executable is located in
<program_path>\application\extracted_components\updater
named HostUd.exe
There is an example batch file showing how the updater can be started with the user interface.
It is also possible to start the updater without a user interface, allowing for an automatic project download to a USB stick or via CAN.

These two lines are mandatory in the batch file for the executable file to find all necessary libraries:

set PATH=%PATH%;..\shared_libraries\
set QT_PLUGIN_PATH=%QT_PLUGIN_PATH%;..\shared_libraries\

The executable has the following parameters:

- ProjectPath (required) - the path to the project that should be downloaded - example: ProjectPath=C:\Workspace\Projects\customer_project2
- GUI (optional) - flag whether the updater should start with the gui (true) or silently (false) - example: gui=false
- AppName (optional) - A name that will be displayed in the gui title (only considered if gui=true) - example: AppName=MyPersonalUpdater
- ShowCANWarning (optional) - Flag whether the CAN warning screen should be shown in the gui (only considered if gui=true) - example: ShowCANWarning=true
- interface (CAN/USB)- Defines where the project should be downloaded to (when no gui is used) / which target should be selected in the gui (if the gui is used) - example: interface=USB
- path - Defines the target path if the program is used without gui and USB is set as target - example: path=D:\

Example for a download to USB without gui:

HostUD.exe projectpath=C:\Workspace\Projects\customer_project2 gui=false interface=USB path=D:\

FAQ

How can I add my own files to a Project Package?

You can add your own files to the folder user_files of the [updater folder](#)[643] . The updater folder is a subdirectory of the project directory.
All files which are copied to user_files will be added to the [Project Package](#)[644] .
With the shell scripts postinstall_script.sh and preinstall_script.sh you can edit your files on the device.
You can also add user files in GUI, see here: [Add User Files To Project Package](#)[639]

How can I figure out which CAN settings the ClientUpdater uses on the device?

The settings for the ClientUpdater are stored in the the file **ClientSetting.ini** in the folder **/opt/ud4/bin/**

Why does the CAN Transfer not work?
- ☐ Please first read the [**CAN Transfer Checklist**](#)[645] before you try to transfer a project via **CAN mode**.
- ☐ If all steps are done and the CAN Interface on the device is enabled, please check the CAN network:
  1. Check if the device is connected to the CAN network (check if CAN Port 1 and/or CAN Port 2 is connected).
  2. Check if the baud rate of the ClientUpdater is set properly.
  3. Check the bus status of CAN network.
  4. Check the ID settings of the Updater (Client and Host)
- ☐ Check if the ClientUpdater is running on the device and/or restart device.

Is it possible to update more than 1 device parallel via CAN?

No, it is not possible. Please connect only 1 device to CAN network when a project update is performed.

Why is the project not updated on the device via USB drive?

1. Please check if the the folder "updater" is created on the USB drive properly as described **here**[619].
2. Please check the value of the predefined variable **@EnableUpdater**[644] .

10.2   OS / PClient Update

This feature provides the copying of all necessary files in an easy way to a USB drive for a device update. Read more about the update process here[768] .

You can choose 2 options:
    - PClient update and/or
    - OS update

Afterwards the USB drive can be plugged into the device. When device is booted in **service mode**, the new OS version and/or PClient version will be installed automatically. The dialog to create a bootable USB drive is located in *Menu - Update - Update PClient / OS*.
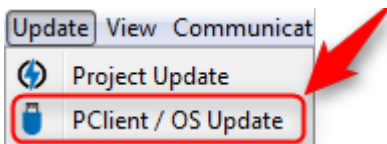
Create Bootable USB Drive Dialog

When the menu entry *Menu - Update - Update PClient / OS* is clicked, the following dialog appears:
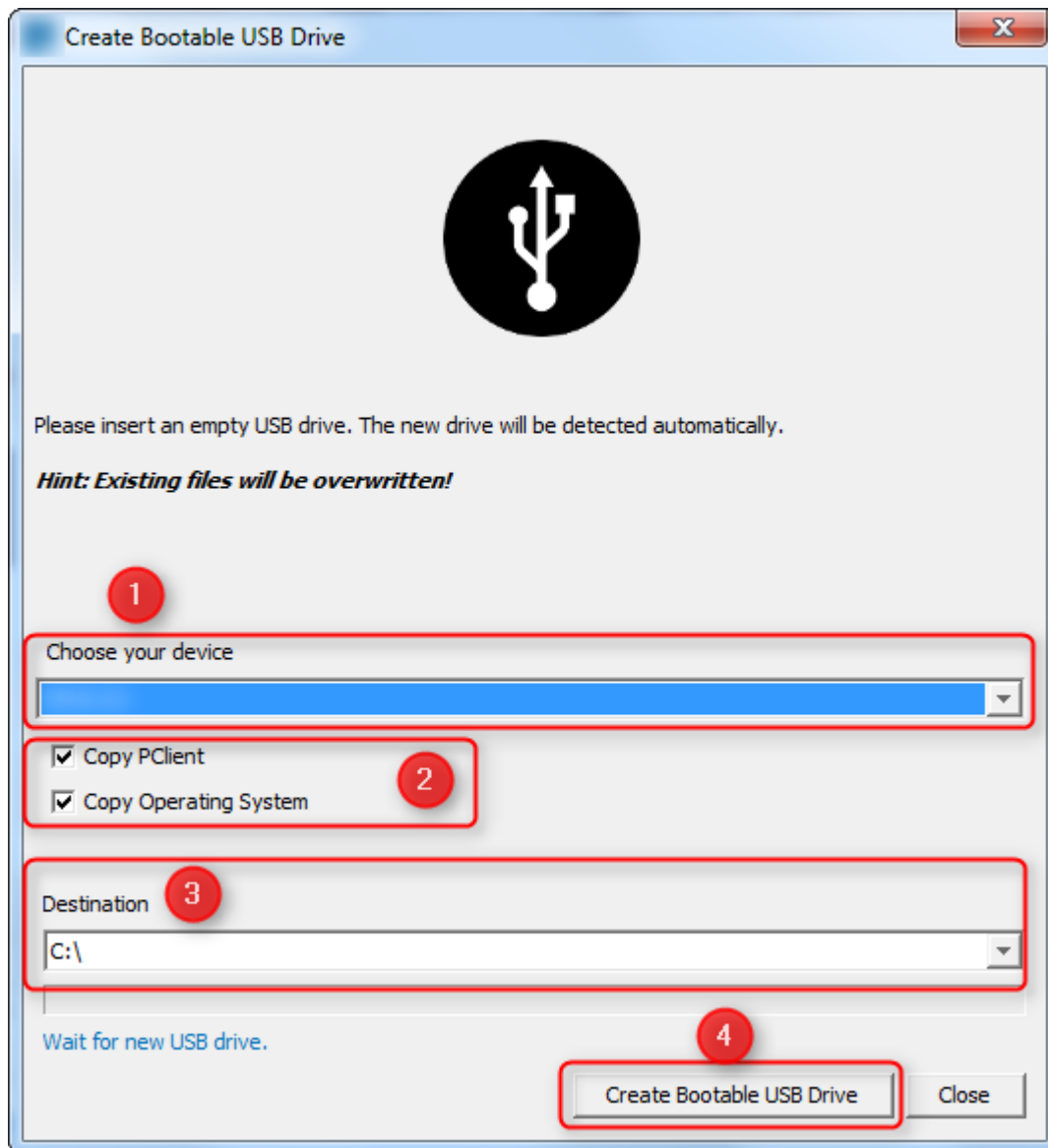
Legend:

(1) The combo box shows all available devices which can be updated.
NOTE: If no device is shown, please install the desired *PClient* module and/or the needed *OS* module(s) in the **Plugin Center**.
(2) Here you can choose the components which you want to update on the device.
Additionally, the respective versions of PClient and OS for the chosen device are displayed
(3) This combo box shows the destination drive where the update files will be copied to.
Please select the appropriate USB drive.
NOTE: If a USB drive is mounted, it will be selected automatically.
(4) When the device and the components are chosen (1-3), you can start the copy process with the button **'Create Bootable USB Drive'**.

USB Auto Detection
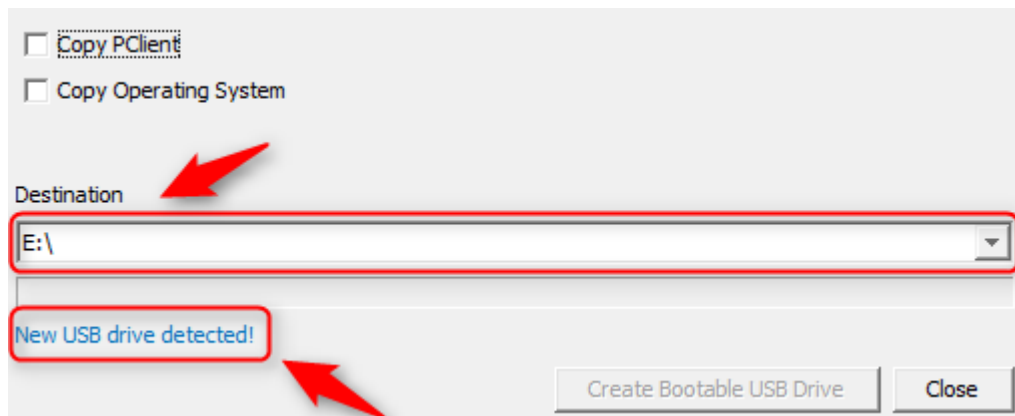
This feature provides a USB auto detection. Newly inserted usb mass storage drives will be selected automatically.
You can also select your desired device manually.

Copy Process
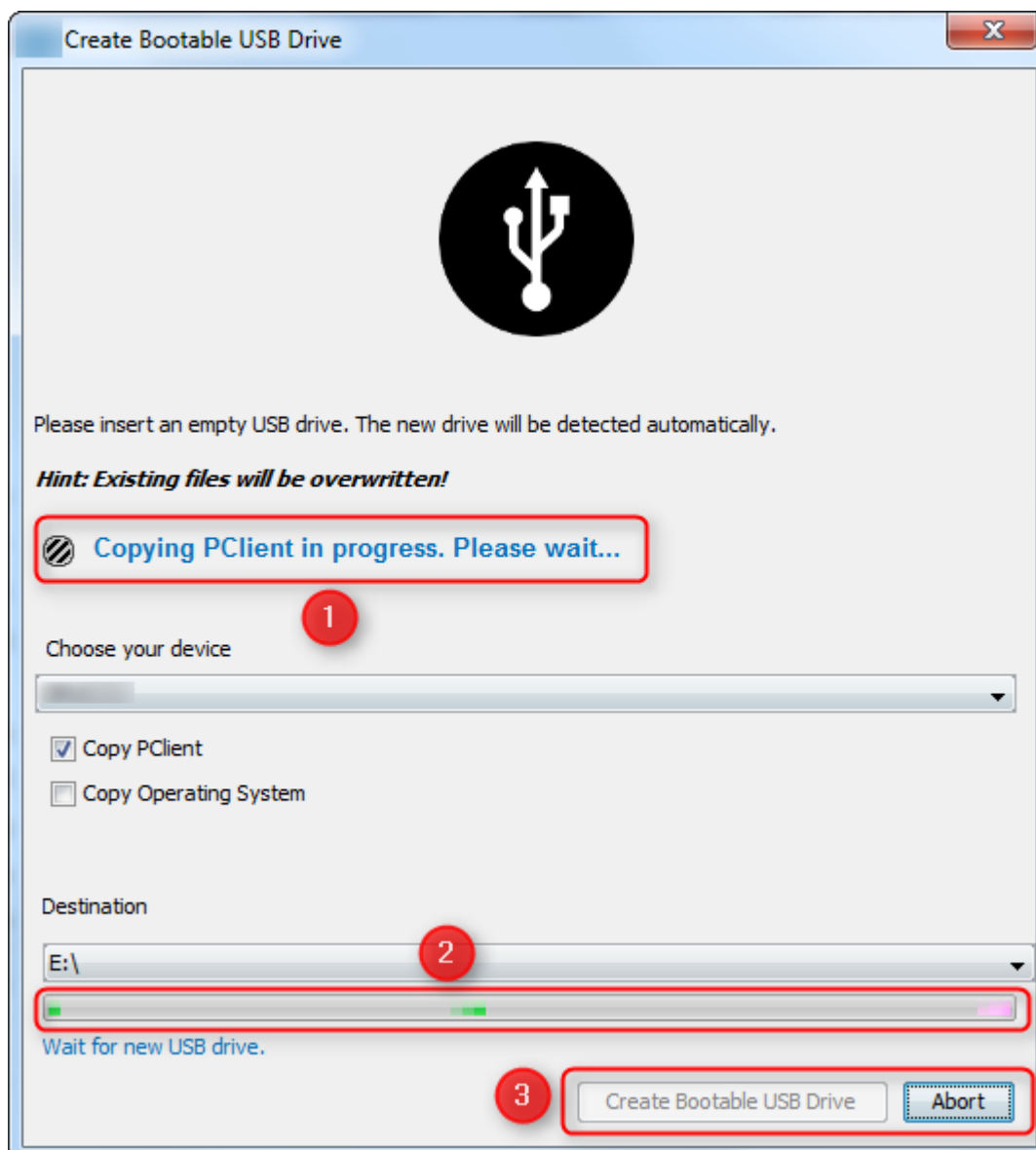
Legend:

(1) When the button 'Create Bootable USB Drive' is clicked, the current copy state is shown here.
(2) During copy progress, the progress bar is running.
(3) When the copy process starts, the button 'Create Bootable USB Drive' will be disabled.
To abort the copy process, please press 'Abort'.
Copy process successfully
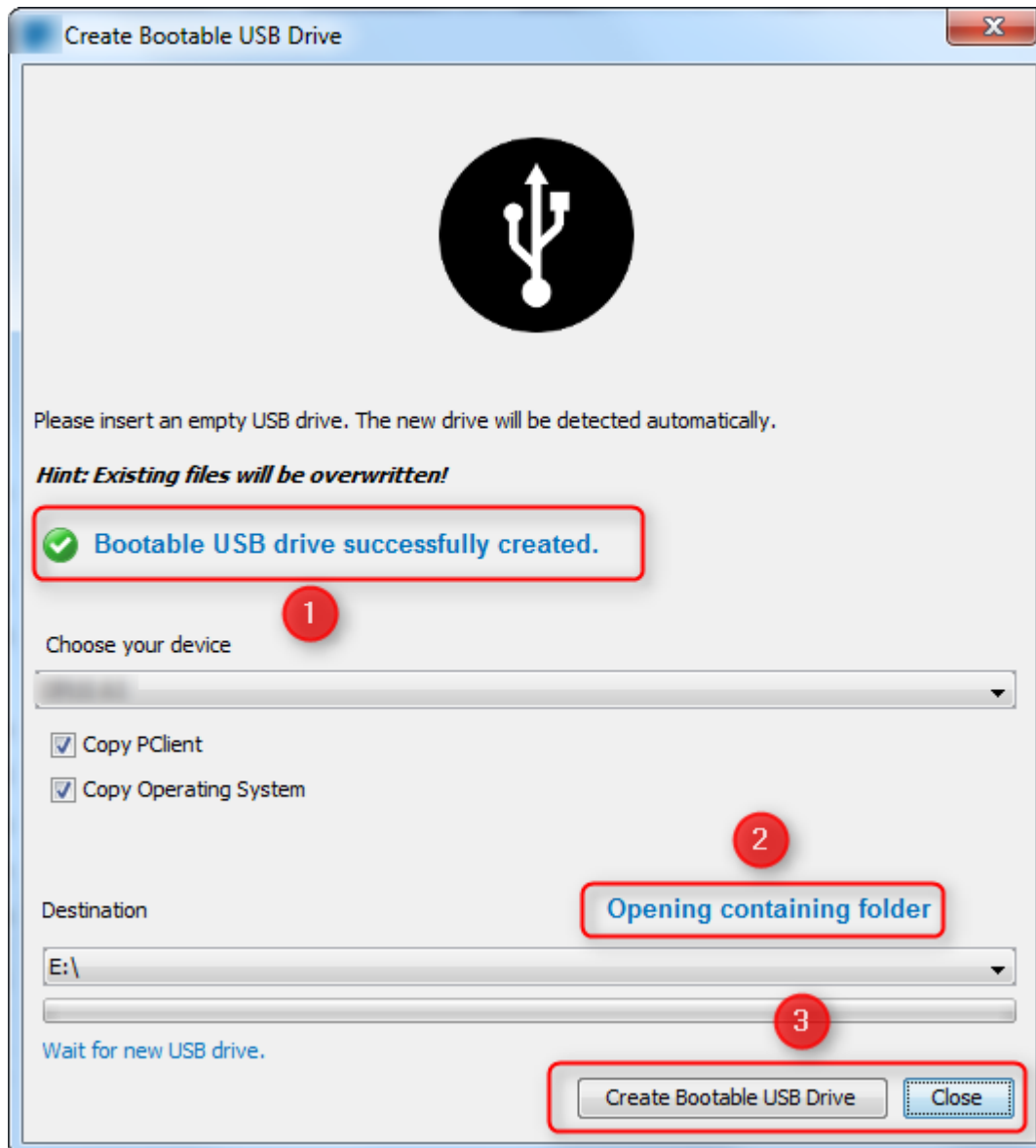
Legend:

When the copy process is finished successfully, the green check icon is visible.
(2) Here you can open the destination folder directly, after the copy process is finished.
(3) After the copy process, the button 'Create Bootable USB Drive' is enabled again.

Copy process failed
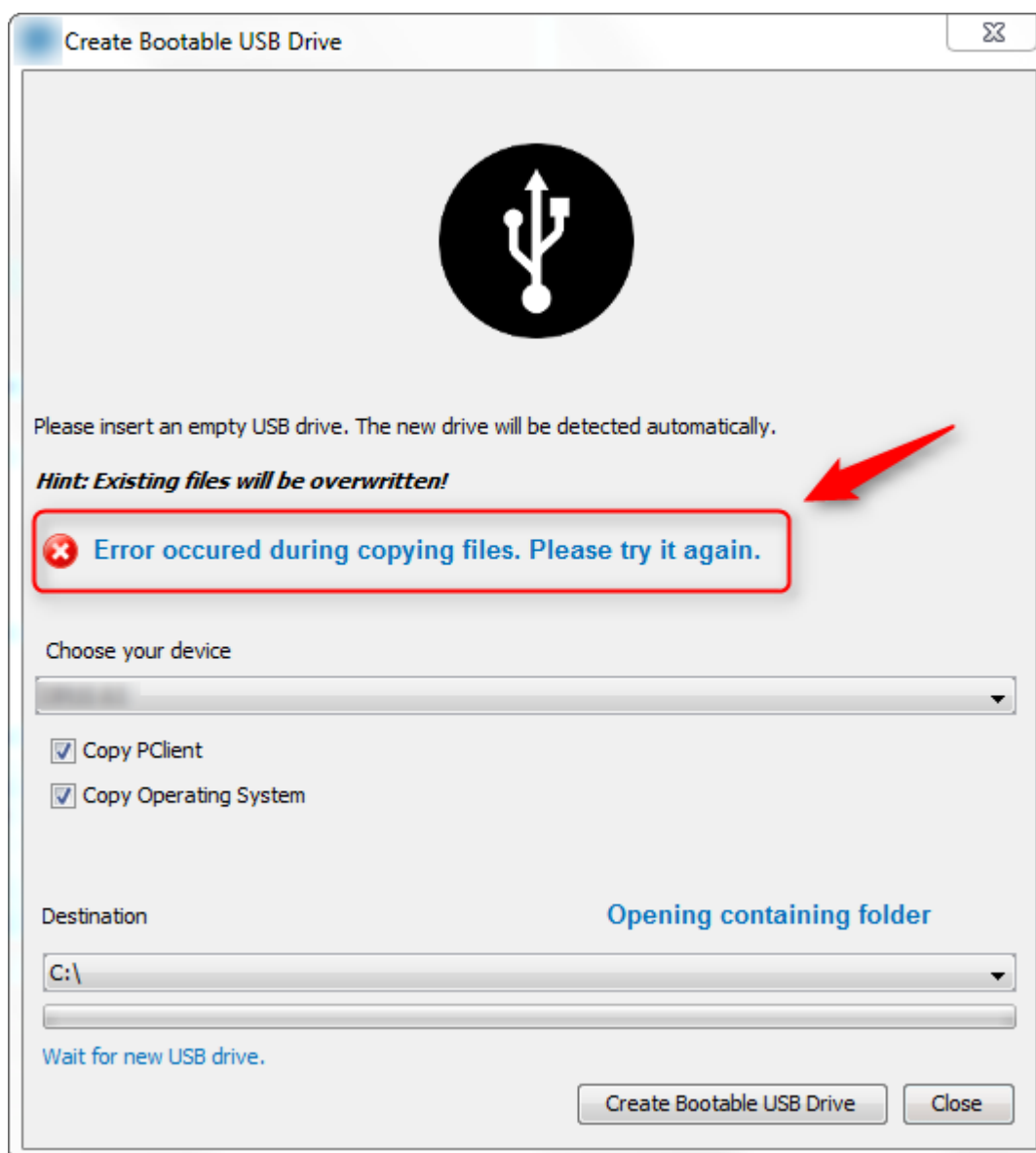
When the copy process fails, a red cross icon is visible.

NOTE:
The copy process fails e.g. if disk space is to low, or USB device is removed during the copy process.

10.2.1 Software Versions Overview

This page provides an overview of released program versions and what OS/BSP combinations should be used on the devices.

Note: Since version 1.12 the version numbers of all components of the program are the same (PC program/PClient/Downloader/etc.)
Note: When the hardware of a device has to be changed, it might be necessary to release a new OS for that device family. The table below reflects the latest OS versions that can run on all series versions of the devices (newest and oldest).

Program Version	OS Version for I.MX35 Devices	OS Version for I.MX6 EGSPRO Devices	OS Version for I.MX6 EGSCORE Devices
2021.4	2.2.6	2.3.4	1.2.2
2020.4	2.2.5 (2.2.6 for EA devices)	2.3.4.	1.1.1 (only B3)
2020.1	2.2.3 (2.2.6 for EA devices)	2.3.2	0.6.1
2019.4	2.2.2 (2.2.6 for EA devices)	2.2.0 (2.3.4 for newer devices)	-
2018.4	2.2.0 (2.2.6 for EA devices)	2.2.0 (2.3.4 for newer devices)	-
1.12.1 (Releasedate: 2017/05/05)	2.2.0 (2.2.6 for EA devices)	1.1.3	-
1.12.0 (Releasedate: 2017/04/10)	2.2.0 (2.2.6 for EA devices)	1.1.3	-
1.11.16 (Releasedate: 2016/10/28)	2.2.0 (2.2.6 for EA devices)	1.1.3	-

- program version 3.11.16
- PClient 2.15.16 => 2.15.20
- Universal Downloader 4.3.14

1.9.16 (Releasedate: 2016/10/21)or lower	1.4.7	-	-

- program version 3.9.16
- PClient 2.13.40 **or lower**
- Universal Downloader 4.3.06

Please Note: All PClient versions lower than 2.13.40 were not explicitly tested for I.MX35 OS version 1.4.4 or higher. They are very likely to run on devices with this OS but validation needs to be done by the vehicle manufacturer / system integrator.

Miscellaneous

11 Miscellaneous

11.1 Online Update

With this version we are introducing the possibility to receive online updates of the program. These can be bug-fixes, new features or updates of the device software.

Please note: The update is modular, but it always comes in one package, i.e. you cannot update only some of the parts that are updated, either you keep the version you have or you update to the new one available.

Please note: If OS or PClient files are updated, they will overwrite the existing files. If you still need the old versions, please create a backup for yourself. The OS and PClient files can be found in the user folder under the program name, then in extracted components and then os / pclient.

Please note: During deinstallation of the program, the version shown will be the version originally installed, not the version reached through the online updates.

Please note: Files that were downloaded for the online updates will not be deleted during deinstallation, as they were not part of the original installation. If needed, delete these files yourself.

Settings

All relevant settings can be made in the menu Tools -> Plugins.
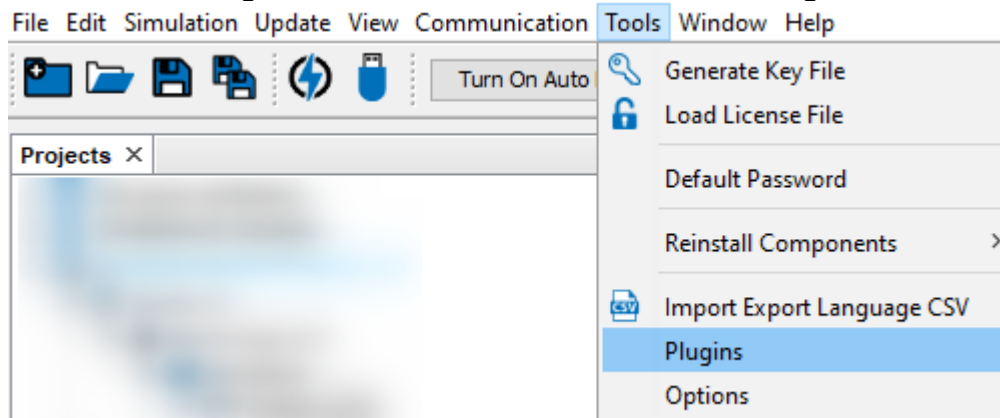
In the Settings tab, the URL for the update check is set. Normally this setting should not be modified by the user.
In the bottom, the interval for the check of available updates can be changed. By default, the program will check for updates at every startup.
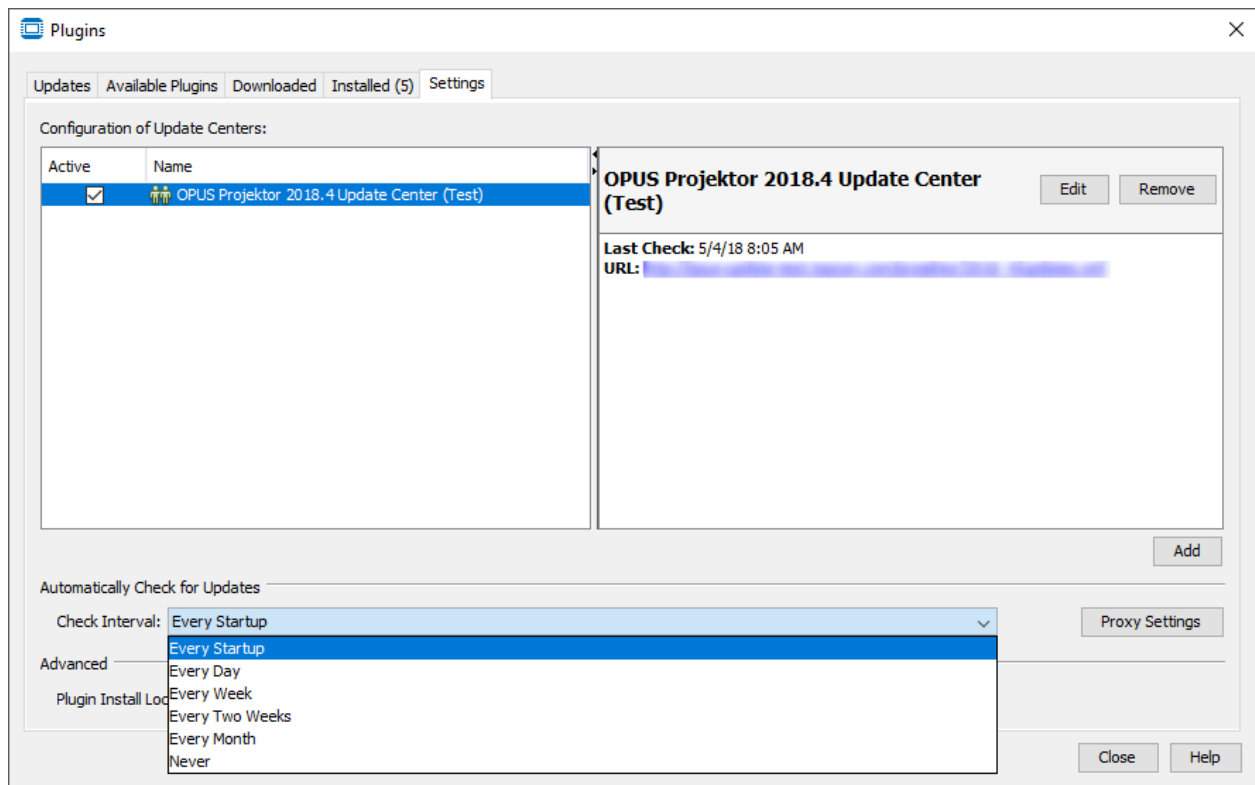
If you don't want to receive notifications about updates, set this to Never.

Please note that you should never deactivate or uninstall modules in the Installed tab. This will break the functionality of the program.

Update Process

When an update is available, you will be notified with this symbol in the lower right corner:
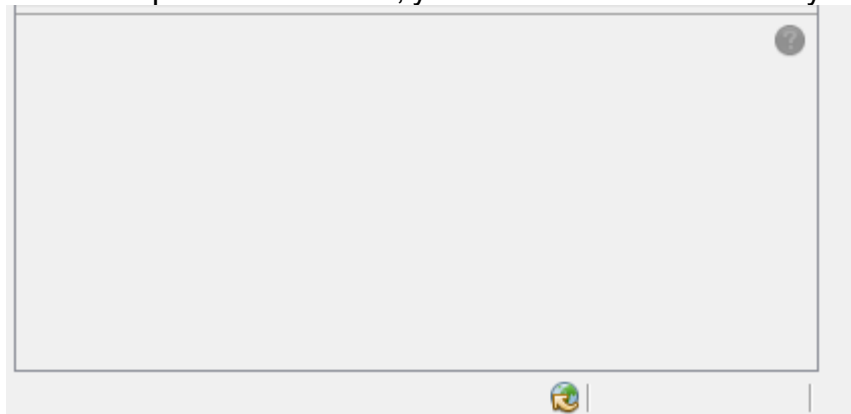
Additionally, an information bubble will be shown when the update is first recognized:
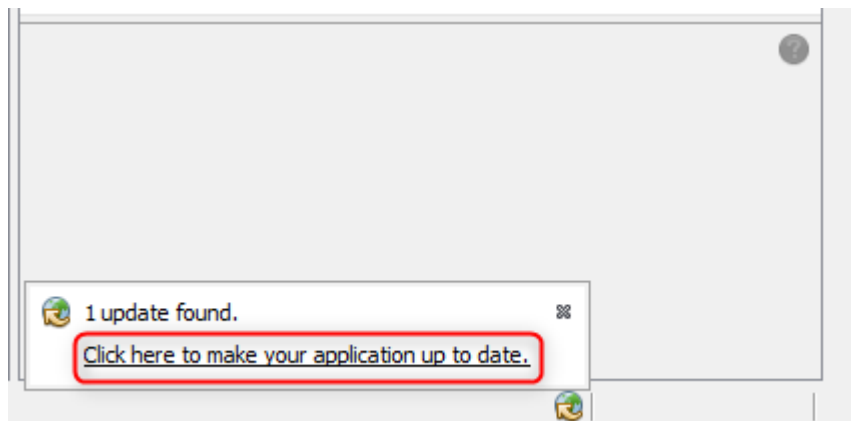
If you close the bubble, you will not be notified again until the set check interval occurs. Click the link in the bubble to see the content of the update.

Alternatively, you can check for updates yourself in the dialog Tools -> Plugins in the tab Updates and click the button Check for Updates. Click the button Update to start the update process.
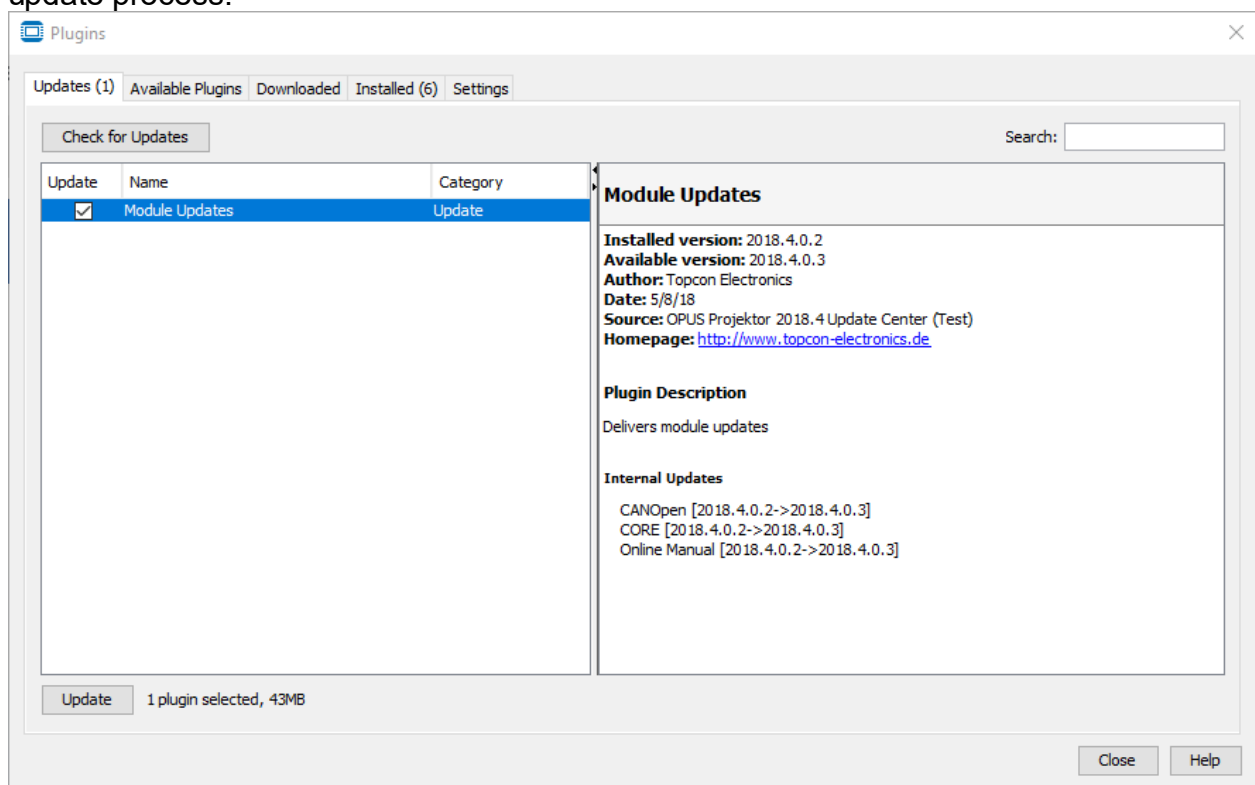
In the next screen you can see the list of modules, i.e. parts of the OPUS Projektor, that are updated. Cancel stops the update process and you will not be notified again about this update until the set check interval occurs. Click Next to continue.
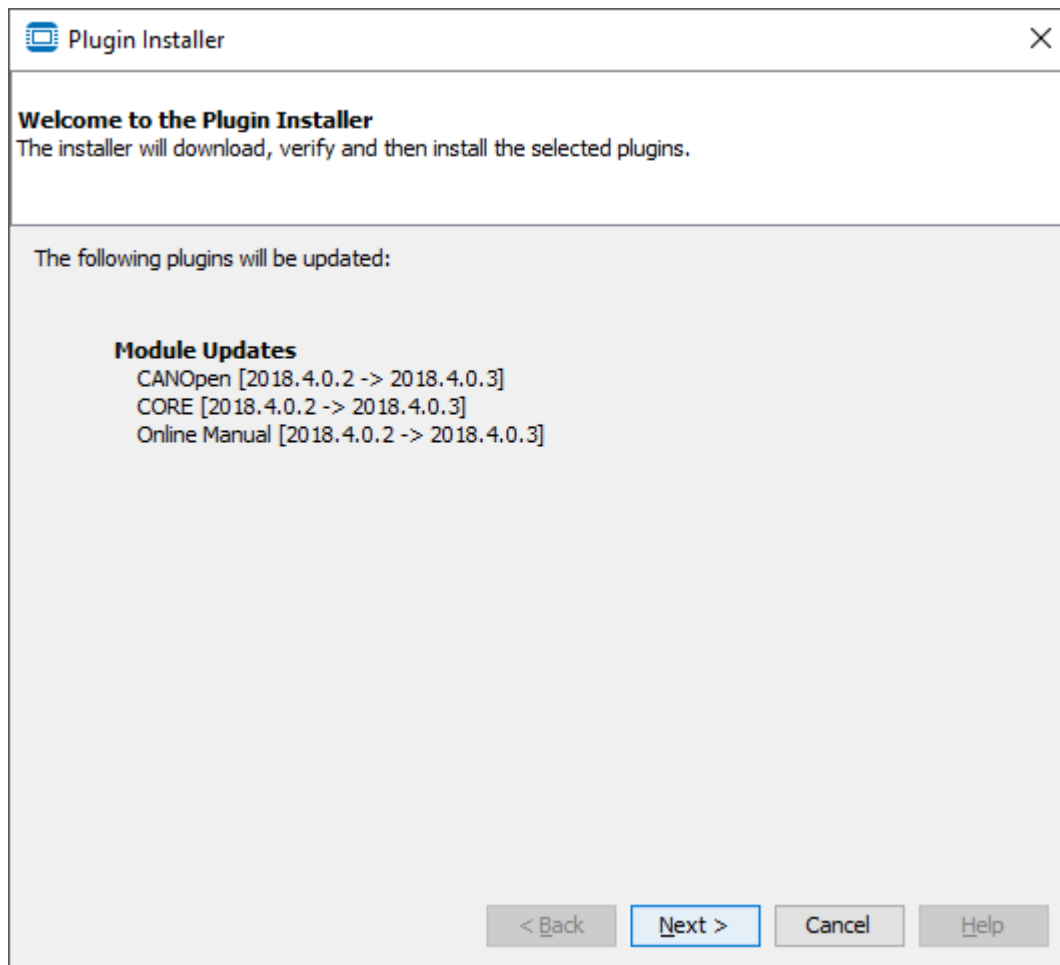
In the next screen you might see a license agreement for each module (depending on whether a new update is already available at the time of the original installation). In most cases these will be identical for all modules. Cancel stops the update process.
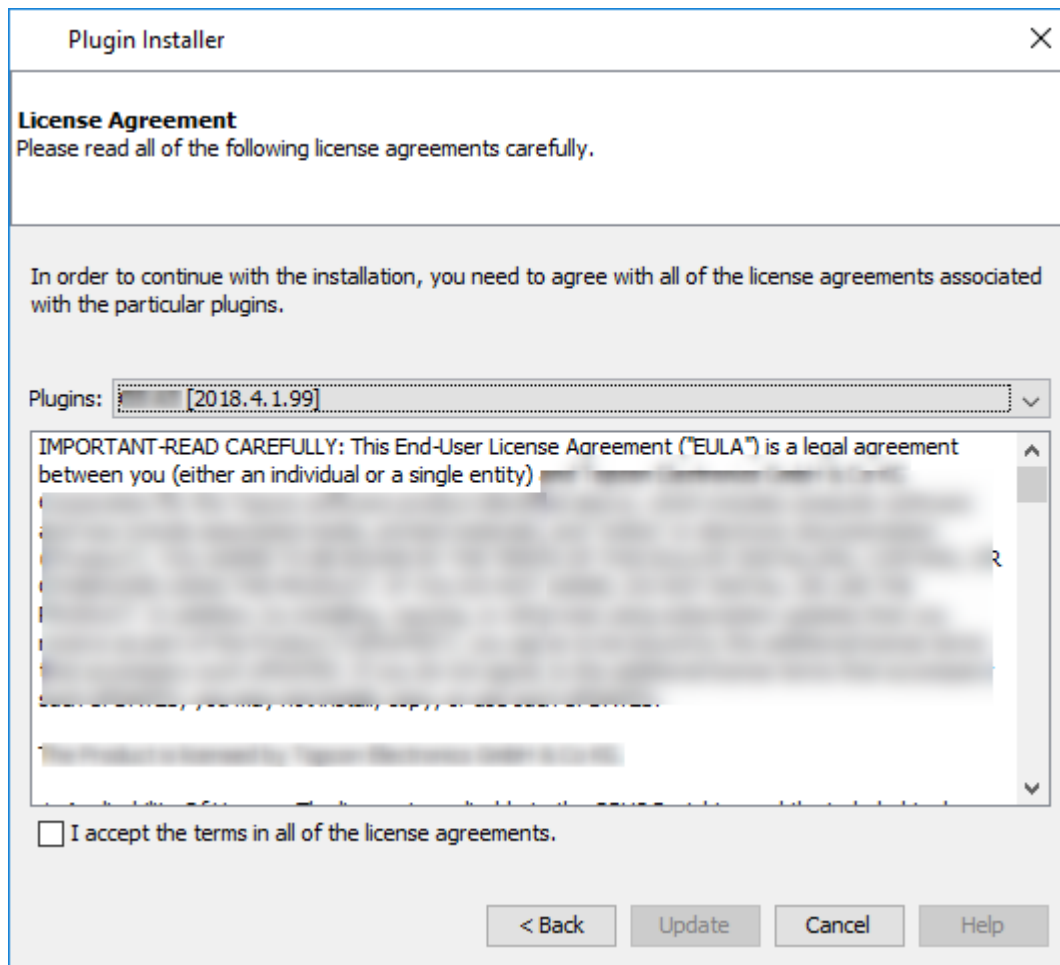
Click the check box to accept the license agreements and then click Update to continue with the update.

If you have started the update from the bubble, it will be downloaded in the background, so you can continue to work. The download progress will be shown in the bottom right. Clicking the X symbol stops the download of the update content.
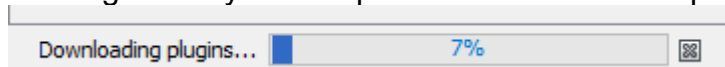
If you have started the update from the Plugins dialog, you will see a popup with information about the download. If you activate the check box Run In Background, the popup will be closed and the download will be shown in the status bar in the bottom right.
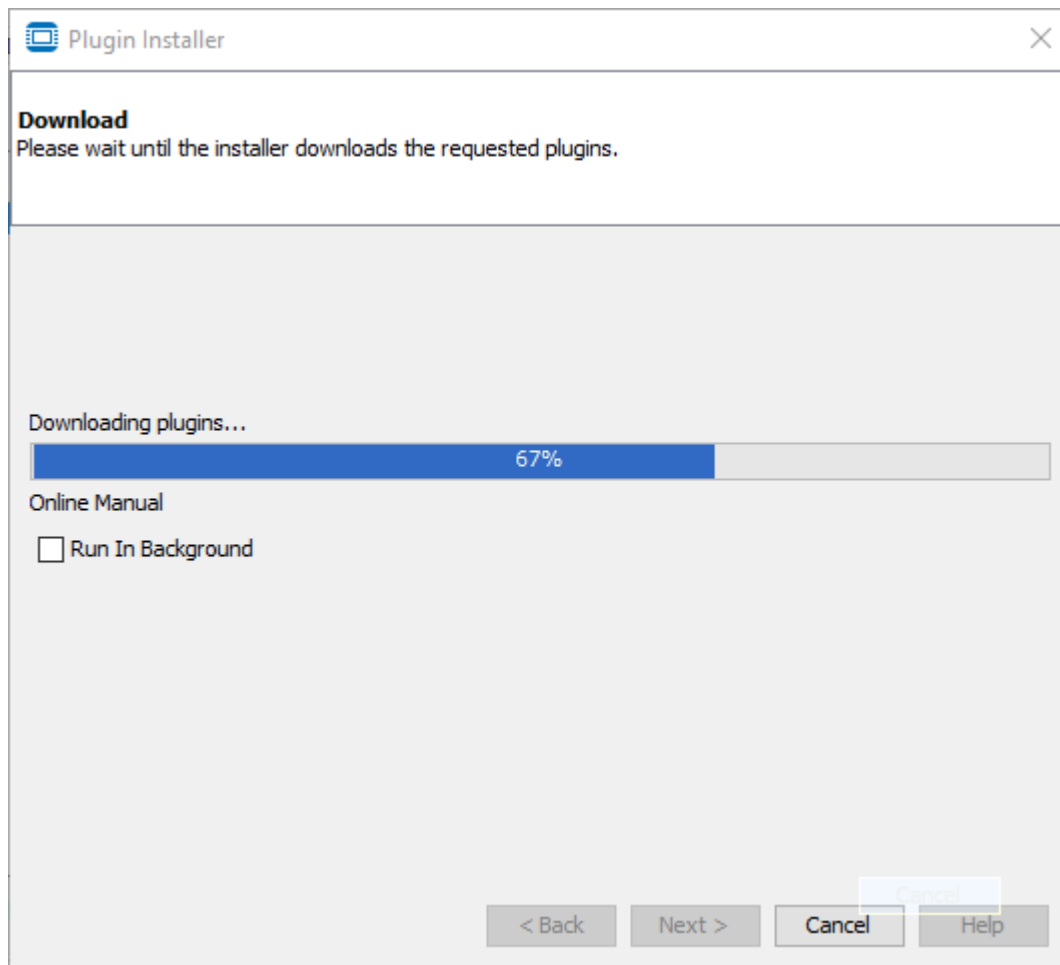
Once downloading has finished, a dialog with the module certificate verification pops up.
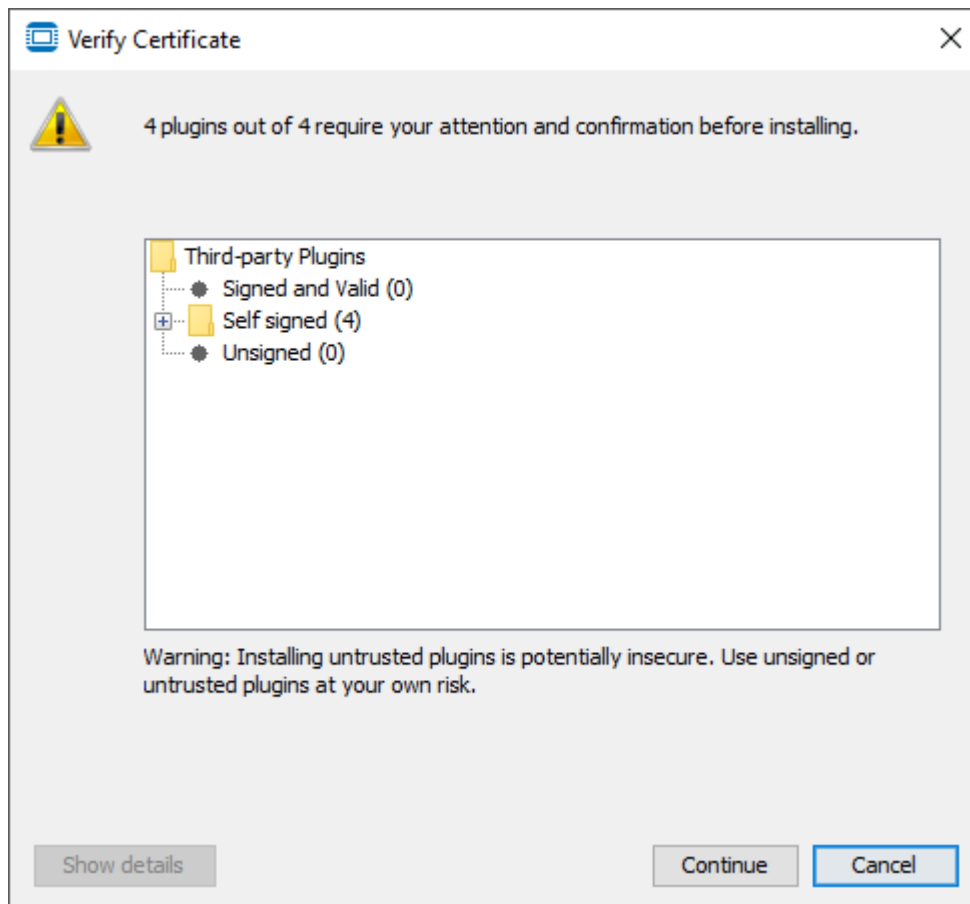
There are 3 possibilities for modules:

- Signed and Valid - This will not be used by us since we can't let our modules be validated externally
- Self Signed - This is where our modules will be. We have signed them ourselves
- Unsigned - Our modules will never be unsigned. If you ever have an unsigned module listed here, contact us with a screenshot and don't install the update

You can check the certificate for each module by selecting a module and clicking Show details.
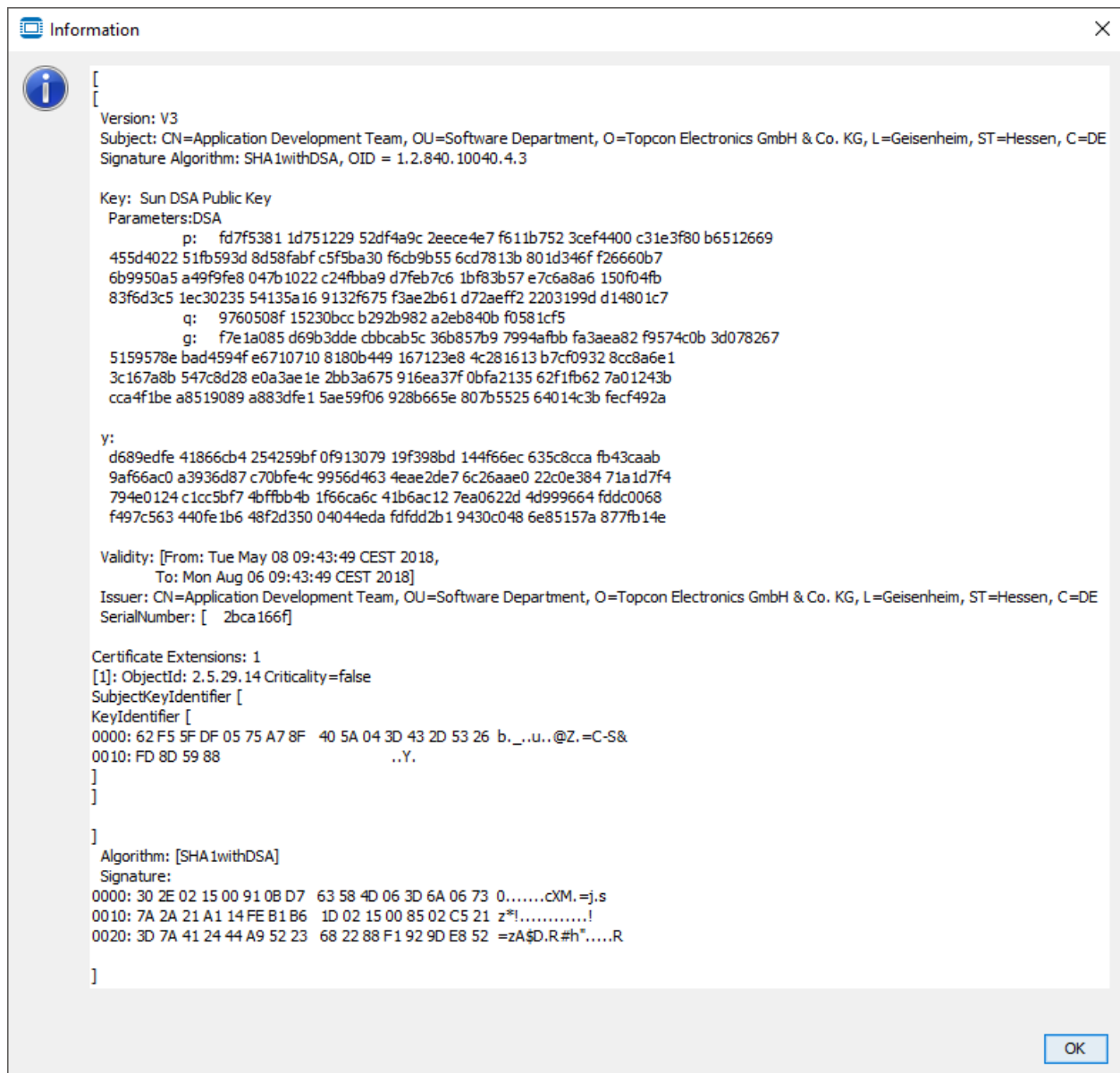
Click OK to close the certificate information.

In the certificate verification dialog, press Continue to progress in the update installation. Cancel stops the update process. Click Next to start downloading the update.

After the verification, a bubble will show up that asks you to restart the program to complete the installation.
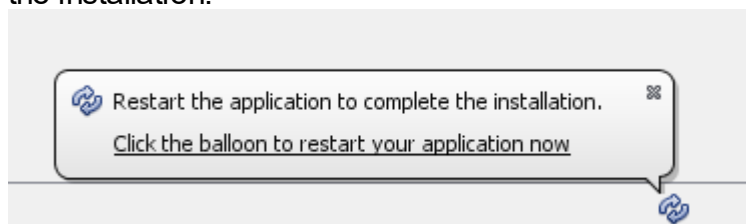
Click the balloon to restart the program and finish the update installation.

Alternatively, if you started the update from the Plugins Dialog, you will see a confirmation dialog that asks you to restart the program. Cancel stops the update process. Select Restart Now and then click Finish to install the update. Select Restart Later to continue working and restart later. The bubble shown above will pop up in the status bar in the bottom. Click the bubble to restart the program. You can also close the Projektor Tool manually and at the next start, the update will be finished.

You will first see a screen with the update progress and information what is installed.
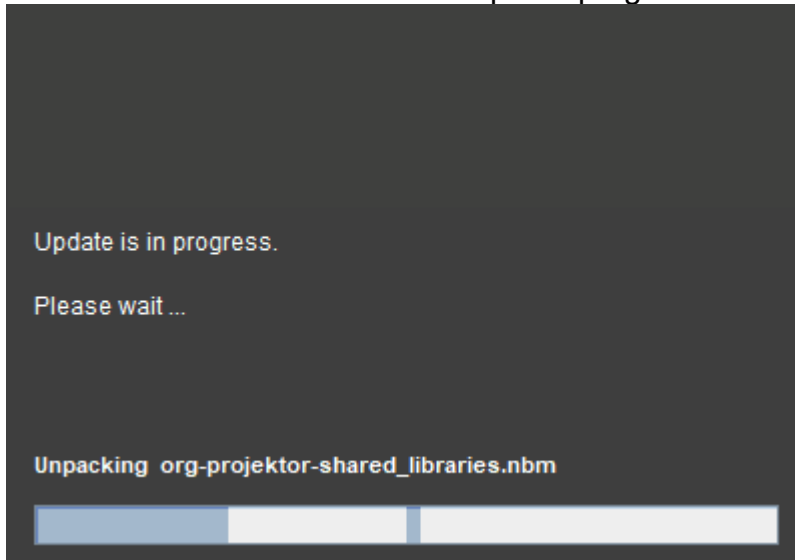
Afterwards the OPUS Projektor will start again and the update is finished.
Please note that it might take a while until the splash screen pops up.

After the program has opened, check the release notes in the Start Page for information about the update.

Information about the modules

The following modules can be part of updates:

Branding - This module contains some visual elements and formatting.
Camera ISO17215[790] - This module contains the Ethernet camera tool.
CANFreestyle[308] - This module contains CANFreestyle CAN protocol functionality.
CANOpen[328] - This module contains CANOpen CAN protocol functionality.
CAN IO - This module contains the DBC import[381] functionality.
Converter[40] - This module contains the project conversion functionality.
CORE - This module contains the core functionalities of the program.
Data Storage - This module contains functionalities for the project storage.
Encoding Support[515] - This module contains functionality for UTF8 support.
Java Runtime Environment - Java runtime to run the program.
J1939[342] - This module contains J1939 CAN protocol functionality.
Module Updates - This module doesn't contain specific functionalities, it is a master module for the update process[660].
Object Themes[580] - This module contains the object theme functionality.

Online Manual[7] - This module contains the offline version of the manual you are reading right now.
OS[653] - These modules contain the OS files for the different devices.
PClient[653] - This module contains the PClient files for the different devices.
Project Samples[130] - This module contains the sample projects.
SharedLibraries - This module is used by external C++ applications (Updater, Simulation GUI).
Symbols ISO7000[135] - This module contains the ISO 7000 symbol library.
Third Party - This module includes third party libraries used by other modules.
Updater[653] - This module contains the updater functionality.
Utilities - This module contains some general functionalities.
Variables[174] - This module contains the predefined variables.
Visual CAN Mapping[397] - This module contains the visual CAN mapping functionality.
Welcome - This module contains the contents of the start page that is shown when the program starts.

11.2   Welcome Projects

When you install a PClient on a device, you will see the following welcome project (the versions shown on the display may differ).

Please note that the welcome project is also available as a sample project[130] .
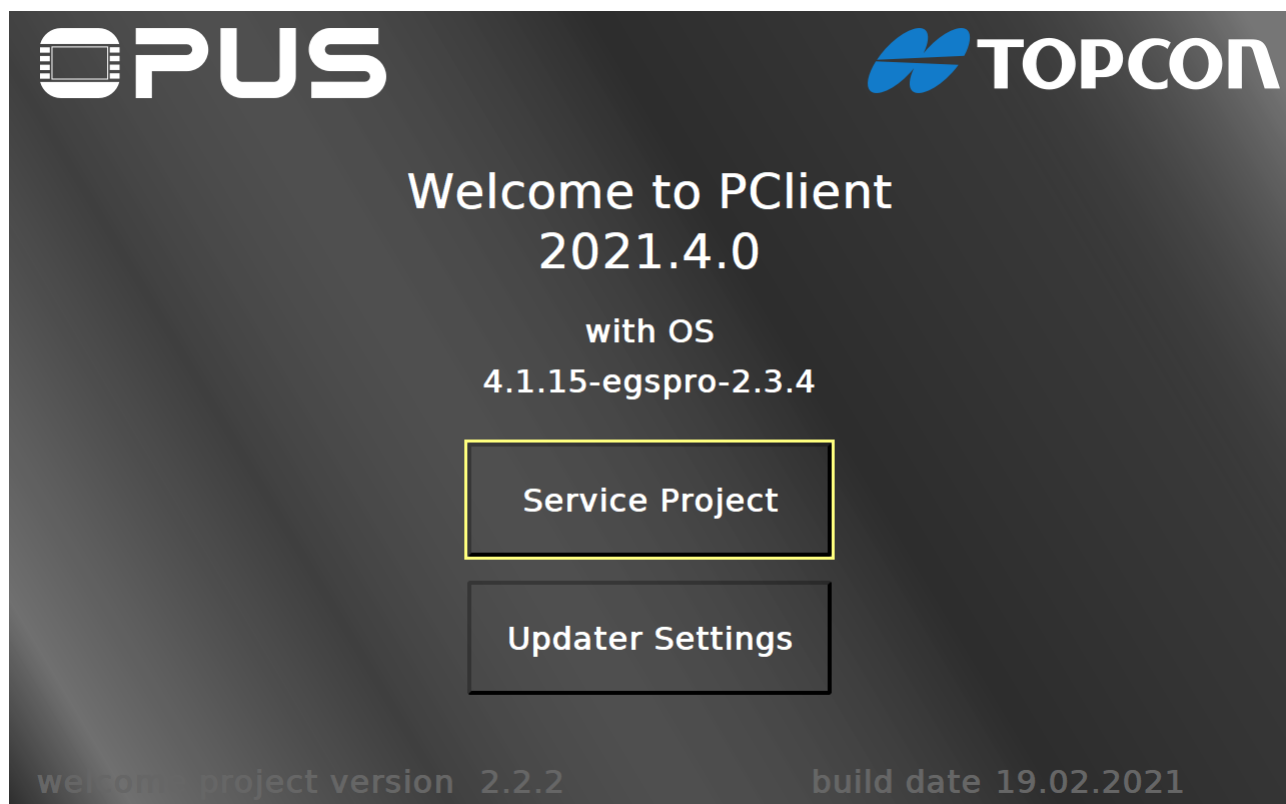
No Touch Mode

If your device has no input method, i.e. no keys and no touch, the welcome project will start in a no touch mode.
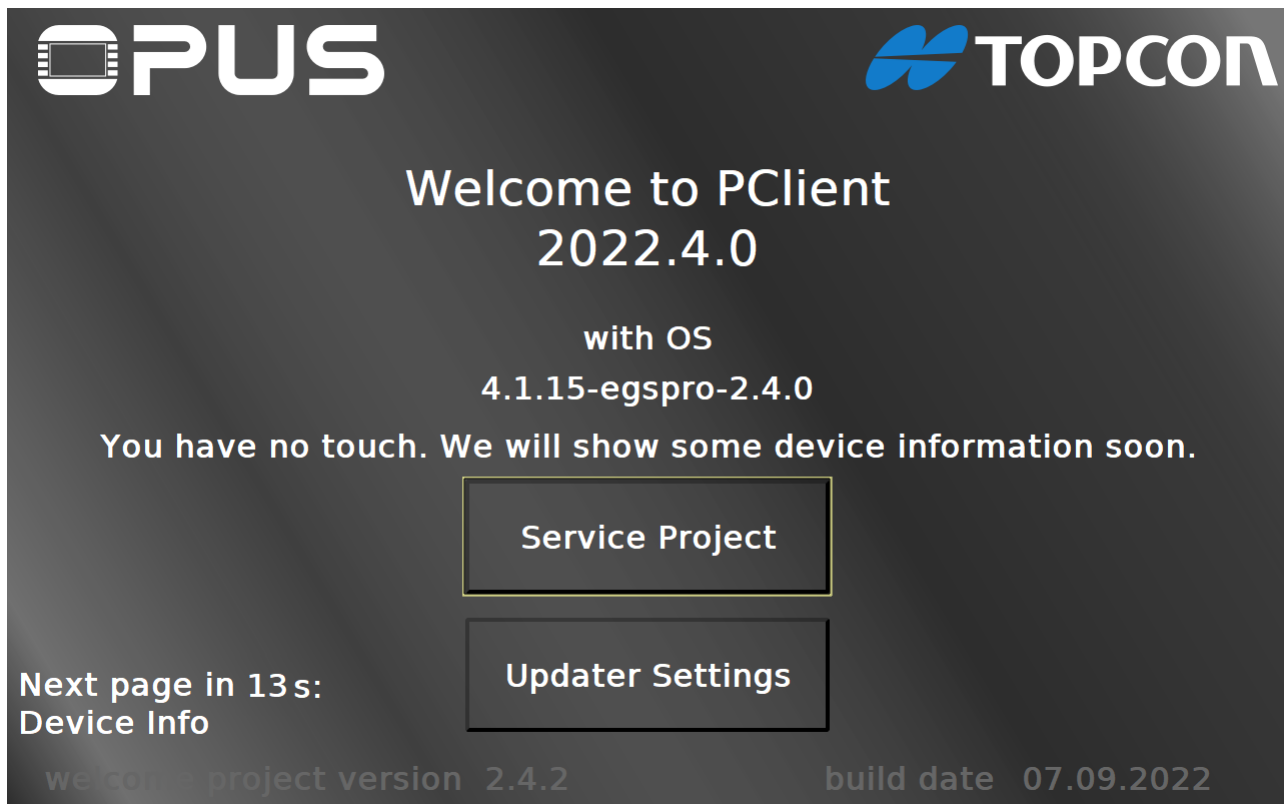
In this mode, the project will show different screens to show some information about the device. The pages are
Home page -> Device Info -> Sensors -> Power Modes -> Power Mode Time Settings -> Power Modes Configuration -> Info to USB
There will be a counter in the lower left, showing when the next page will be shown. The next page will be written there, as well.

On the Info to USB page it is possible to copy the information files to the USB stick by disconnecting the ignition for a short time (just a couple of seconds, make sure you reconnect before the low-power time out time).

Updater Settings

By pressing the Updater Settings button you can navigate to the Updater Settings page.
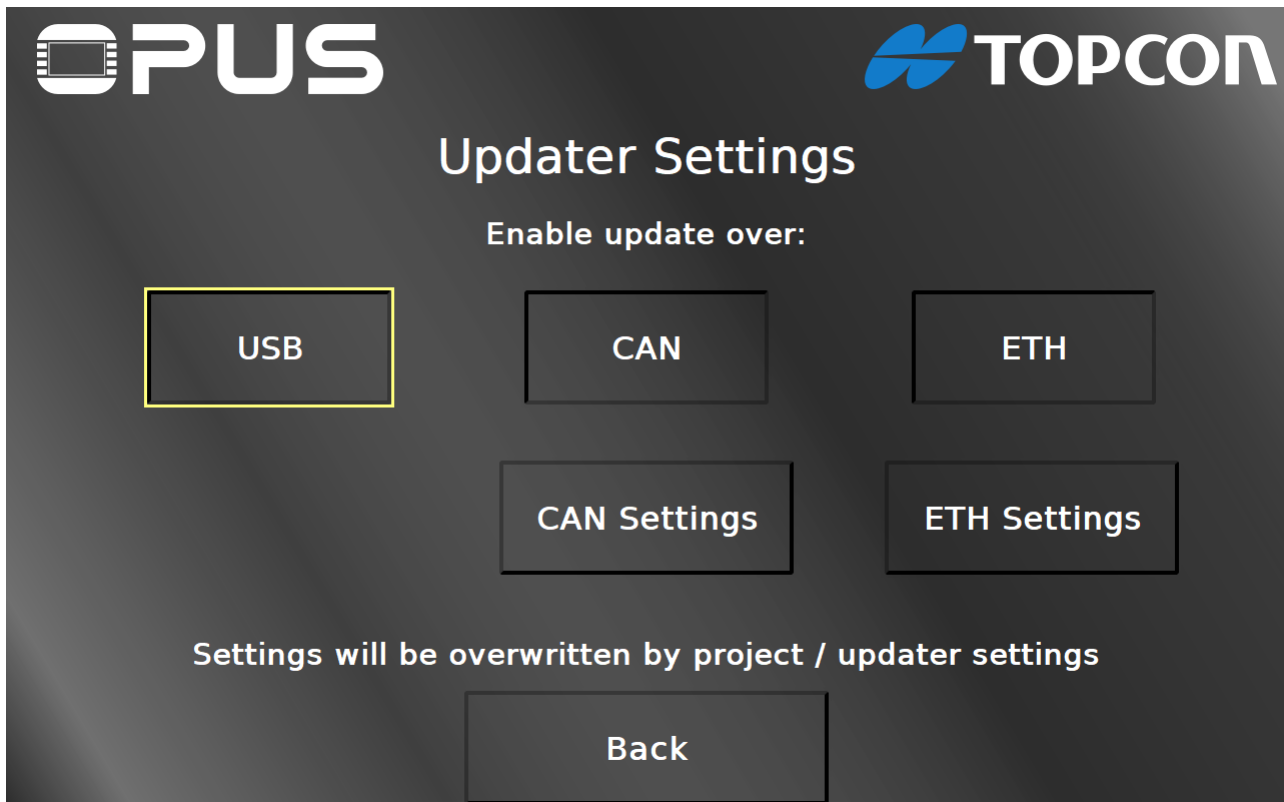
On this page you can enable / disable the project download over USB, CAN and ETH (Ethernet) by pressing the according buttons. Pressing them again will inverse the current setting. By pressing the CAN Settings button you can navigate to the CAN Settings page. By pressing the ETH Settings button you can navigate to the Ethernet Settings page. Both settings buttons will only be visible when the according transfer medium is enabled. Back returns to the main welcome page.
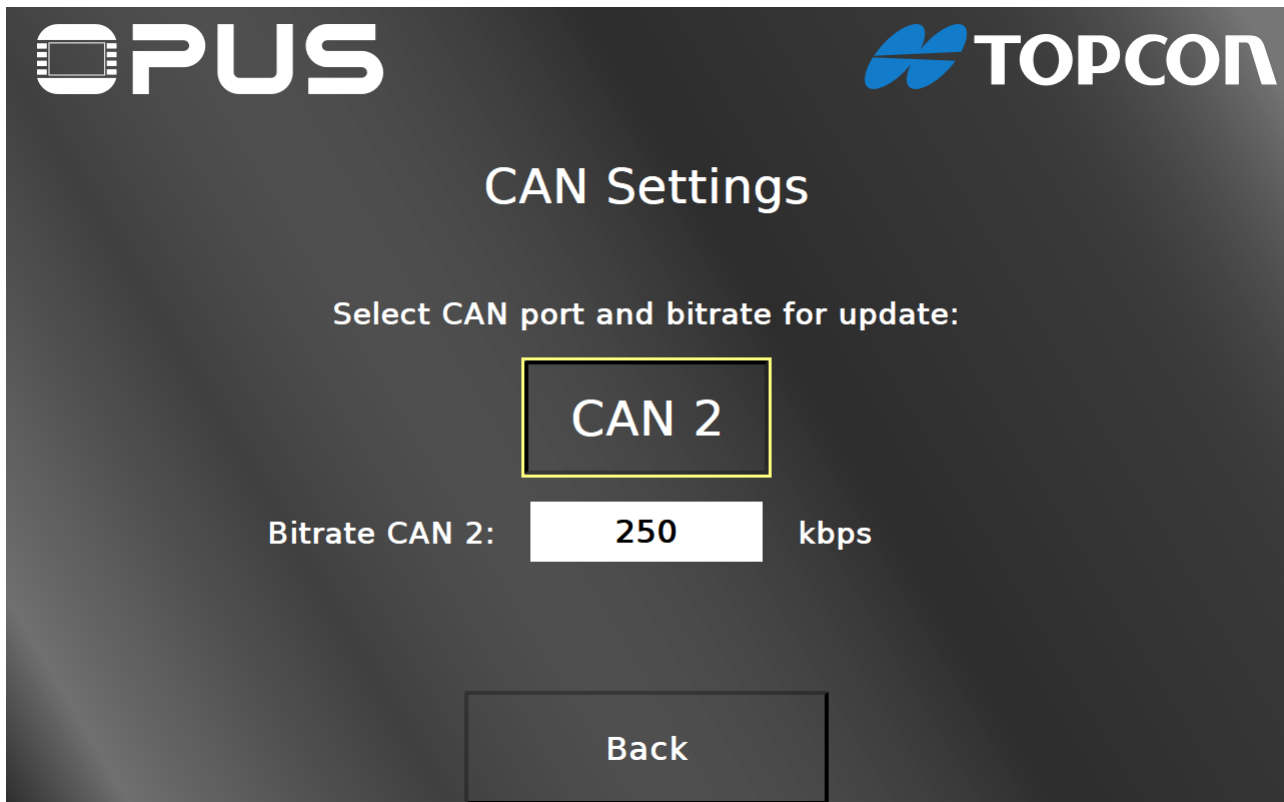
On the CAN Settings page you can set the CAN port that should be used for the project download and the Bitrate for the project download for that port.
Back returns to the Updater Settings page.
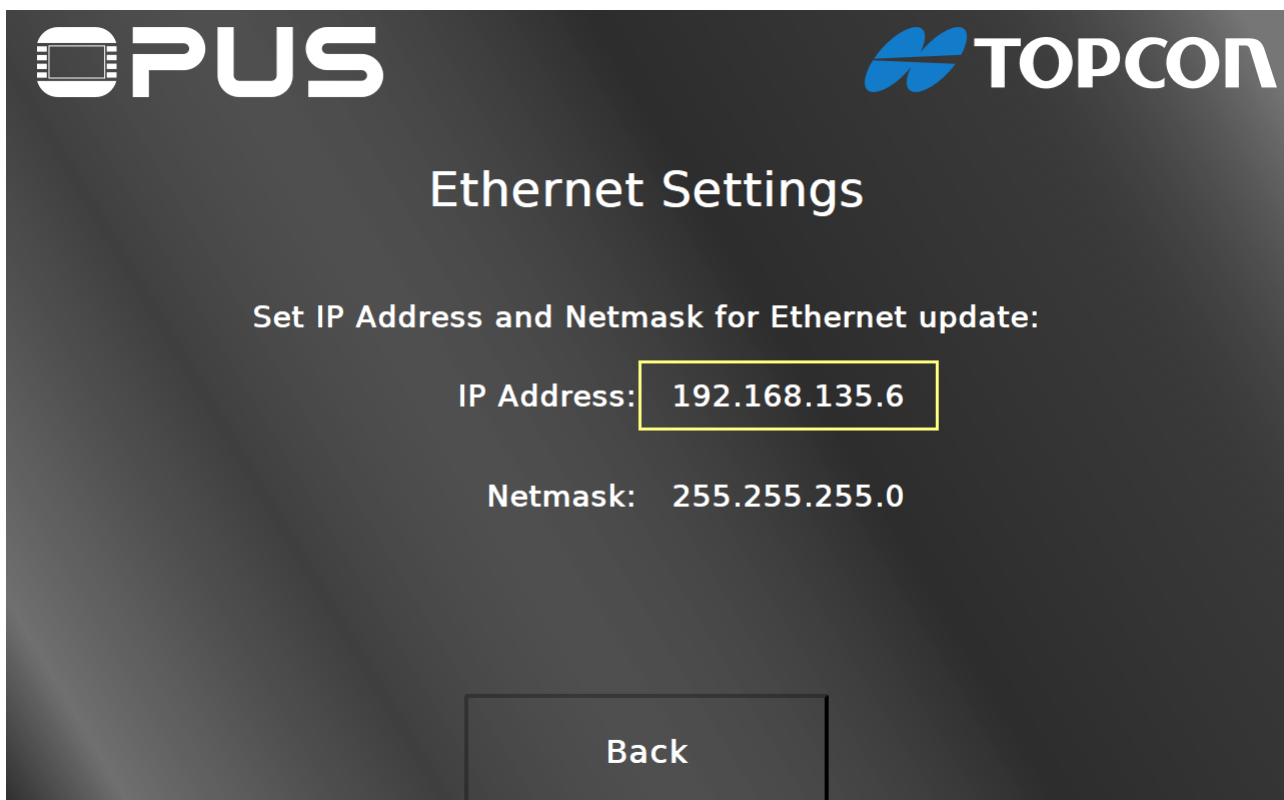
On the Ethernet Settings page you can set the IP Address (10) and Netmask (11) that the device should have in the network. With the **Back** button (12) you can go back to the Updater Settings page.

Please note: All settings made in this project are temporary and only valid for the first project transfer. All settings will be overwritten by settings in the transferred project.
You should make the according settings in the project you want to transfer to the device:

- The variable @EnableUpdater variable controls which update is allowed
- The CAN bus bitrate will be taken from the bitrate set in the transferred project
- The CAN port used for the transfer can be set with the variable @CANx_Bitrate
- The IP address can be set with the variable @Ethernet0_IPAddress
- The Netmask can be set with the variable @Ethernet0_Netmask

Back returns to the Updater Settings page.

Service Project

By pressing the **Service Project** button you can navigate to the Service Project main page.
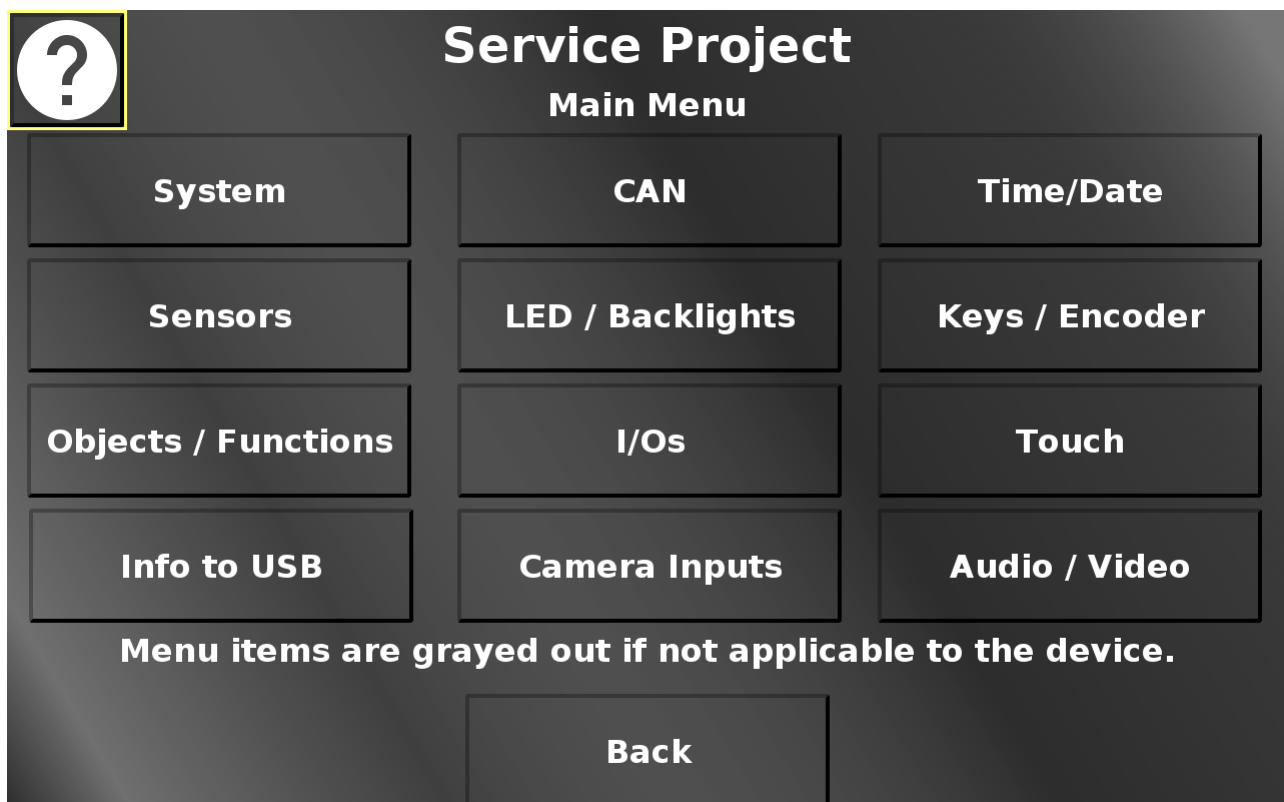
In this screen the different main menu items can be chosen. Back returns to the main screen.

On every page, a help button is available, which gives a short explanation what can be done on the page.

Menu items, behind which there are only features that are not available for the current device will be grayed out. It is still possible to go into the according pages or use the according buttons, but nothing will happen or work.

On most pages in the Service Project an "Eject USB" button will be displayed next to the help button if a USB stick is connected to the device. If you want to unplug the USB stick, press this button first to unmount it properly from the device.

Service Project -> System
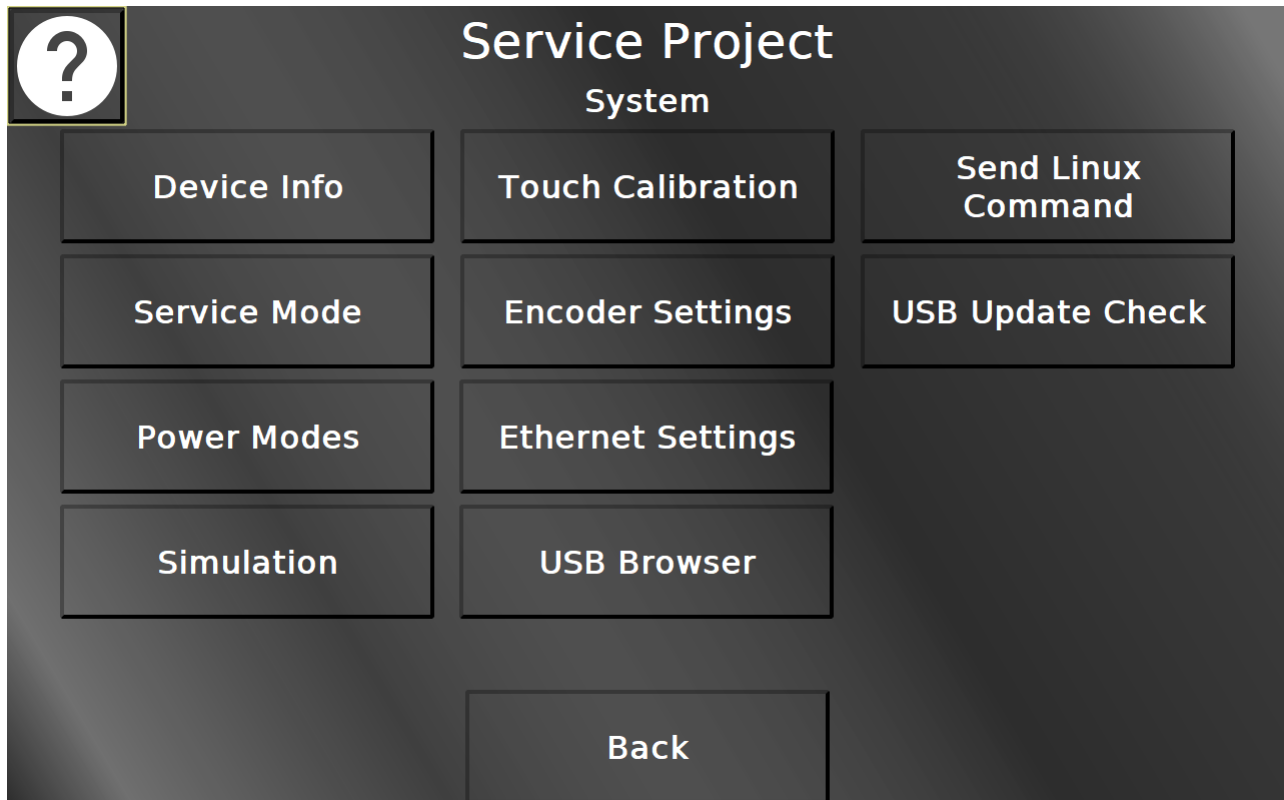
In this screen different system information and system settings can be chosen. Back returns to the main menu.

Service Project -> System -> Device Info
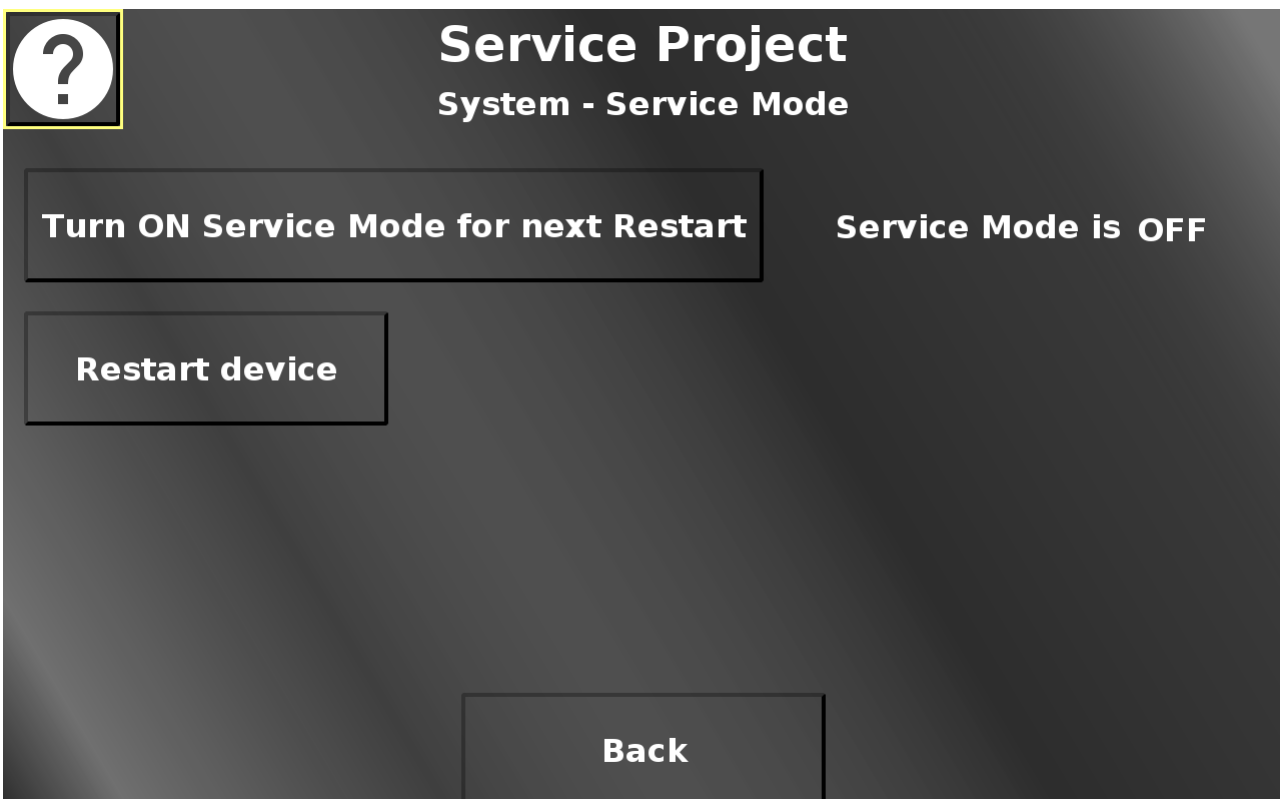
Service Project
System - Device Info

Article number: **OPUSA8SN1CANF000**

Serial number: **1845001AD**

Device type: **12.1" Standard Full**

OS version: **4.1.15-egspro-2.3.2-1**

PClient version: **2020.4.0.3**

Operating time: **3 d, 14 h, 22 m**

Current Uptime: **0 h, 25 m, 33 s**

Mass storage: **2806.800 MB / 3010.360 MB free**

Back

In this screen different device data is displayed. Back returns to the system menu.

Service Project -> System -> Service Mode

Service Project
System - Service Mode

Turn ON Service Mode for next Restart   Service Mode is OFF

Restart device

Back

In this screen the service mode can be activated for the next reboot to update OS / PClient. Pressing the restart button opens a confirmation dialog.
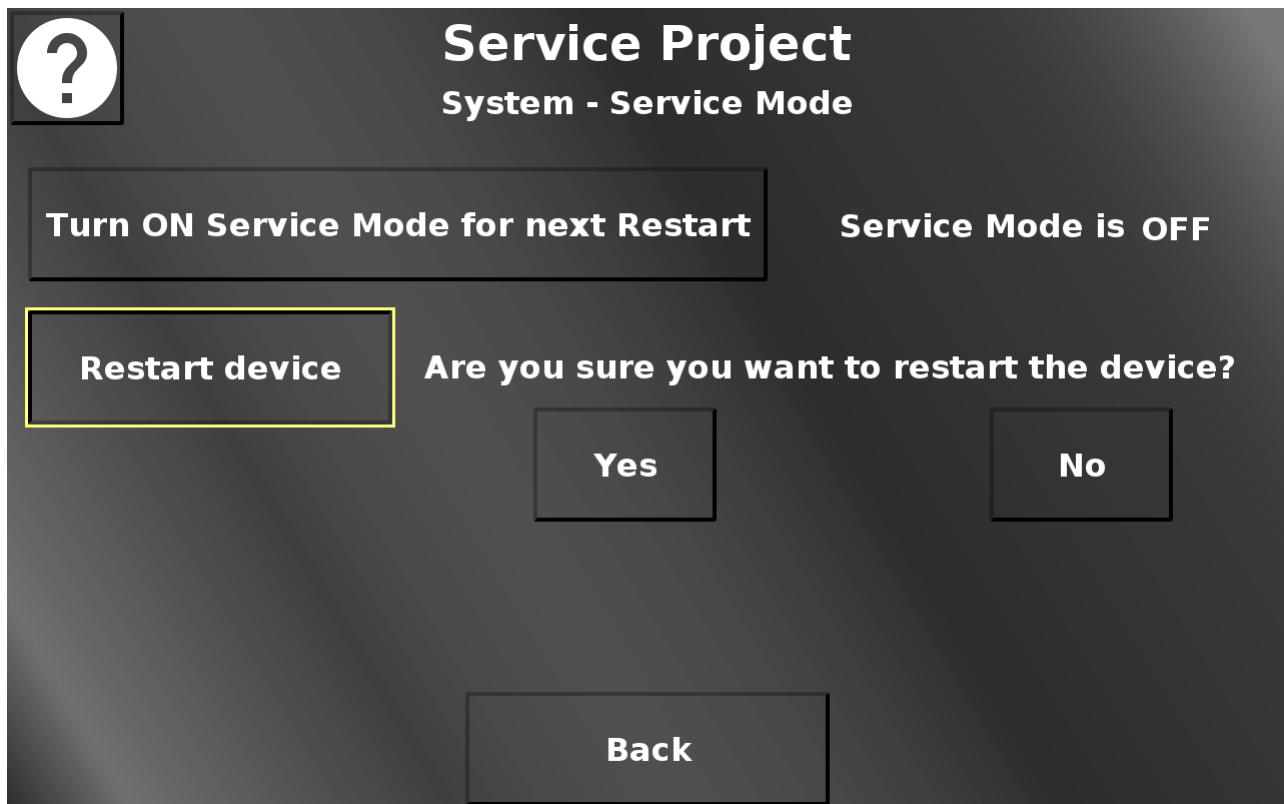
By pressing Yes the device will reboot. By pressing No the dialog will be closed again. The service mode setting will be kept. Back returns to the system menu.

Service Project -> System -> Power Modes
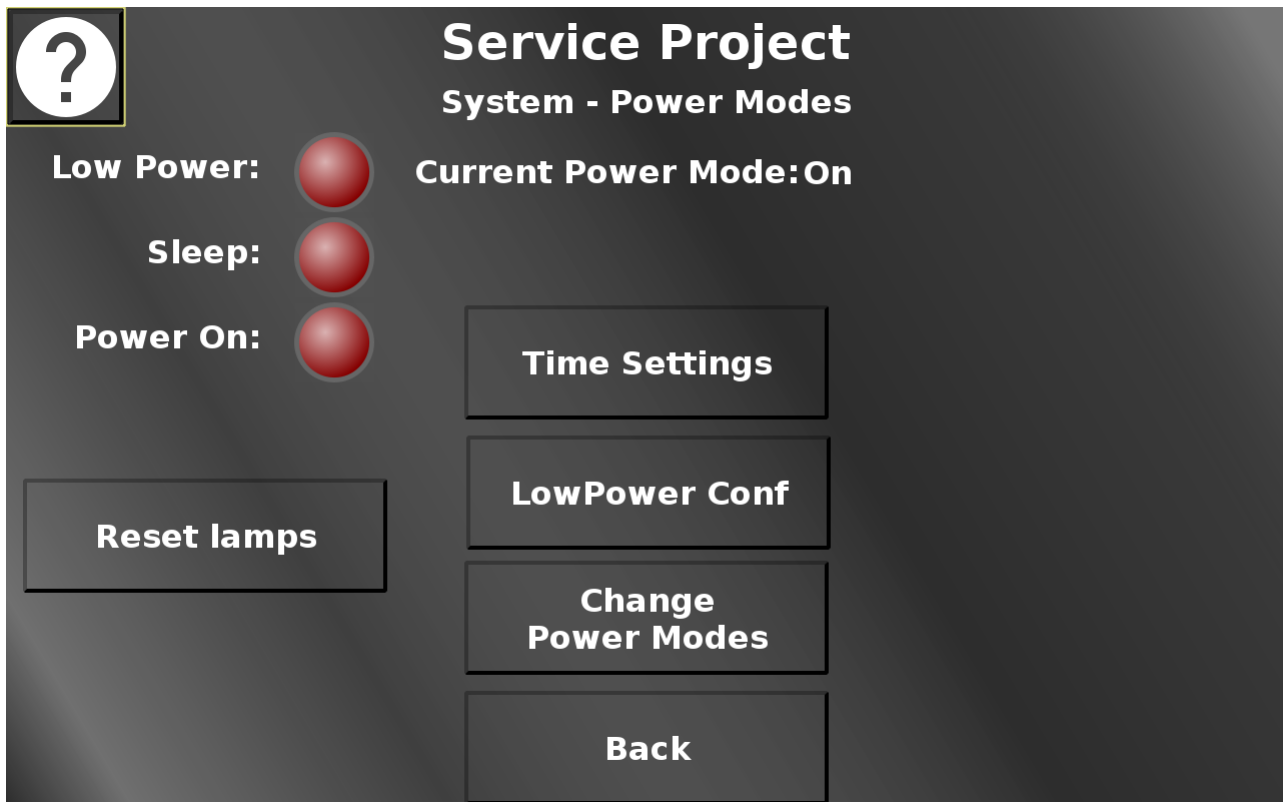
In this screen the different <u>power events</u>[746] can be tested. The current power mode is shown in the top. The lamps will light up when the according power event is used. Disconnect clamp 15 /ignition pin to enter low-power mode, and then sleep mode, after which the device will turn off. Reset lamps turns all lamps off.

Additionally, three configuration screens can be entered. Time settings allows setting the times for the different power modes. LowPower Conf leads to a page where hardware elements can be configured for the low-power mode. On Change Power Modes some power modes can be entered.

Back returns to the system menu.

Service Project -> System -> Power Mode Time Settings

Service Project

System - Power Modes - Time Settings

Modifier: 1s

| Low Power Time | 0 d, 0 h, 0 m, 5 s | - | + |
| Sleep Time | 0 d, 0 h, 2 m, 0 s | - | + |

Back
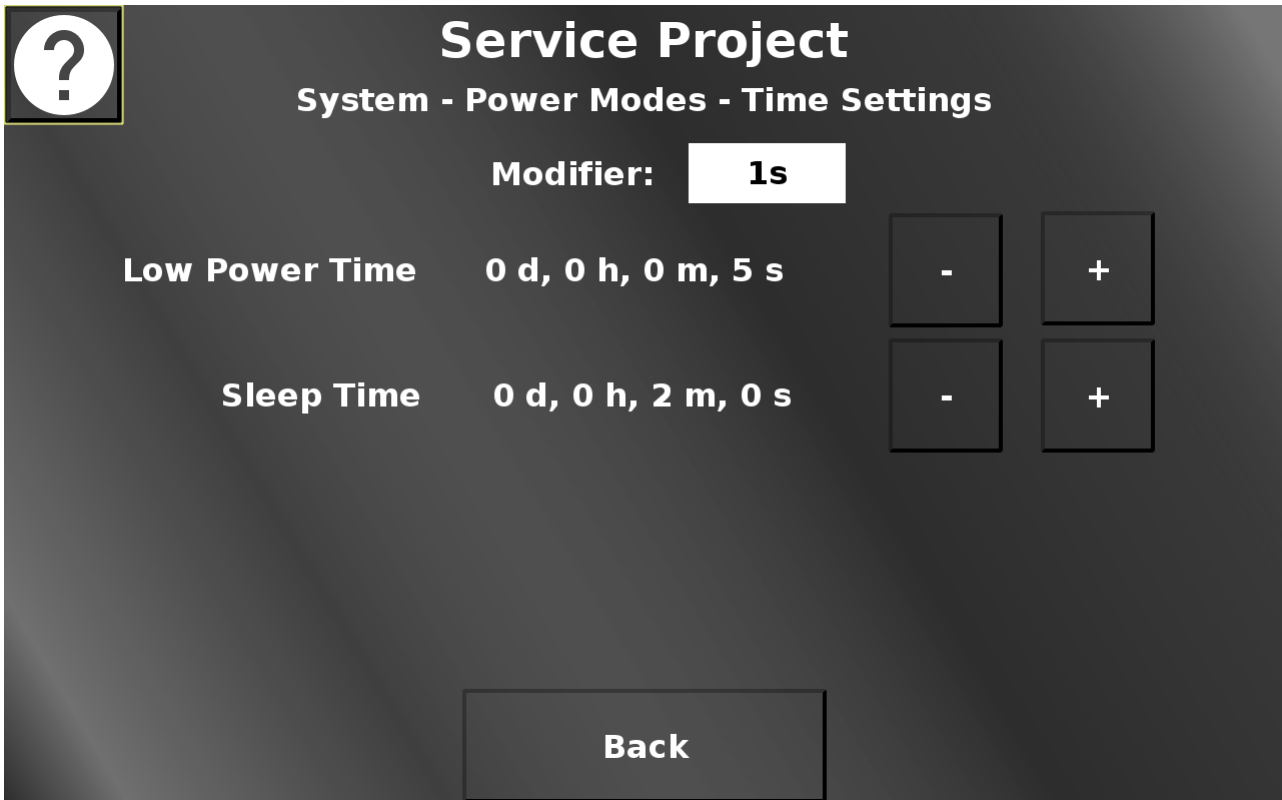
In this screen the times for the low-power mode and the sleep mode can be set. - (minus) decreases the time, + (plus) increases the time. For some devices, the silent on to low power time can be set, as well. The amount that the current value is modified by, can be chosen with the Modifer dropdown. Back returns to the Power Modes screen.

Service Project -> System -> Low Power Configuration
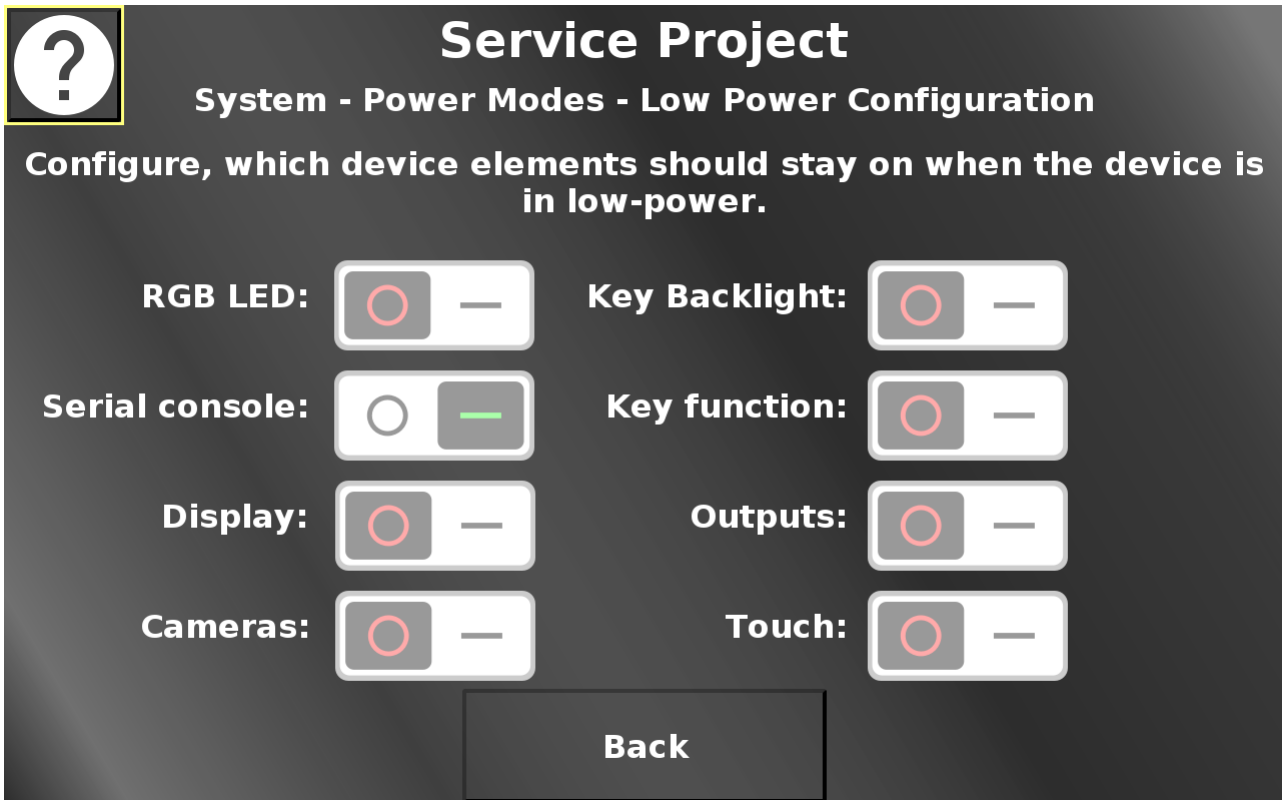
In this screen, hardware elements can be configured to stay on or turn off when low power mode is entered. If the switch is in the OFF position, the element will turn off when low-power mode is entered. If the switch is in the ON position, the element will stay on in low-power mode.
Back returns to the power mode screen.

Service Project -> System -> Change Power Modes

Service Project
System - Power Modes - Change Power Modes

Low Power: Current Power Mode: On

Sleep:

Power On:

[Set Low Power Mode]   [Set On Mode]

[Reset lamps]

[Reboot]

[Shut down]
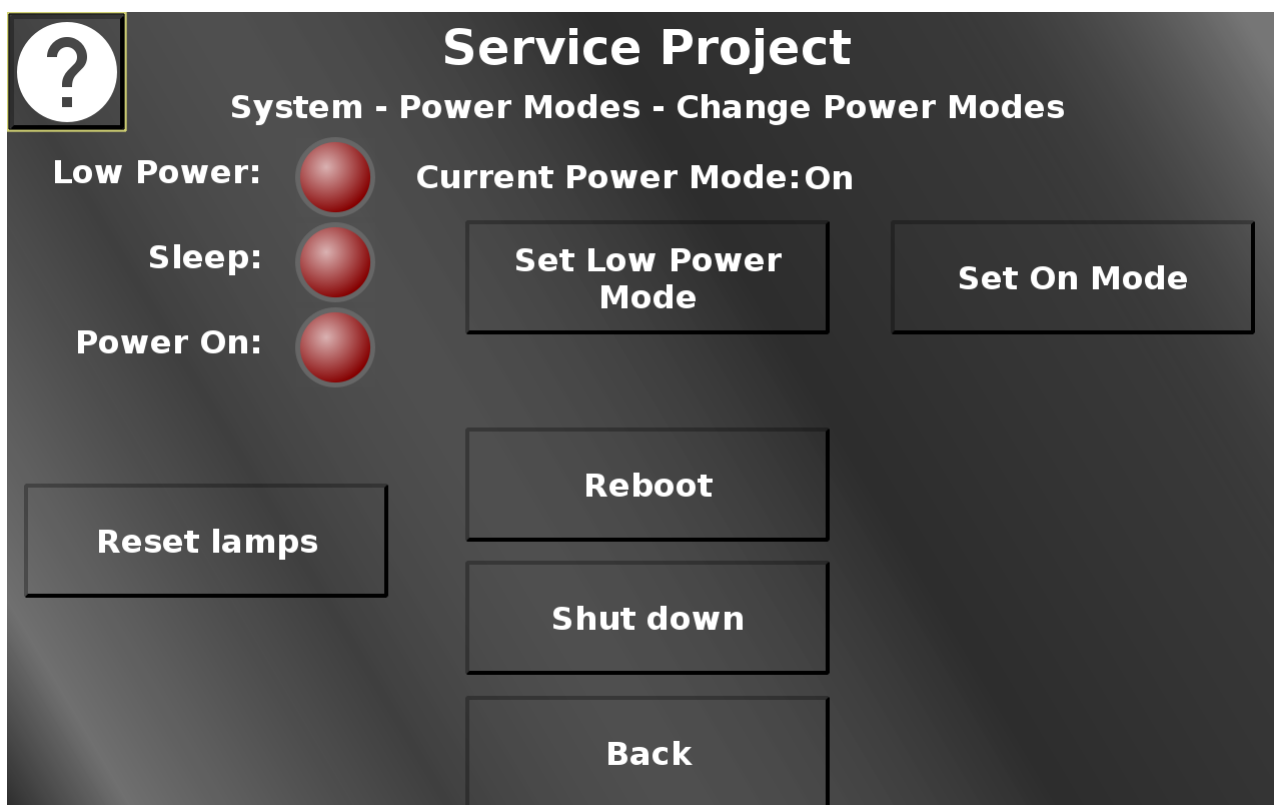
[Back]

In this screen different power modes can be activated. Low power and on mode can be set here. For some devices, a timed low power mode can be set, as well. For both low power modes a popup will be shown, warning that a way back to on mode should be considered, either by disconnecting and re-connecting ignition, or by setting display, touch and or keys to still work in low power mode. Additionally, the device can be rebooted and shut down. The lamps and the current mode are identical to the power mode screen.
Back returns to the power mode screen.

Service Project -> System -> Simulation
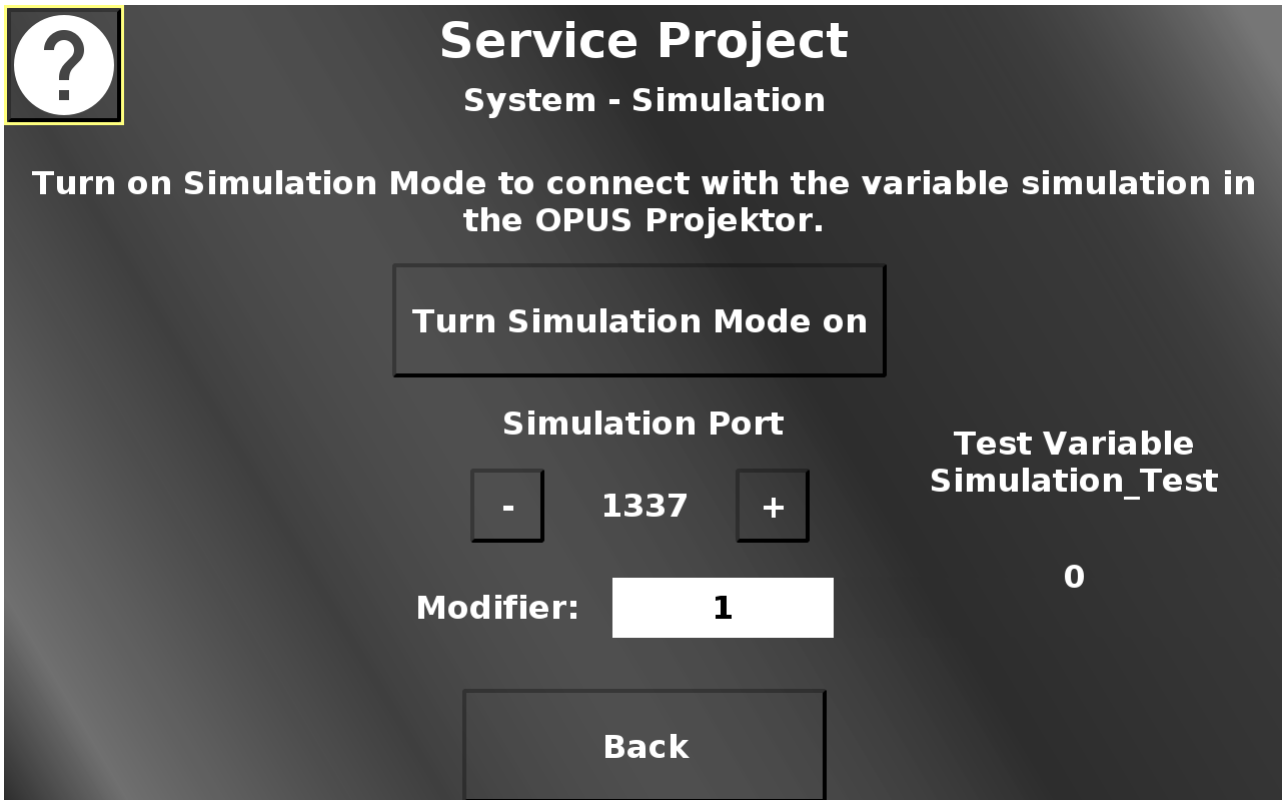
In this screen the simulation mode can be enabled to be able to use the variable simulation [601] in the OPUS Projektor for the device.
Additionally, the port for the simulation can be set by pressing the - (minus) and + (plus) buttons.
The modifier drop down chooses the value by which the - and + buttons change the port value.
Back returns to the system menu.

Service Project -> System -> Touch Calibration

Service Project
System - Touch Calibration

Note: The calibration is only available for G1 devices (A3 / A6 G1) with touchscreen.

Start calibration

Back

In this screen the touch screen calibration can be started. 5 cross hairs will appear one after another (top left, top right, bottom right, bottom left, center). Press the cross hairs to calibrate the screen.
Please note that the PClient stays on during this operation, so the cross hairs will be painted over this screen.
Please note that the calibration is only available for A3/A6 G1 touch screens. Devices with capacitive touch screens do not neet this calibration.
Back returns to the system menu.

Service Project -> System -> Encoder Settings

Service Project
System - Encoder Settings

With Roll Over, the encoder focus will roll over from the last object on the frame back to the first.

Dectivate Roll Over

This changes the encoder navigation direction when scrolling through objects.

Clockwise

Back
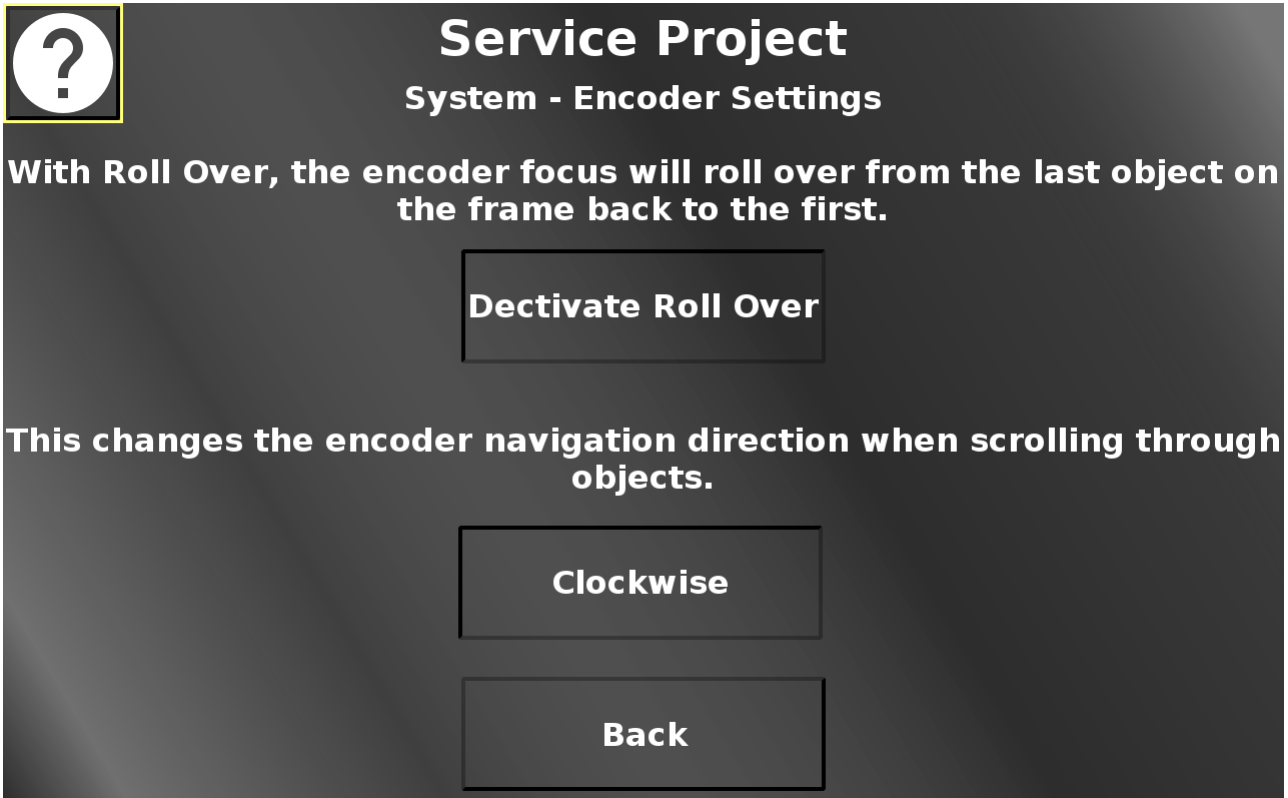
In this screen two encoder settings can be made.
Roll over means that the yellow selection rectangle will not stop at the last object, but it will roll over to the first object again.
Additionally, the navigation direction for the encoder can be switched between clockwise and anti-clockwise.
Back returns to the system menu.

Service Project -> System -> Ethernet Settings

Service Project
System - Ethernet Settings

Tap on the address values to change them.

IP Address: 192.168.135.6

Netmask: 255.255.255.0

Enable SSH/Dropbear Service: **Enable SSH**

Back

In this screen, the IP address and Netmask can be set, similarly to the updater screen described above.
Additionally, SSH can be enabled here. On devices with Automotive Ethernet, the mode can be switched between master and slave here, as well.
Back returns to the system menu.

Service Project -> System -> USB Browser
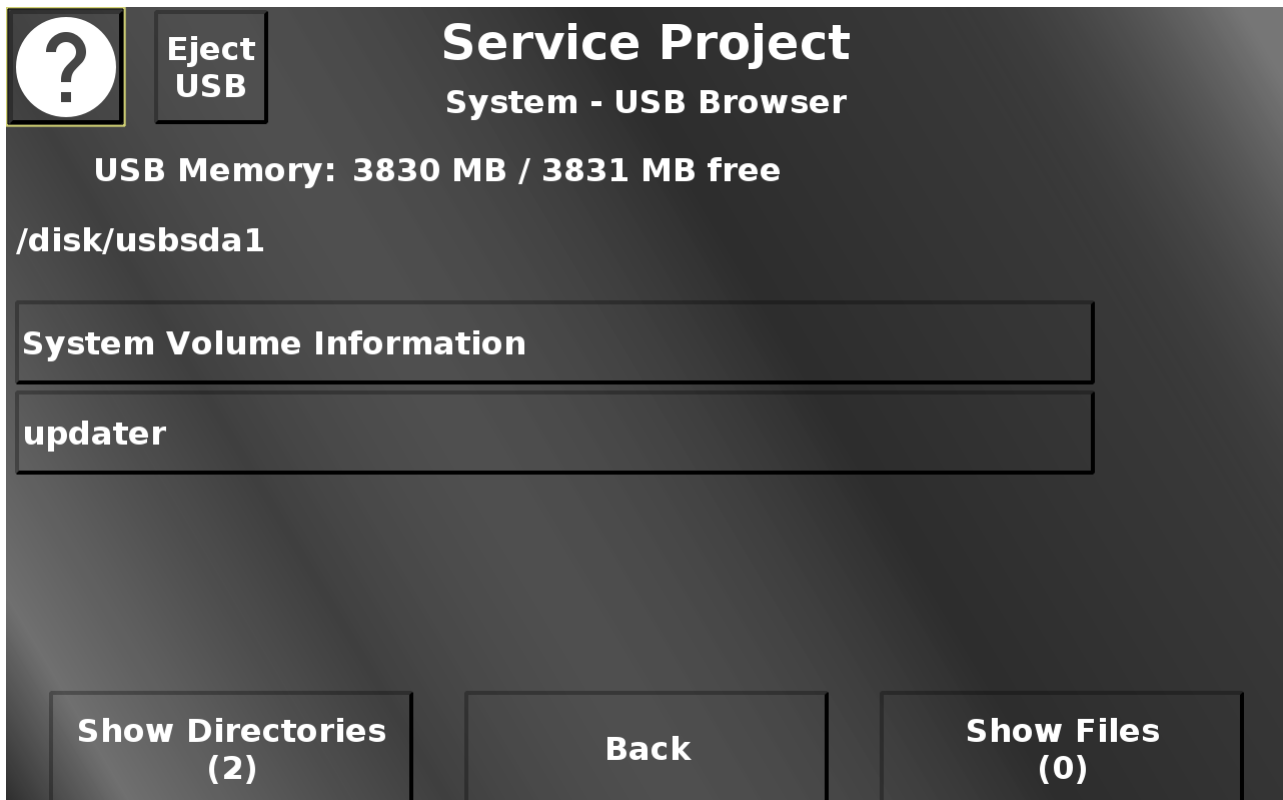
In this screen a rudimentary USB file browser is implemented. If a USB stick with a valid file system is plugged in, the content will be shown. Next to the help button, an eject/unmount button will be shown to unmount the USB stick before unplugging it. The view in the browser can be switched between directories and files.

For directories, a press of the according button will show the content of this directory. The current path is shown in the top of the screen. When directories are enterered, a "back" button will be displayed to go back up to the root of the USB stick. Both for files and directories, "up" and "down" buttons will be shown if more than 4 directories / files are present in the current directory.

For files, a button press will open an alarm with a preview of the file. For image files the image will be shown. Note that the aspect ratio will not be correct. For compatible audio and video files an alarm with a multimedia player will be shown to play the file. Note that the aspect ratio of the video will not be correct. For pdf files the pdf reader will be opened. For all other file types the beginning of the file (up to 2MB) will be shown as text. The text can be scrolled to see more content. Back returns to the system menu.

Service Project -> System -> Send Linux Command
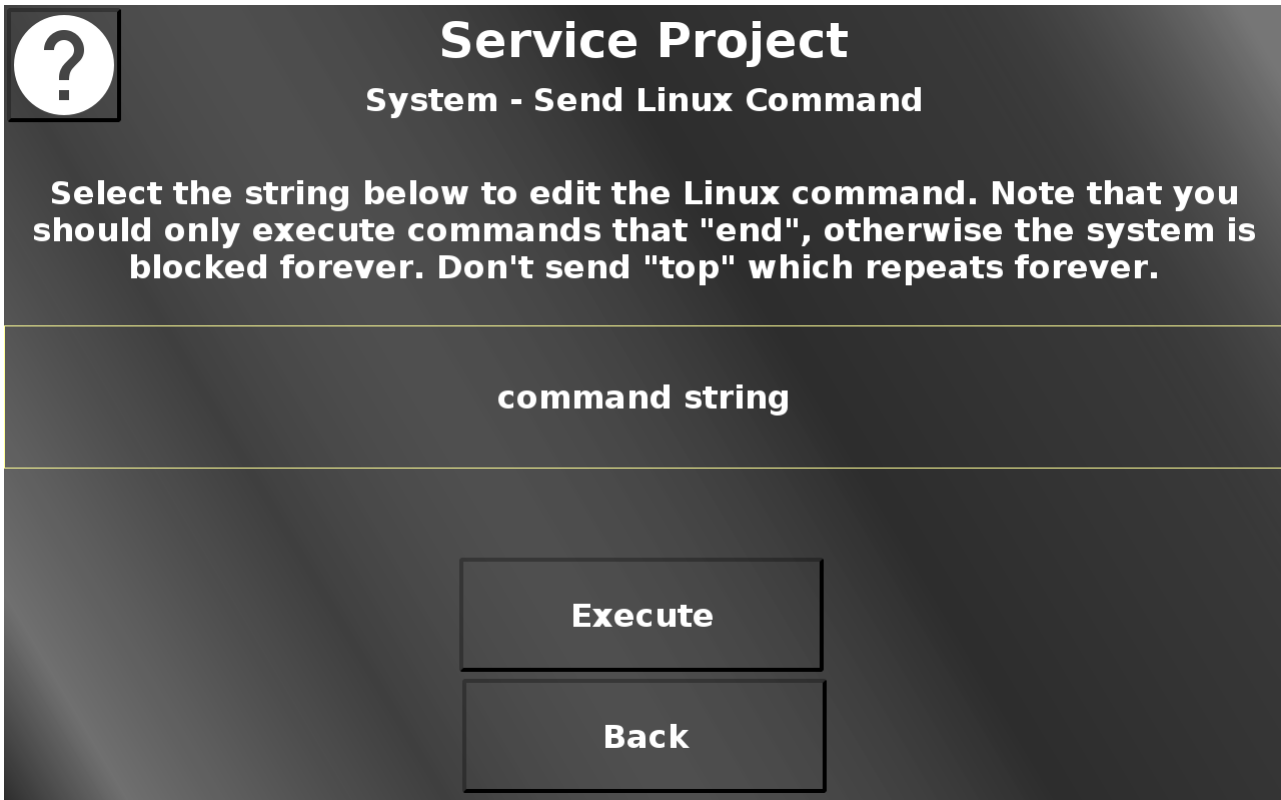
In this screen a Linux command can be executed. Select the field that says "command string" and a virtual keyboard will open. Enter a command of your choice. With the Execute button the command will be executed in the Linux system. If the command is faulty, it will be displayed that there was an error executing this command.

Please note that only valid commands should be entered, and only commands that will finish in a finite time. Since the execution of the command is done in JavaScript, the PClient will wait until the Linux command has finished before continuing. So a simple command like "top" will lead to a non-reactive device, because top will show the CPU usage of the processes and refresh that forever. In such a case the device has to be rebooted.

Back returns to the system menu.

Service Project -> System -> USB Update Check
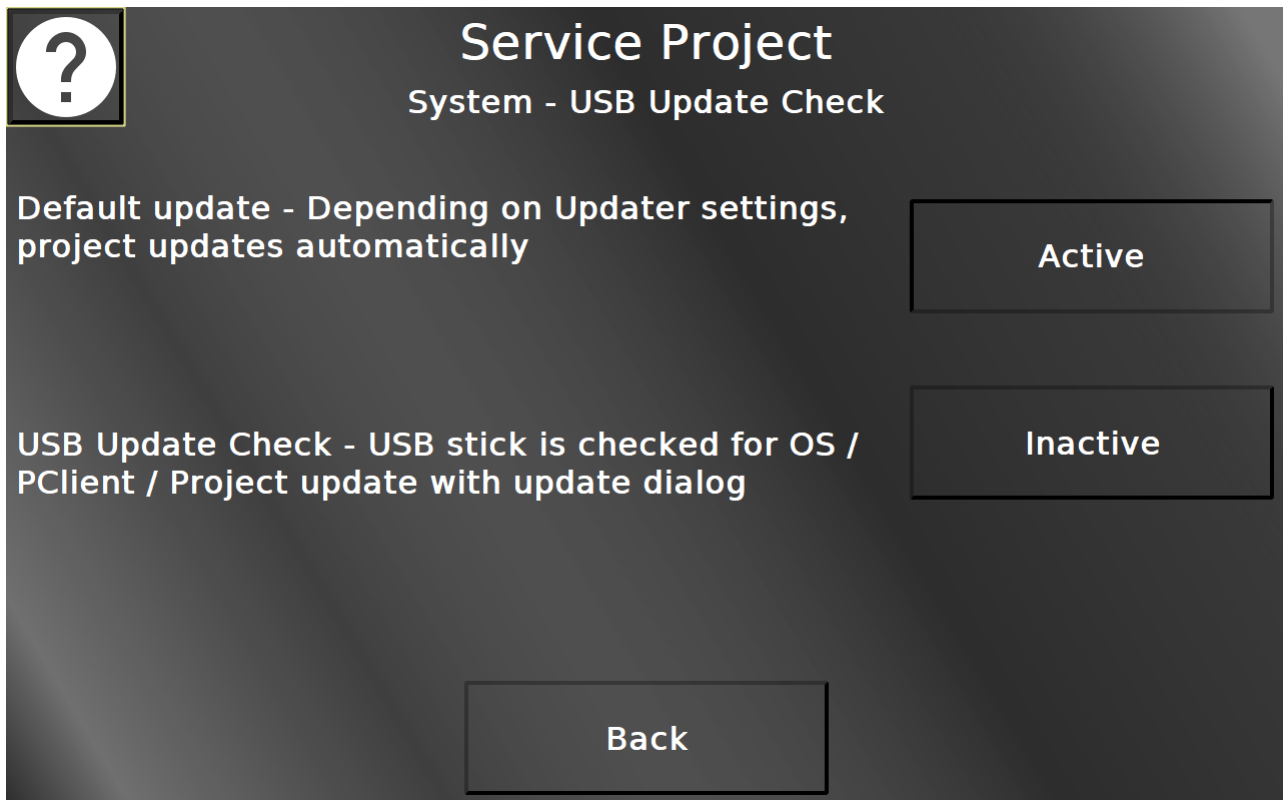
In this screen the update behavior when inserting a USB stick can be selected. The default update option (active by default) is the behavior as you know it - when a USB stick with a project is inserted and USB update is enabled in the Updater section, the project will be loaded.

With the USB Update Check the behavior will be different. If a USB stick is inserted, it is checked whether the USB stick contains a project and/or OS/PClient files. If it contains either of those, a popup alarm will be displayed, giving the options to load the project or to reboot in service mode to install the OS / PClient files.

Back returns to the system menu.
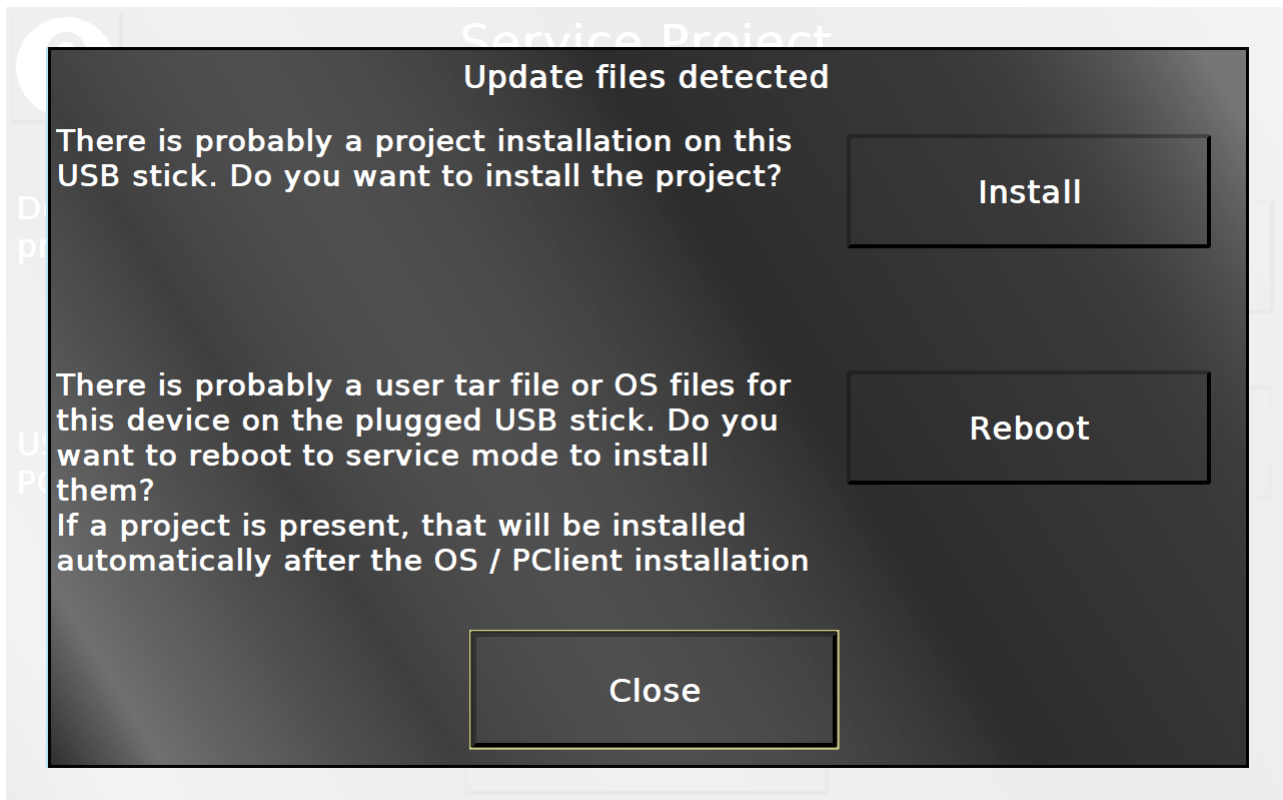
By pressing Install, the project on the USB stick will be installed.

By pressing Reboot, after a confirmation the device will be rebooted in service mode in which the OS / PClient files will be installed. Note that afterwards the device reboots again, starts in service mode, and if a PClient is installed and a valid project is on the USB stick, that project will be installed automatically, since the USB Update Check settings are volatile.

Close will return to the page that was shown before.

Service Project -> Sensors
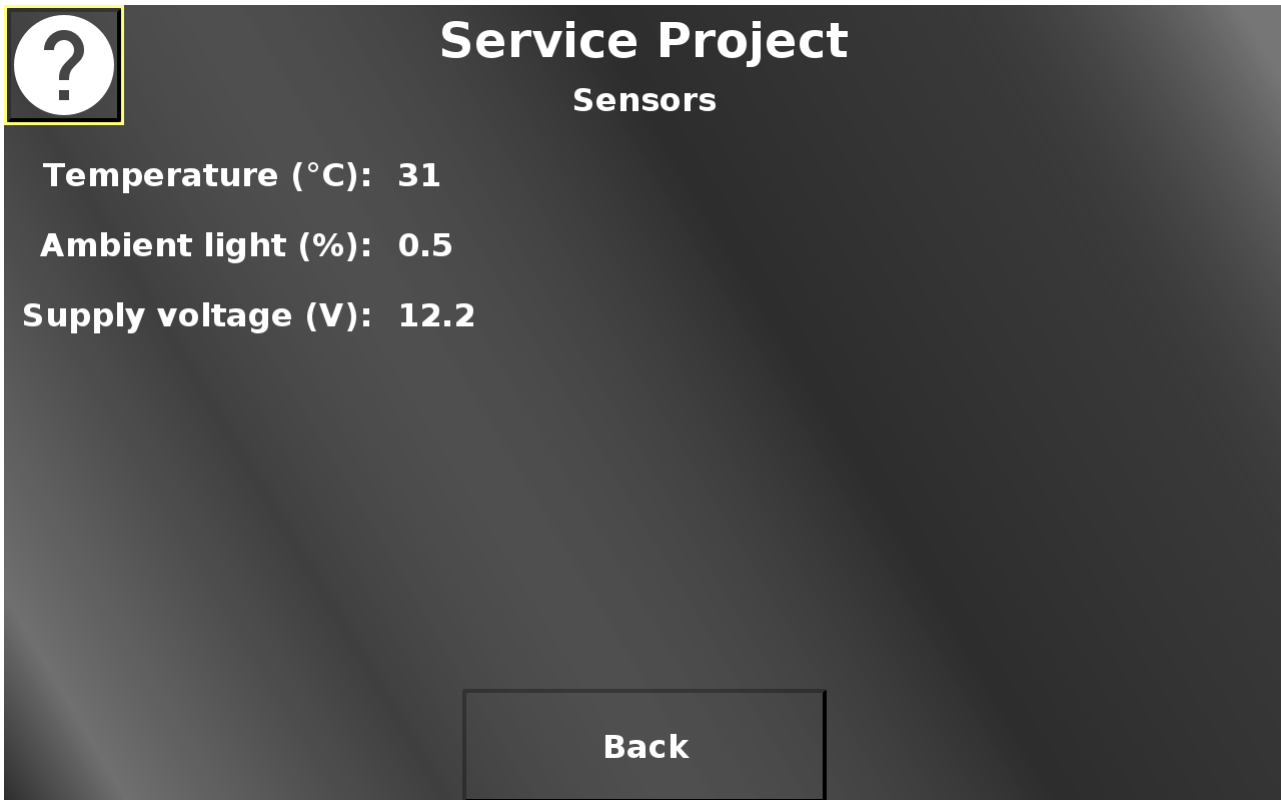
In this screen, the available hardware sensor values can be viewed. Back returns to the main menu.

Service Project -> Objects and Functions
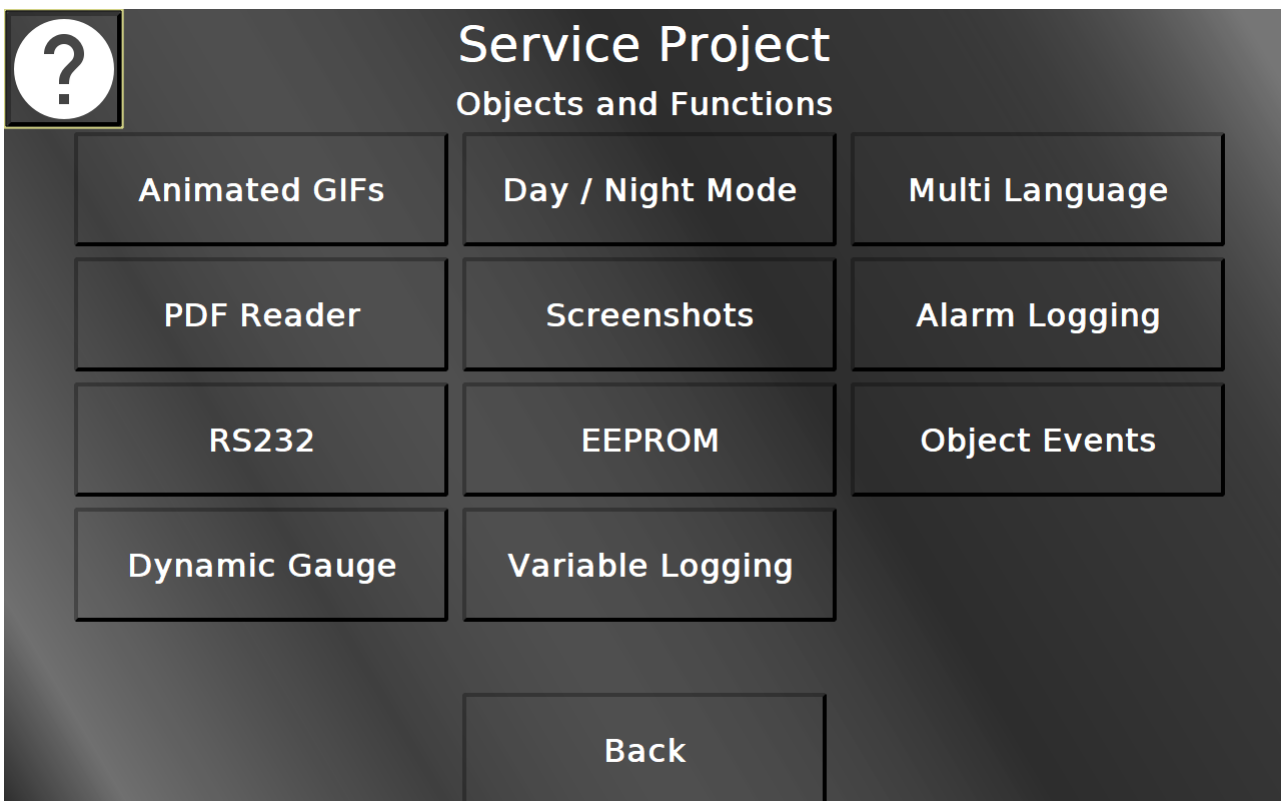
In this screen some functionalities of the OPUS Projektor and the PClient can be tested.

Service Project -> Objects and Functions -> Animated GIFs
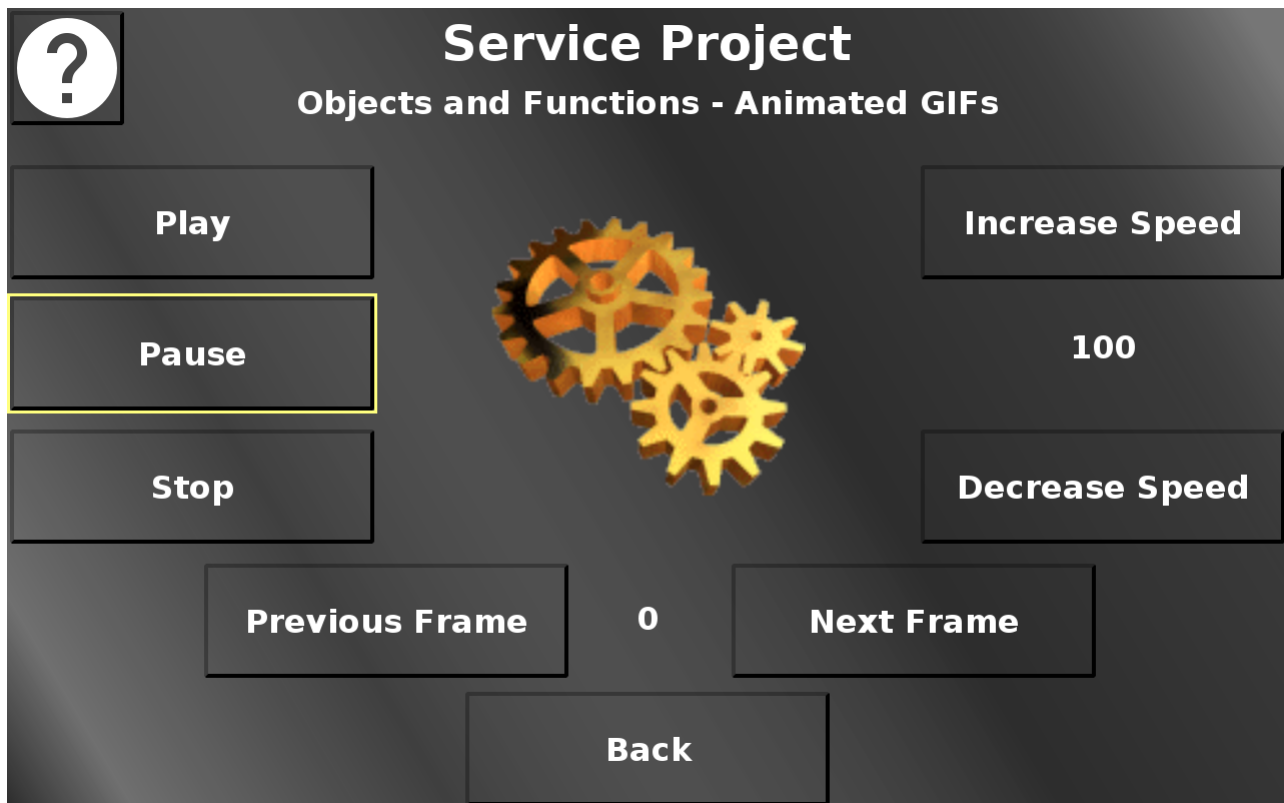
In this screen, an exemplary animated GIF file can be controlled.
Play starts the animation.
Pause pauses the animation (but stays at the current frame).
Stop stops the animation (and resets it to the first frame).
Increase speed increases the playing speed.
Decrease speed decreases the playing speed.
Back returns to the Objects and Functions menu.
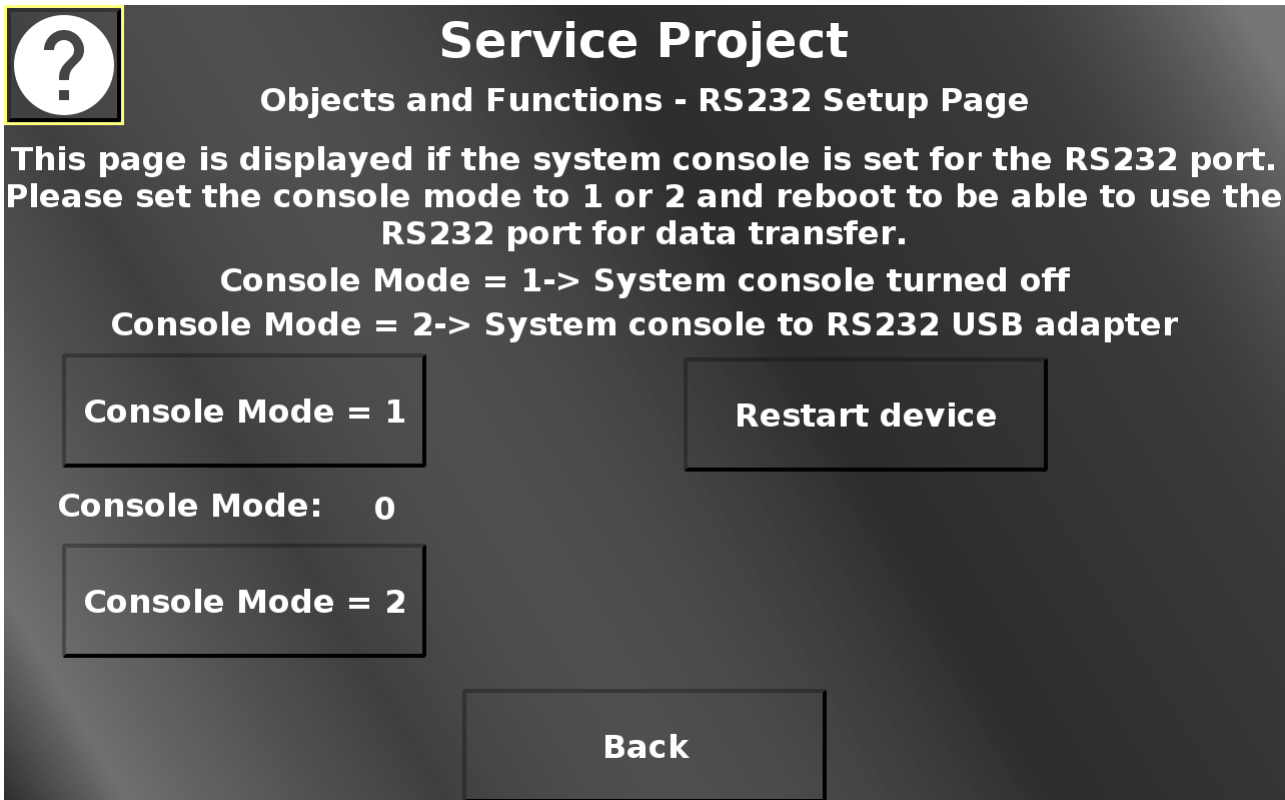
Service Project -> Objects and Functions -> PDF Reader
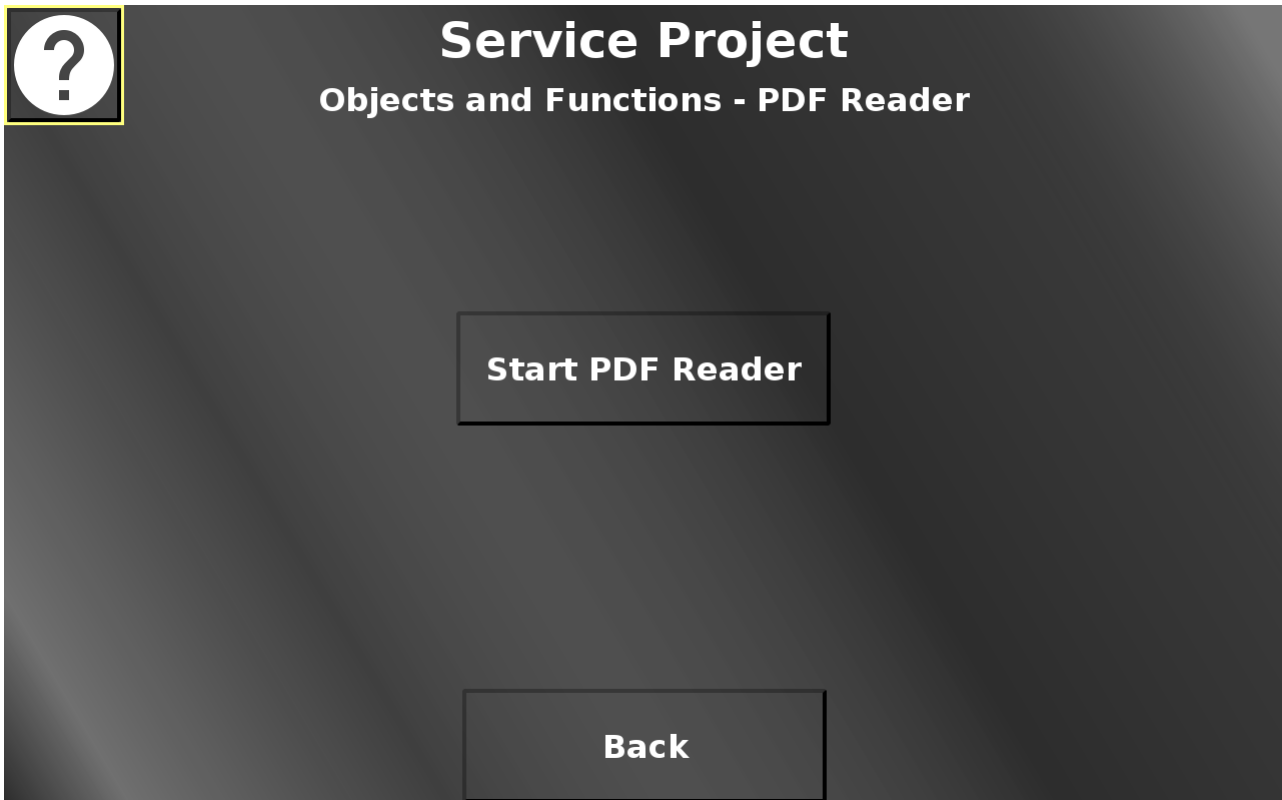
In this screen the PDF reader can be started with an exemplary PDF file.
Please note that it might take a while for the PDF reader to be started. Please be patient.
Please note that the PDF reader opens as a separate full screen application. Close it to go back to the PClient.
Back returns to the Objects and Functions menu.

Service Project -> Objects and Functions -> RS232 Setup Page

Service Project
Objects and Functions - RS232 Setup Page

This page is displayed if the system console is set for the RS232 port. Please set the console mode to 1 or 2 and reboot to be able to use the RS232 port for data transfer.

Console Mode = 1-> System console turned off
Console Mode = 2-> System console to RS232 USB adapter

Console Mode = 1   Restart device

Console Mode: 0

Console Mode = 2

Back
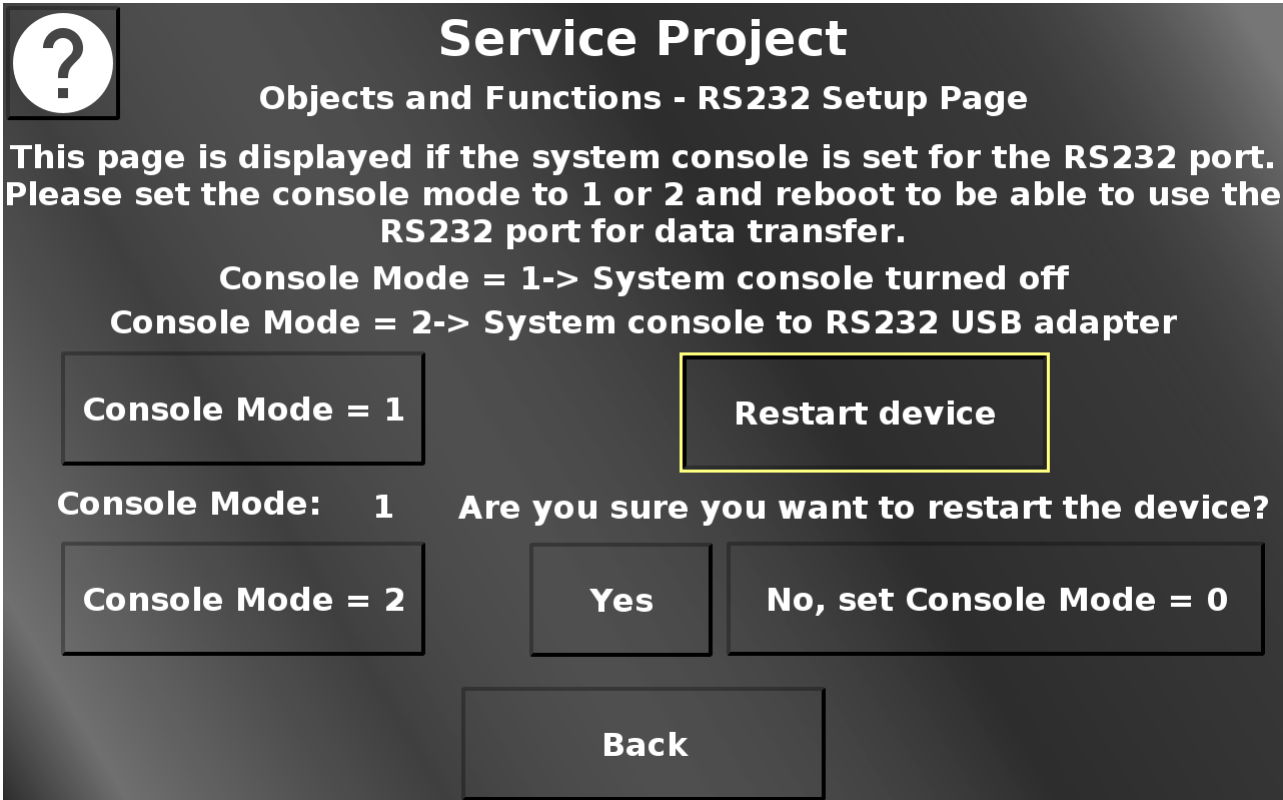
In this screen necessary settings to use the RS232 port for data communication can be made. By default, the RS232 port will be used by the device operating system console. To be able to use the RS232 port for data communication, the console mode needs to be set to 1 (system console off) or 2 (system console goes to the first USB-RS232 adapter (with USB side plugged into the device).
Please note that this screen is only displayed if the console mode is 0.
The device needs to be restarted for this setting to be activated.
The current console mode is displayed between the two console mode buttons.
Restart device opens a confirmation dialog.

Service Project
Objects and Functions - RS232 Setup Page

This page is displayed if the system console is set for the RS232 port. Please set the console mode to 1 or 2 and reboot to be able to use the RS232 port for data transfer.

Console Mode = 1-> System console turned off
Console Mode = 2-> System console to RS232 USB adapter
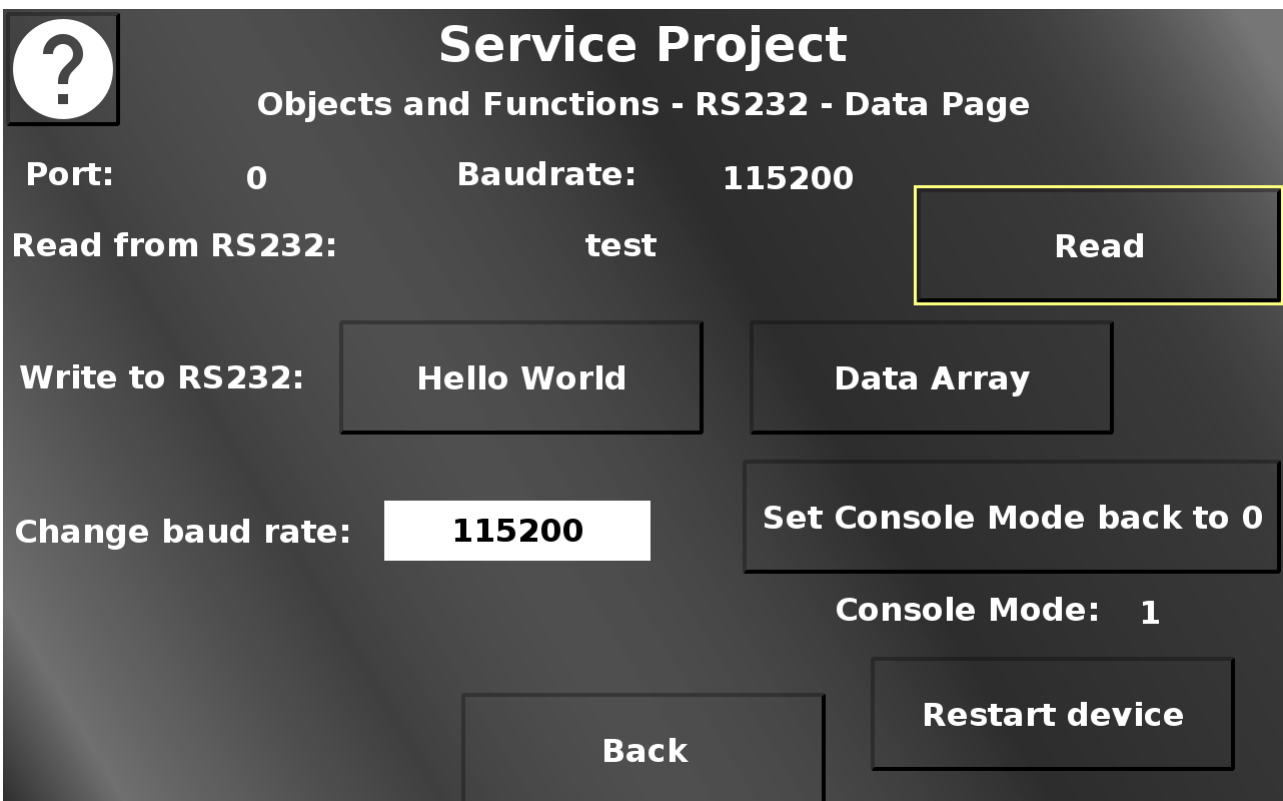
Console Mode = 1		Restart device

Console Mode: 1   Are you sure you want to restart the device?

Console Mode = 2	Yes	No, set Console Mode = 0

Back

Yes restarts the device with the chosen console mode setting. No closes the dialog and resets the console mode back to 0. Back returns to the Objects and Functions menu.

Service Project -> Objects and Functions -> RS232 Data Page

Service Project
Objects and Functions - RS232 - Data Page

Port: 0   Baudrate: 115200

Read from RS232:   test   Read

Write to RS232:   Hello World   Data Array

Change baud rate: 115200   Set Console Mode back to 0

Console Mode: 1

Back   Restart device

In this screen, RS232 data communication can be tested.
Please note that this screen is only shown if the console mode is 1 or 2 (see previous screens).
Read reads the RS232 buffer and displays the result (if it is text).
Hello World writes a "Hello World" to the RS232 port to be received by the other end.
Write Arrays writes a data array to the RS232 port to be received by the other end.
Set Console Mode back to 0 sets the console mode back to 0.
Restart device opens a confirmation dialog.

Service Project

Objects and Functions - RS232 - Data Page

Port:	0	Baudrate:	115200

Read from RS232: test **Read**

Write to RS232: **Hello World**   **Data Array**

Change baud rate: 115200   **Set Console Mode back to 0**

Are you sure you want to restart the device? Console Mode: 0

Yes **No** **Back** **Restart device**

Yes restarts the device with the current console mode setting.
No closes the dialog and reverts the changed console mode setting.
Back returns to the Objects and Functions menu.

Service Project -> Objects and Functions -> Dynamic Gauge

Service Project
Objects and Functions - Dynamic Gauge

Ticks -	Ticks +		Ticks Off	MinTicks Off
TickHeight -	TickHeight +		DigValue Off	Values Off
Values -	Values +	DirectionL>R	Min -	Min +
Boundary -	Boundary +		Max -	Max +
MinTicks -	MinTicks +	Reset	Value -	Value +

Back

In this screen the dynamic possibilities of the Gauge object can be tested. The different button change the object at runtime.
Reset resets the object to the default setup.
Back returns to the Objects and Functions menu.

Service Project -> Objects and Functions -> Day / Night Mode
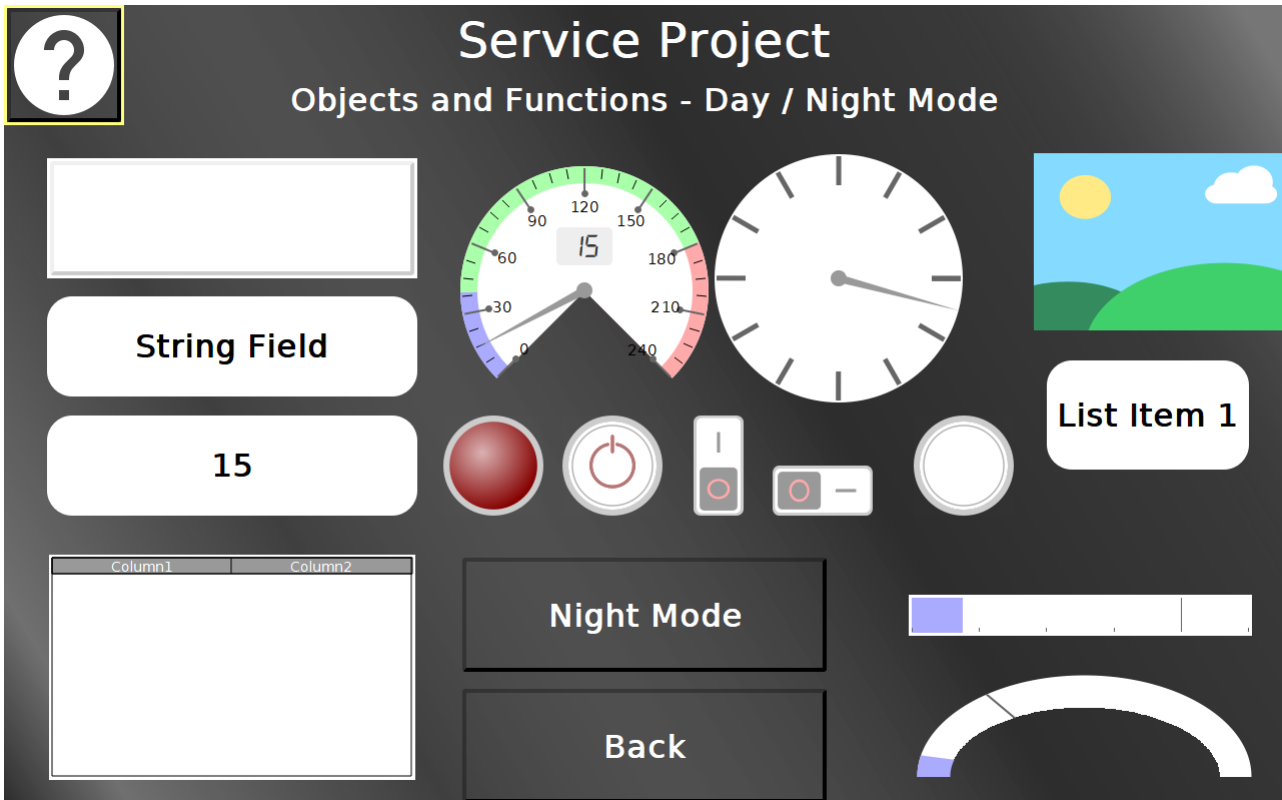
In this screen the day / night mode can be tested.

The Night Mode / Day Mode button switches all objects between their day mode / night mode theme.

The input objects can be tested, including the touch input for the list object.

Back returns to the Objects and Functions menu.

Service Project -> Objects and Functions -> Screenshots

Service Project

Objects and Functions - Screenshots

When screenshots are enabled, a screenshot can be taken by
a) a long press on the function key (3 arrow)
b) a press-release on the touch button in the upper right corner

Disable Screenshots

USB stick found. Please press button to copy screenshots.

Copy Screenshots   **Unmount USB stick**

Back

In this screen, screenshots can be enabled by pressing "Enable Screenshots". When screenshots are enabled, a screenshot can be taken on every page by either pressing the camera button in the upper right, or by long-pressing the 3 arrow key on the device. Screenshots will be stored on the file system. If a USB stick is plugged in and mounted, the screenshots can be copied to the USB stick by pressing "Copy Screenshots". The USB stick can be unmounted by pressing "Unmount USB stick". Back returns to the Objects and Functions menu.

Service Project -> Objects and Functions -> EEPROM

Service Project
Objects and Functions - EEPROM

Value Modifier: 1

Value to write

[-] 0 [+]   **Write value**

Position Modifier: 1

Position read / write

[-] 0 [+]   **Read value**

Value from EEPROM: 0

Back

In this screen the EEPROM can be tested by writing to it and reading from it. The value that should be written into an EEPROM position can be set between 0 and 255 with the + and - buttons. They will increment and decrement either by 1 or 10, depending on the value modifier setting. Write value will write the currently set value to the currently set position. The position can be set between 0 and the maximum for the device with the + and - buttons. They will increment and decrement by the value chosen in the position modifier drop down. Read value will read the value from the currently set position and display it. Back returns to the Objects and Functions menu.

Please note that EEPROMs have limited write cycles, so avoid writing into the same position too many times without reason to do so.

Service Project -> Objects and Functions -> Variable Logging

Service Project
Objects and Functions - Variable Logging

Stop Logging

Log file name: data_log_file.log

Log file size (kB): 3

USB stick found. Please press button to copy log files.

View Log File   **Copy Log Files**   **Unmount USB stick**

Back
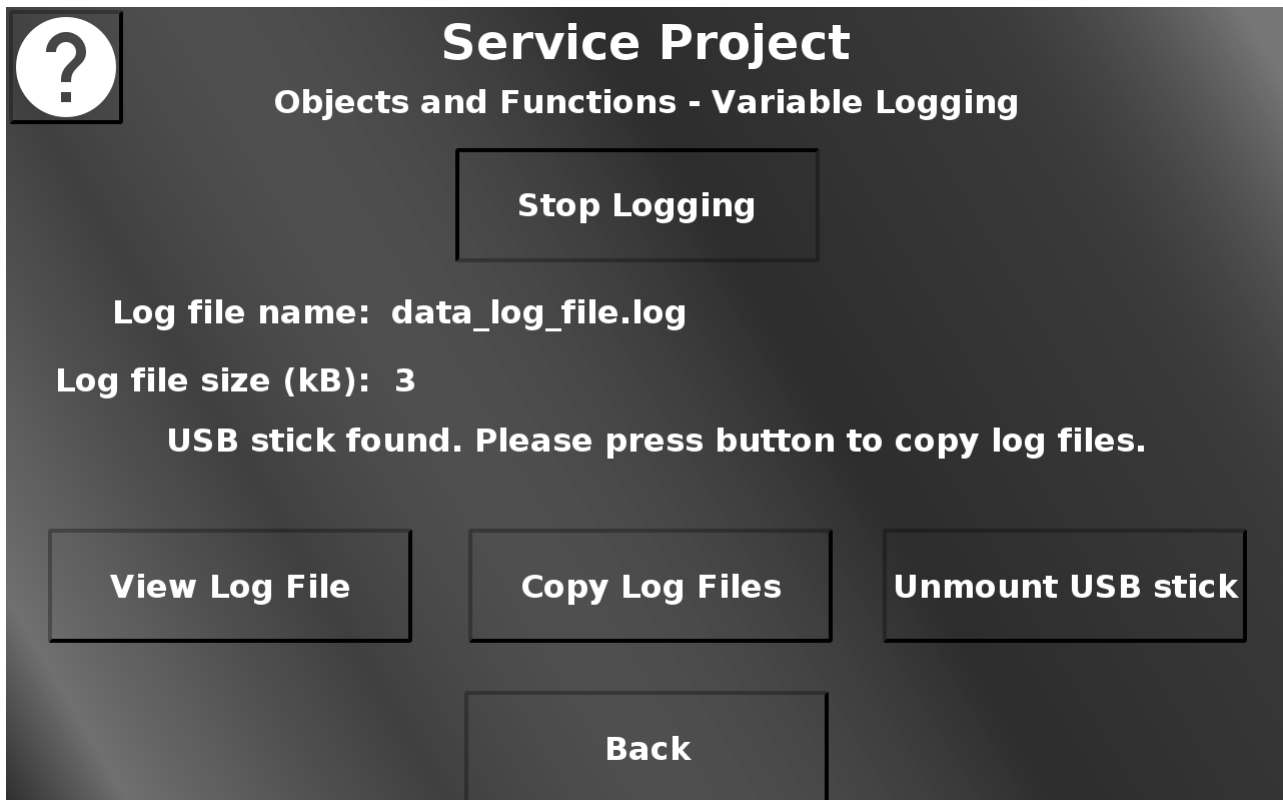
In this screen the variable logging functionality can be tested. Press Start Logging to start the variable logging. The logged variable is the RTC seconds variable, so there will be one event each second. The file can be viewed by pressing View Log File. This opens up an alarm showing the content of the file. Close the alarm by pressing Close. The Log File(s) can be copied to a connected USB stick by pressing Copy Log Files. Press Unmount USB stick to unmount the USB stick. Back returns to the Objects and Functions menu.

Service Project -> Objects and Functions -> Multi Language

In this screen the multi language functionality can be tested. The five language buttons switch the texts and the image between the languages. Back returns to the Objects and Functions menu.
The used language flag icons were created by Freepik from www.flaticon.com. Back returns to the Objects and Functions menu.

Service Project -> Objects and Functions -> Alarm Logging
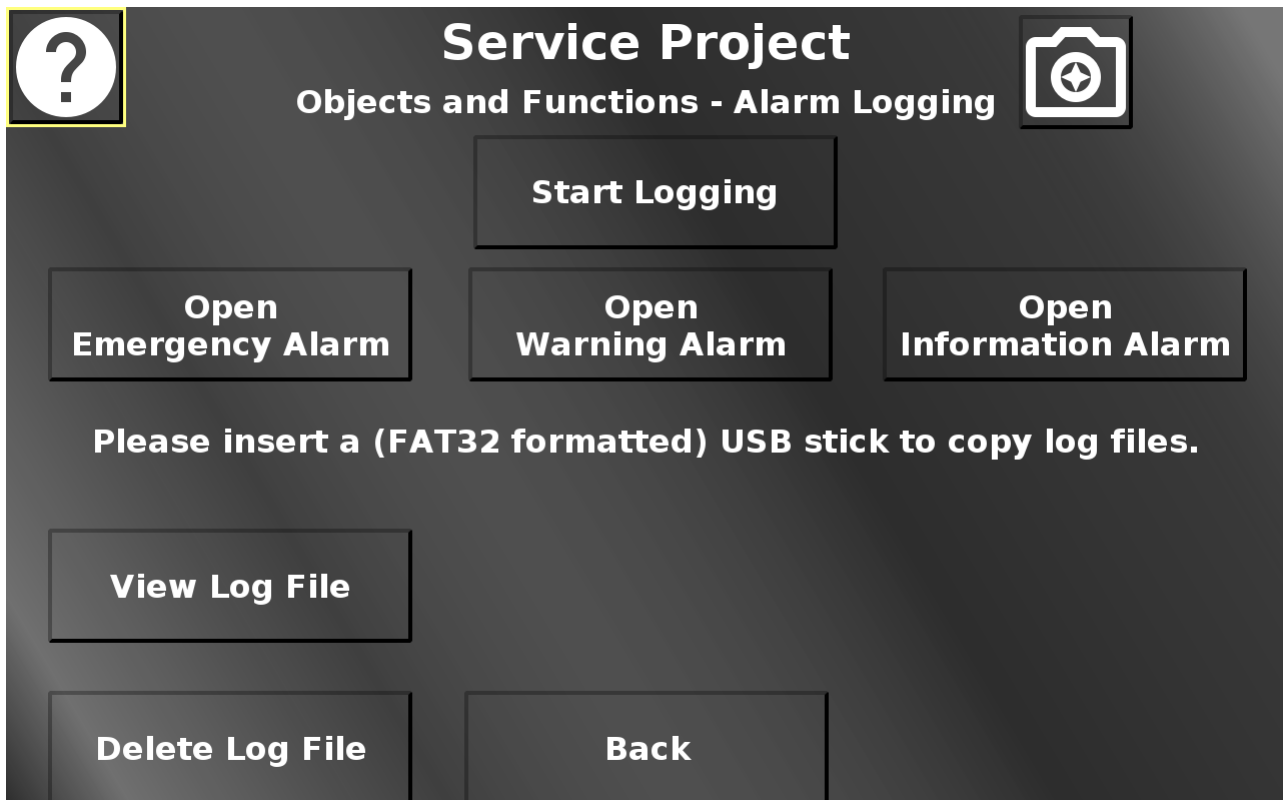
In this screen alarms and alarm logging can be tested. To enable alarm logging, press the Start Logging button. To stop it, press it again. The three Open ... Alarm buttons open an alarm from one of the categories Emergency, Warning or Information(al). From the opened alarm pop up, the alarms of the other two categories can be opened as well, to demonstrate the enqueuement and priority of the different alarm categories. View Log File opens an alarm that shows the newest entries in the alarm log file. Press Delete Log File to delete the current log file. If a USB stick is inserted, the alarm log file can be copied to the USB stick. The USB stick can also be unmounted.
Back returns to the Objects and Functions menu.

Service Project -> Objects and Functions -> Object Events

Service Project
Objects and Functions - Object Events

Choose object   **Numeric Field**

OnValueChange

OnEnterValue

OnEnterNewValue

OnEscape

OnOpenForInput

VKB

15

Encoder

15

Clear all

Back

OnEnabledWithFocus

OnEnabledWithoutFocus

~~Disabled~~

~~Send Directly~~

On this page most object events of all input objects can be tested. In the list object in the top an object type can be selected, from String Field, Numeric Field, Button, Table, List, Meter, Gauge, Arched Bargraph, Linear Bargraph.

For the selected object type, the available events are listed on the left and right of the object in the center. Use the object and/or change its value to activate the events. An activated event will get a green background.

With Clear all, the event backgrounds can be reverted to white.

With the disabled button the object under test can be disabled. That will deactivate it as an input, and it will be shown grayed out and strike through.

With the Send Directly button the behavior of the according object can be changed. With send directly, each encoder rotation will directly invoke a value change. Without it, only when confirming the value, it will actually be changed.

Back returns to the Objects and Functions menu.

Service Project -> Info to USB
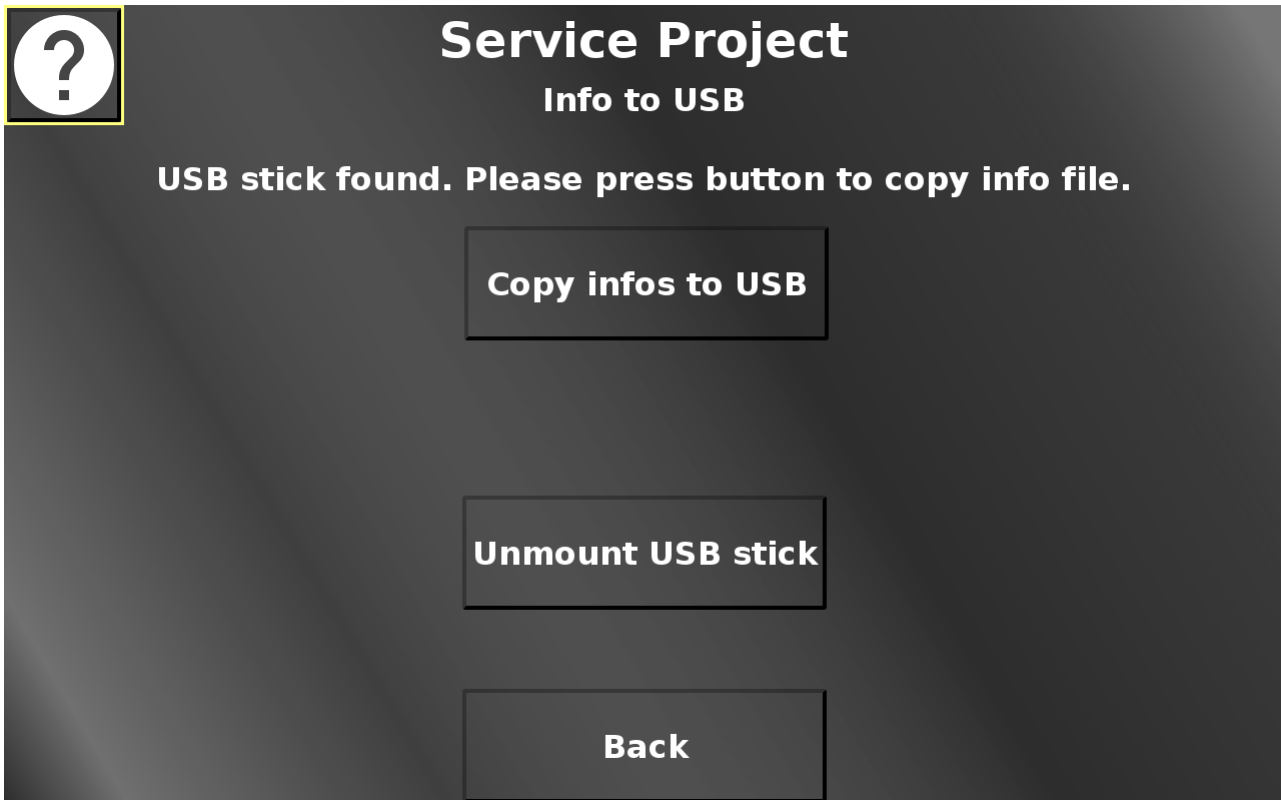
In this screen a file with information about the device can be written to a connected USB stick for analysis or to be sent to technical support.
Copy infos to USB creates the info file and puts it on the USB stick. Please note that this button is only shown if a valid USB stick is connected.
This is an exemplary content of an info file:

Device Info
Article number: OPUSA8SN1CANB000
Serial number: 1611001AA
Device type: 12.1" Standard Basic
OS version: 4.1.15-egspro-2.3.2-1
PClient version: 2020.4.0.2
Project version: 0.1
Operating time: 2 d, 7 h, 12 m
System Uptime: 0 h, 6 m, 28 s
Date / Time: 2020/2/28 - 14/16/18
System memory: 1192.784 MB / 1312.900 MB
Device temperature: 41.2 °C
Device voltage: 24.4 V

Service Project
Info to USB

USB stick found. Please press button to copy info file.

Copy infos to USB

Copy finished. Unmount the USB stick before ejecting it.
File name:
/disk/usbsda1/device_info_2020-4-2_6-58-52.txt

Unmount USB stick

Back

After copying, the file name of the created file will be shown.
Unmount USB stick unmounts the USB stick so it can be ejected from the device.
Please note that until the USB stick is mounted, the file might not be finalized on the USB stick. Ejecting the USB stick before unmounting means a probable loss of the info file.

Service Project
Info to USB

Please insert a (FAT32 formatted) USB stick to copy the info file.

Unmount finished. Please eject the USB stick.

Back

When the USB stick is successfully unmounted, an information will be displayed on the screen that the stick can be ejected.
Back returns to the main menu.

Service Project -> CAN
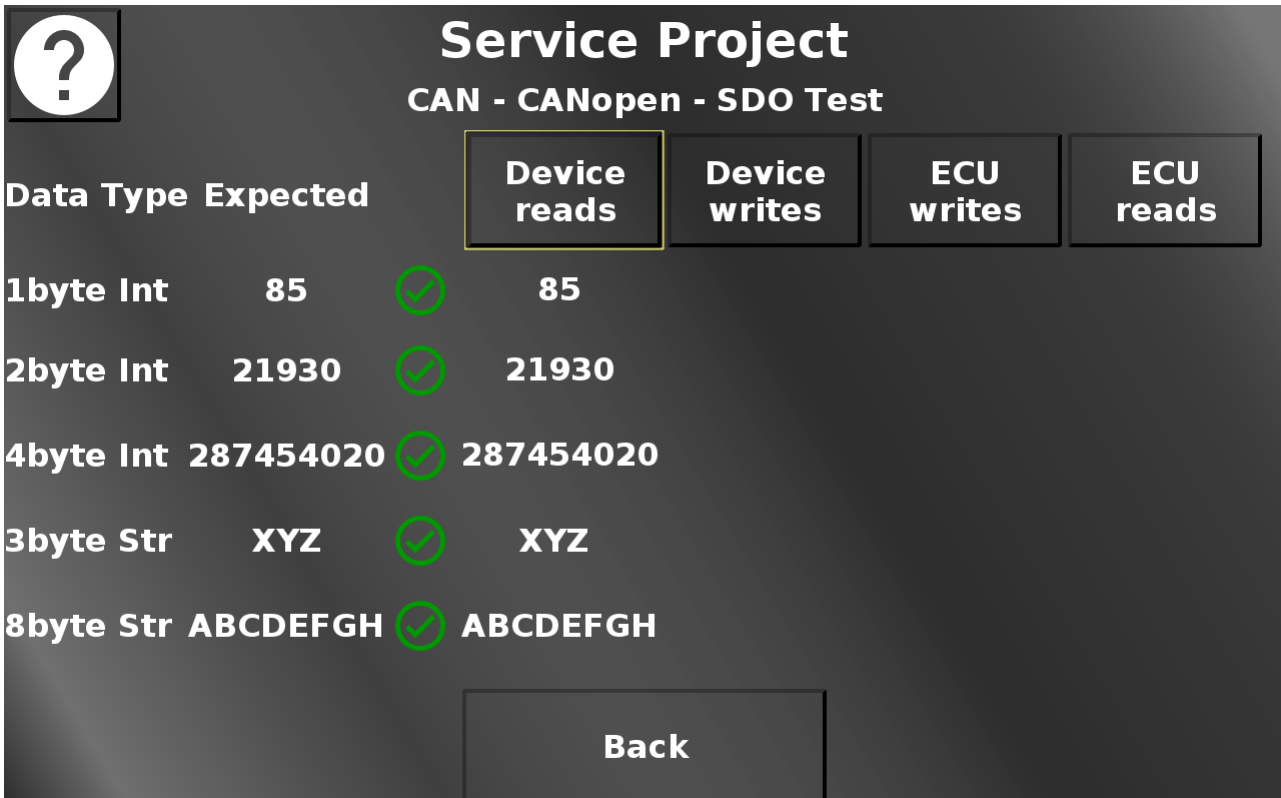
In this screen the different CAN protocols can be selected for some tests.
Please note that if the device is connected to a CAN network, it is your responsibility that the messages created in these tests don't collide with the CAN network.
Back returns to the main menu.

Service Project -> CAN -> CANopen

Service Project
CAN - CANopen

SDO Test

Connect CAN1 and CAN2 for this test
The ECU will be simulated by the other CAN port.

PDO Test

Connect CAN1 and CAN2 for this test
The ECU will be simulated by the other CAN port.

Heartbeat Test

Connect CAN1 and CAN2 for this test
The ECU will be simulated by the other CAN port.
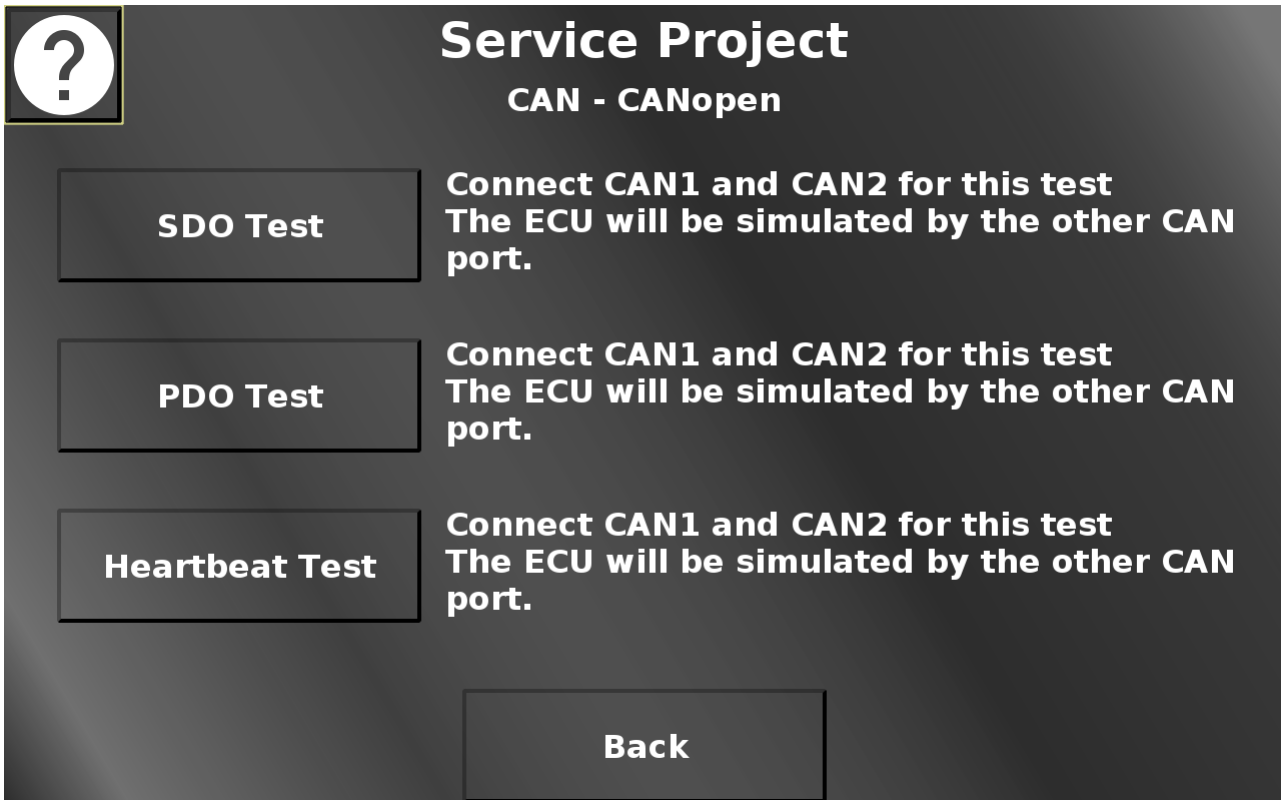
Back

In this screen the SDO and the PDO Test can be selected. Please note that for these tests to work, the device ports CAN 1 and CAN 2 need to be connected.
Back returns to the CAN menu.

Service Project -> CAN -> CANopen -> SDO Test

Service Project

CAN - CANopen - SDO Test

Data Type	Expected		Device reads	Device writes	ECU writes	ECU reads
1byte Int	85	✓	85			
2byte Int	21930	✓	21930			
4byte Int	287454020	✓	287454020			
3byte Str	XYZ	✓	XYZ			
8byte Str	ABCDEFGH	✓	ABCDEFGH			

Back

In this screen different SDO functionalities can be tested.

Device reads means that the OPUS device will ask the ECU (aka CAN port 2) for some data values. If the communication is successful, the values in the Expected column should be the same as in the ask ECU column. The check boxes show if the values match. Create a CAN trace to see the communication process.

Device writes means that the OPUS device will send a write request to some variables belonging to the ECU (aka CAN port 2). If the communication is successful, the values in the Expected column should be the same as in the Device writes column. The check boxes show if the values match. Create a CAN trace to see the communication process.

ECU writes means that the ECU (aka CAN port 2) writes some variables in the OPUS device. If the communication is successful, the values in the Expected column should be the same as in the ask ECU writes. The check boxes show if the values match. Create a CAN trace to see the communication process.

ECU reads means that the ECU (aka CAN port 2) reads some variables from the OPUS device. If the communication is successful, the values in the Expected column should be the same as in the ECU reads column. The check boxes show if the values match. Create a CAN trace to see the communication process.

Back returns to the CANopen menu.

Service Project -> CAN -> CANopen -> PDO Test
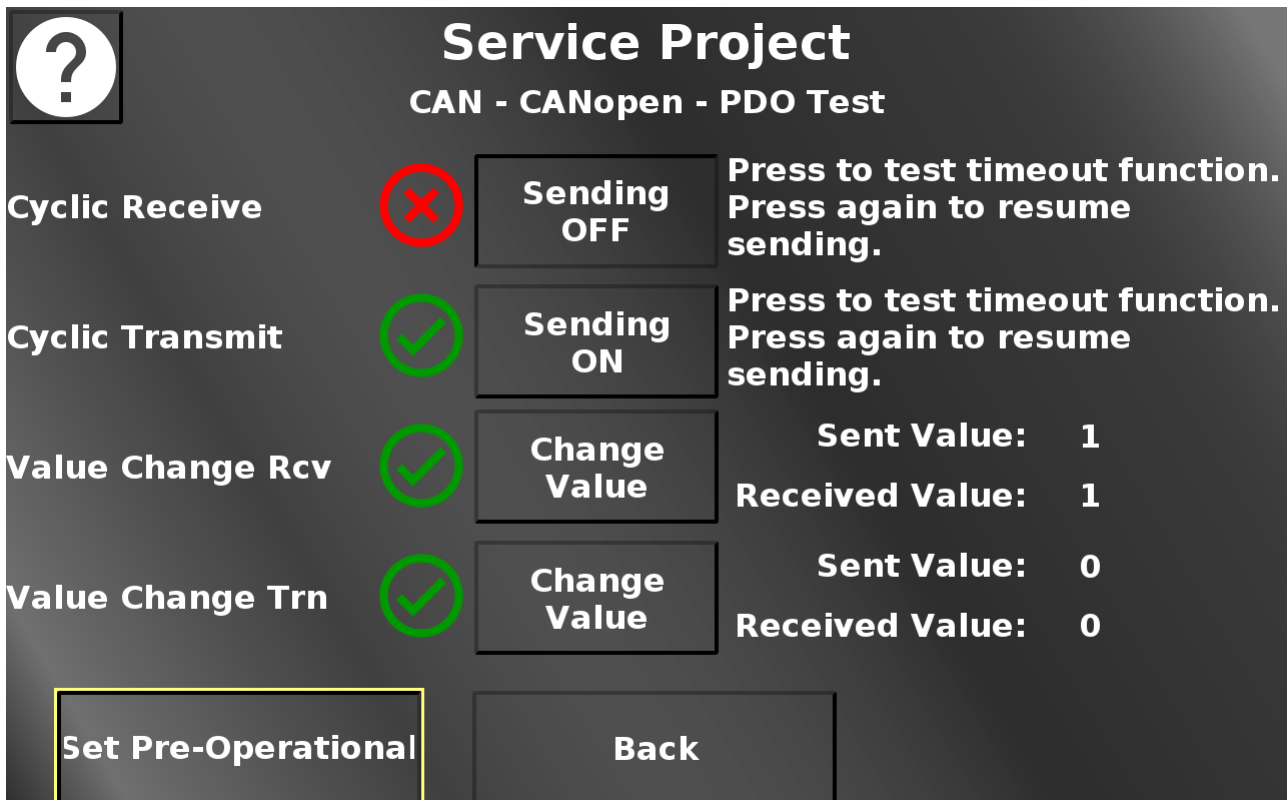
In this screen some PDO functionalities can be tested. For the PDO communcation to start, the device first needs to be set into operational mode with the button in the bottom left.

The first two indicators show a green check mark if the cyclic receive and transmit communication work correctly. You can disable the sending of the appropriate message with the according button. This should make the red X appear. Note that the ECU is simulated with CANFreestyle mappings on CAN port 2.

The second two indicators show a green check mark if the sent value and the received value of a CANopen Receive(Rcv) and Transmit (Trn) mapping are identical.
Press the buttons to change the sent value. The received value should change to the same almost instantly. Only very quick taps on these buttons should make the red X appear.
Initially, the received values are set to invalid, so the red X will appear for both values. Once this communication was established, the green check mark will be displayed as long as the values are the same, even if the device is set to pre-operational (i.e. it cannot send or receive messages). Press the buttons to change the sent value and the red X should appear.
Also note that once the device is back to operational, only the Transmit (Trn) value might automatically update. This is a property of the CANopen standard, when a device first goes into operational mode, it will once send all transmit mappings with their current values. The Receive (Rcv) value is being sent by a CANFreestyle mapping on CAN port 2 which does not have such a feature. Press the Change Value button, then the received value should also update and the green check mark appears.

Back returns to the CANopen menu.

Service Project -> CAN -> CANopen -> Heartbeat Test
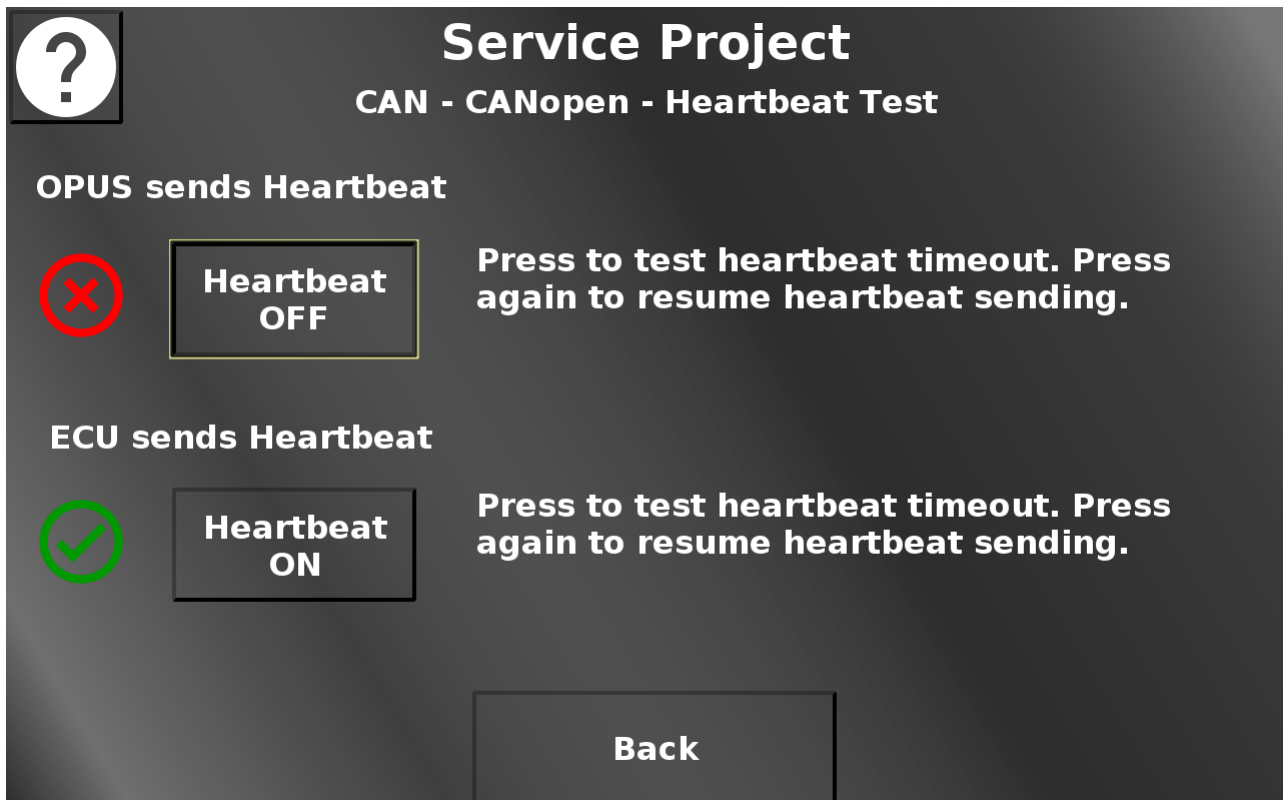
In this screen the CANopen heartbeat functionality can be tested.

When entering this page, the CANopen stack on the device will produce a heartbeat message every 100 ms (ID 0x701) on CAN bus 1. A CANFreestyle receive message on CAN bus 2 checks for this message and shows the red X if the heartbeat isn't received for more than 200 ms. If the heartbeat is received again, the green check mark is displayed.

Simulating an ECU, a CANFreestyle transmit message on CAN bus 2 will be sent to imitate a heartbeat message from the ECU every 100 ms. The CANopen stack on CAN bus 1 will evaluate the message. If it hasn't been received for 200 ms, the red X will be shown. If the heartbeat message is received again, the green check mark is displayed.

Back returns to the CANopen menu.

Service Project -> CAN -> J1939

Service Project
CAN - J1939

| DM1 Test | Connect CAN1 and CAN2 for this test |
| PGN Test | Connect CAN1 and CAN2 for this test |
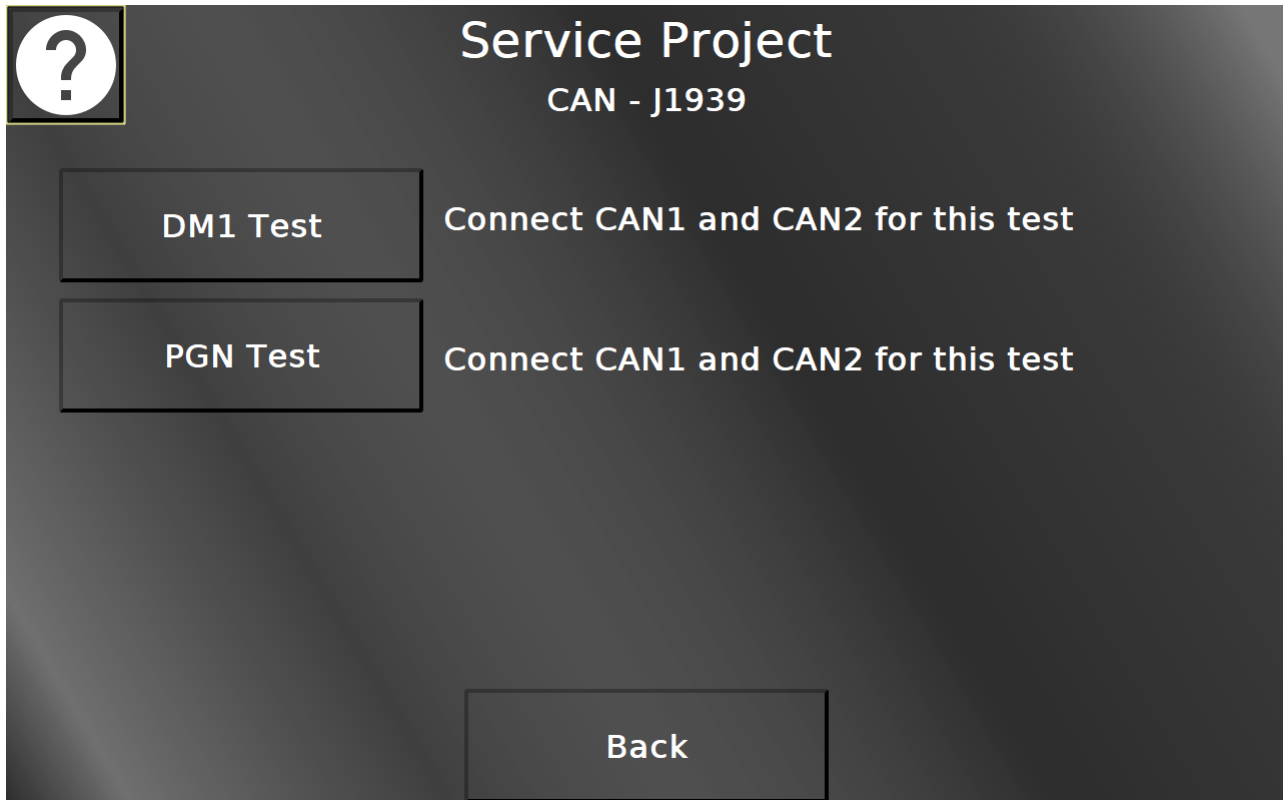
Back

In this screen the test pages for DM1 and PGNs can be opened. Please note that for these tests to work, the device ports CAN 1 and CAN 2 need to be connected.
Back returns to the CAN menu.

Service Project -> CAN -> J1939 -> DM1

In this screen two DM1 tests can be performed.
Send Example 1 will send an error to the OPUS device and let one lamp blink slowly.
Send Example 2 will send a different error to the OPUS device and let all lamps blink fast.
Please note that the errors will disappear after 1 second. DM1 messages have to be sent every 1 s which is why the buffer is automatically cleared after 1 second.
Back returns to the J1939 menu.

Service Project -> CAN -> J1939 -> PGN Test

On this screen, two PGNs can be tested. PGN FDC5 contains some device information. With Send PGN FDC5 the PGN is sent from CAN bus 1 to 2 and the received values are compared to the actual values on the left. If they are correct, green check marks will be displayed. If they are not, a red cross will be displayed instead. The values for Location, Type, Manufacturer and Hardware ID can be edited.

Pressing Enable PGN 0 starts the periodic transfer of PGN 0, usually used to transfer the target RPM. The gauge on the left side shows the value that is transferred, the gauge on the right shows the received value. With the plus / minus buttons the value can be changed.
If the values are identical, a green check mark will be displayed. If they are not, a red cross will be displayed instead.
Back returns to the J1939 menu.

Service Project -> CAN -> CANFreestyle

In this screen the mask, bit shift, scaling and offset test page and a page to send a custom CAN message can be opened. Back returns to the CAN menu.

Service Project -> CAN -> CANFreestyle -> Mask / Shift Test

Service Project

CAN - CANFreestyle - Mask / Shift Test

?

Original Value: 0x8   [-] [+]   Modifier: 0x1

2 bytes, 1 bit right: 0x4	✓	0x4
2 bytes, 1 bit left: 0x10	✓	0x10
2 bytes, offset1 50, scale 0.5: 0x1D	✓	0x1D
2 bytes, scale 0.5, offset2 50: 0x36	✓	0x36
2 bytes, mask 0x5555: 0x0	✓	0x0
4 bits, 3 bits left: 0x40	✓	0x40
4 bits, 2 bits right, mask 0x2, scale 2: 0x4	✓	0x4
Kelvin -> Fahrenheit: -445	✓	-445

Expected Values

[Back]

On this page, different masking, bit shift, scaling and offset functionalities can be tested. For this test to work, CAN1 and CAN2 have to be connected.

In the top the original value can be changed (shown in hexadecimal). The Modifier dropdown chooses the modifier by which plus and minus buttons change the value. In each line below some modifications of the original value are configured in the CAN receive mapping. The left value shows the modified value. The right value shows the same modification done in JavaScript. The two value are compared every time the value is changed and the CAN message is sent again. If the values match, a green check mark is shown. If the values don't match, a red X is shown.

Back returns to the CAN Freestyle menu.

Service Project -> CAN -> CANFreestyle -> Custom CAN Message

In this screen a custom CAN message can be set up which will be sent to the chosen CAN bus.

Please note that creating the "wrong" message can be dangerous for the system the device is connected to.

The CAN 1 / CAN 2 button chooses over which CAN bus the message should be sent.

In the Bitrate dropdown the bitrate can be chosen with which the message should be sent.

In the DLC drowpdown the length of the message can be chosen. Values from 0 to 8 are possible.

The CAN ID - (minus) and + (plus) buttons change the value of the CAN ID of the message.

The Modifier dropdown chooses the modifier by which the CAN ID will be changed. In the displayed setup, a press of the + button will change the CAN ID from 0x1A0 to 0x1B0.

The + and - buttons of the data bytes change the data value of each byte.

The segments for the data bytes will be made invisible depending on the DLC selection.

The modifier button in the bottom switches between 0x1 and 0x10. The - / + buttons for the data bytes modify the value by the selected modifer.

Send Message sends the CAN message with the chosen settings.

Back returns to the CAN menu.

Service Project -> LED / Backlights

Service Project
LED / Backlights

Display Backlight: [-] 80 [+]   LED Color Red: [-] 0 [+]

Key Backlight: [-] 50 [+]   LED Color Green: [-] 0 [+]

Enc. Backlight: [-] 50 [+]   LED Color Blue: [-] 0 [+]

[Store display brightness as startup value]   [LED Auto Mode on]

[Disp Back Auto]   [Key / Enc Colors]   Modifier:
                                               [10]
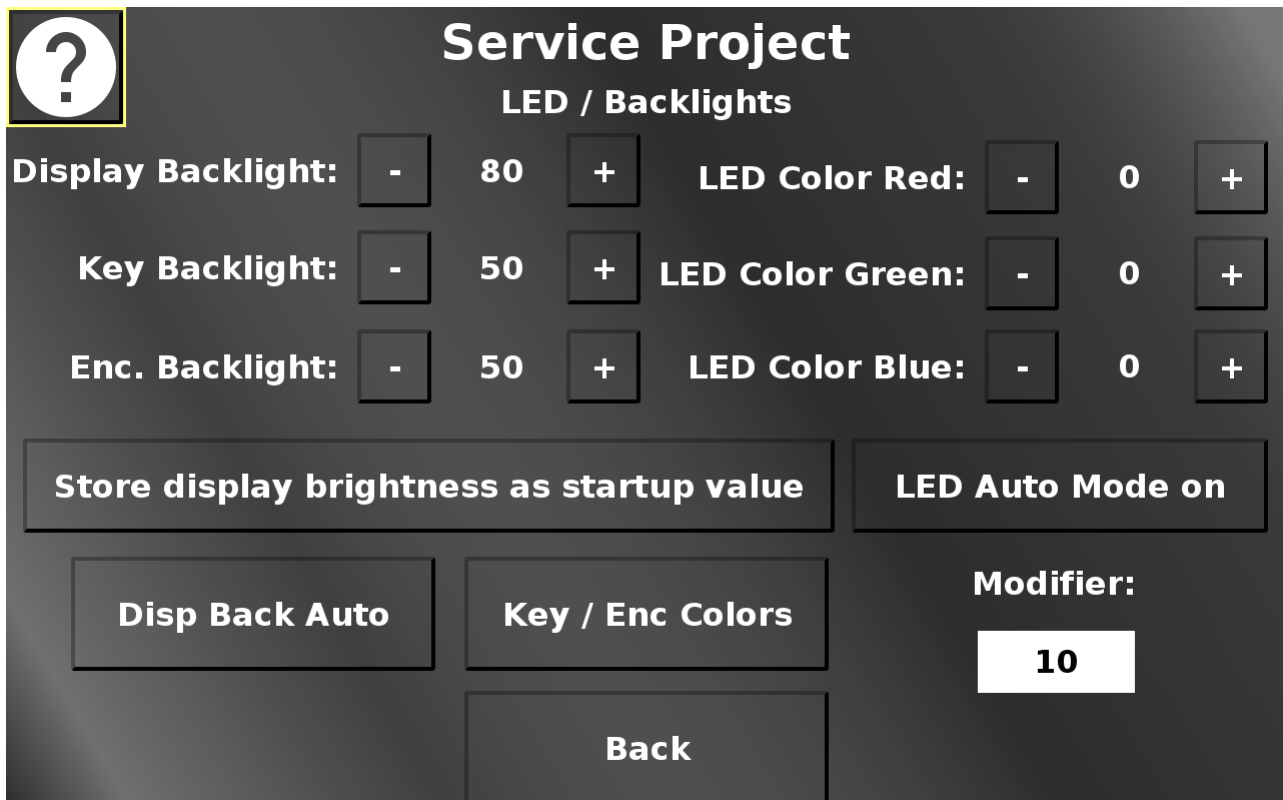
[Back]

The LED / Backlights screen allows changing the values for display, key, encoder backlight and the multicolor LED.

The - (minus) and + (plus) buttons change the according value. The amount that the current value is modified by, can be chosen with the Modifer dropdown.

Store display brightness as startup value changes the startup display brightness (i.e. the screen brightness when the device is powered on until the application setting sets in).

LED Auto Mode on starts gradual color changes for the multi color led.

Disp Back Auto opens the automatic backlight setup screen.

Key / Enc Colors opens the screen for key and encoder backlight colors.

Back returns to the main menu.

Service Project -> LED / Backlights -> Display Backlight Auto Mode
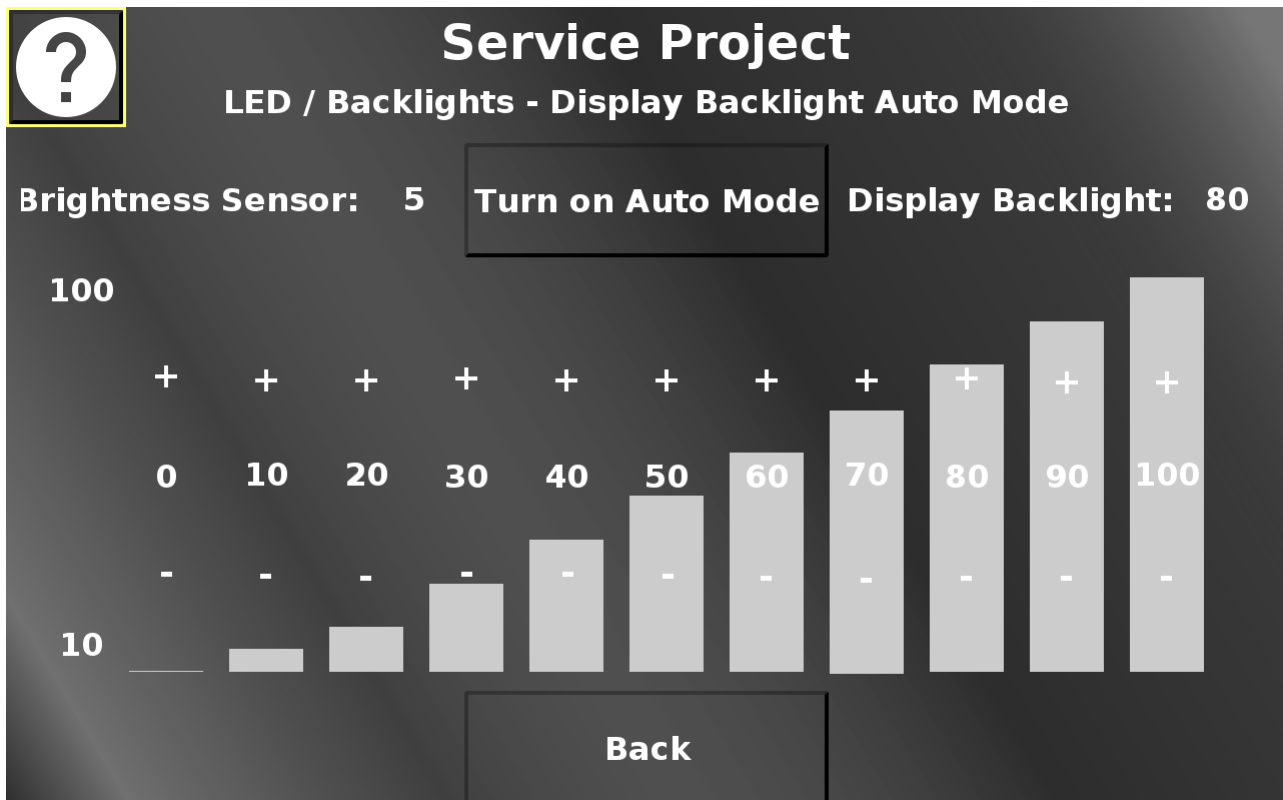
In this screen the automatic backlight control can be activated / deactivated.
Additionally, the backlight curve for the different brightness sensor values can be adjusted.
The first bar sets the brightness of the screen for an ambient brightness of 0 to 10 %.
The second bar sets the brightness of the screen for an ambient brightness of 10 to 20 %,
and so on.
The - (minus) and + (plus) sections decrease / increase the brightness value for the according
ambient brightness.
Back returns to the LED / Backlight menu.

Service Project -> LED / Backlights -> Keyboard / Encoder Colors

Service Project
LED / Backlights - Keyboard / Encoder Colors

Key Color Red:	-	255	+	Enc Color Red:	-	255	+
Key Color Green:	-	0	+	Enc Color Green:	-	0	+
Key Color Blue:	-	255	+	Enc Color Blue:	-	255	+

< copy | copy >

Store as startup value
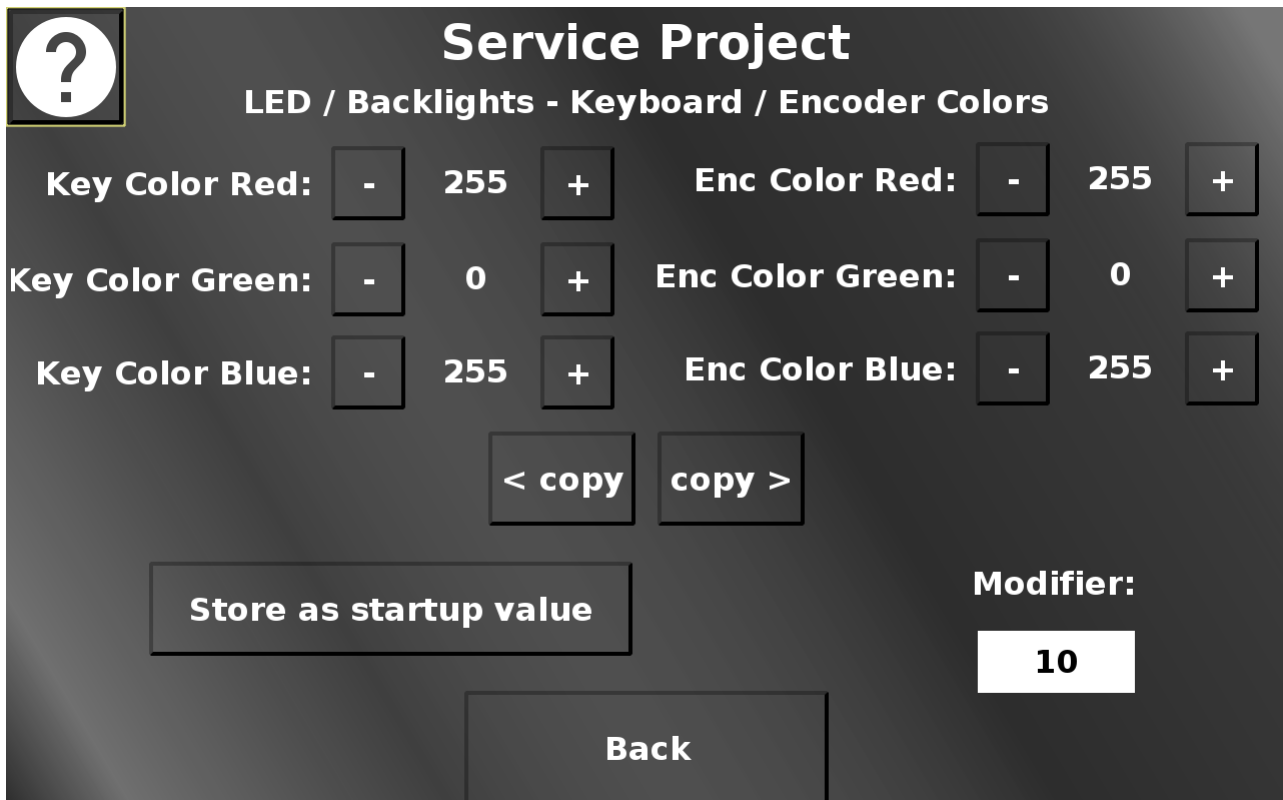
Modifier:
10

Back

In this screen the colors for the key backlight and the encoder backlight can be adjusted (only available on certain devices).
The - (minus) and + (plus) buttons change the according value. The amount that the current value is modified by, can be chosen with the Modifer dropdown.
Store as startup value will set the backlight color that keys and encoder will have at startup until the application setting sets in.
The copy > and < buttons will duplicate the key color values to the encoder values and vice versa.
Back returns to the LED / Backlight menu.

Service Project -> Inputs / Outputs
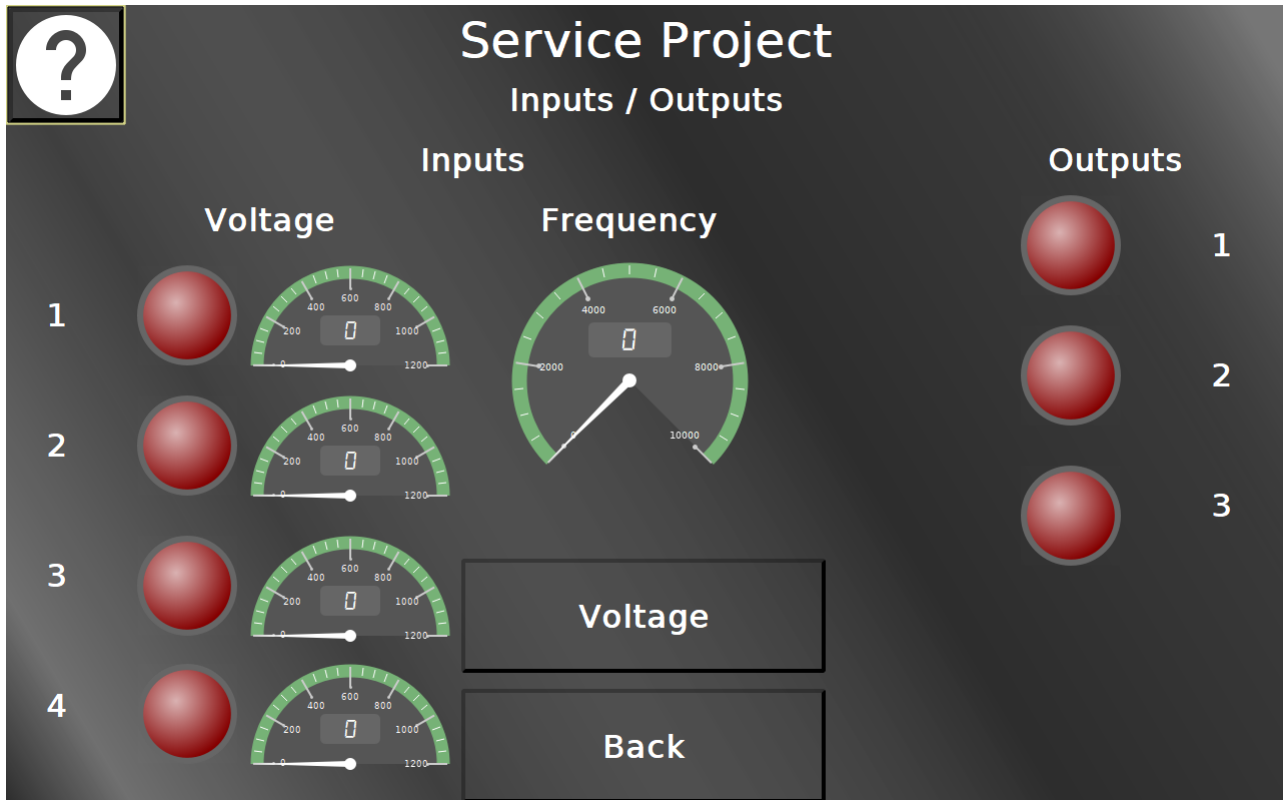
In this screen the analog / digital / current / frequency (where available) inputs can be viewed and the digital outputs can be set.

The lamps for outputs 1 to 3 are inputs and can be activated / deactivated. This activates the according output.

With the Voltage button the inputs can be switched to current inputs.
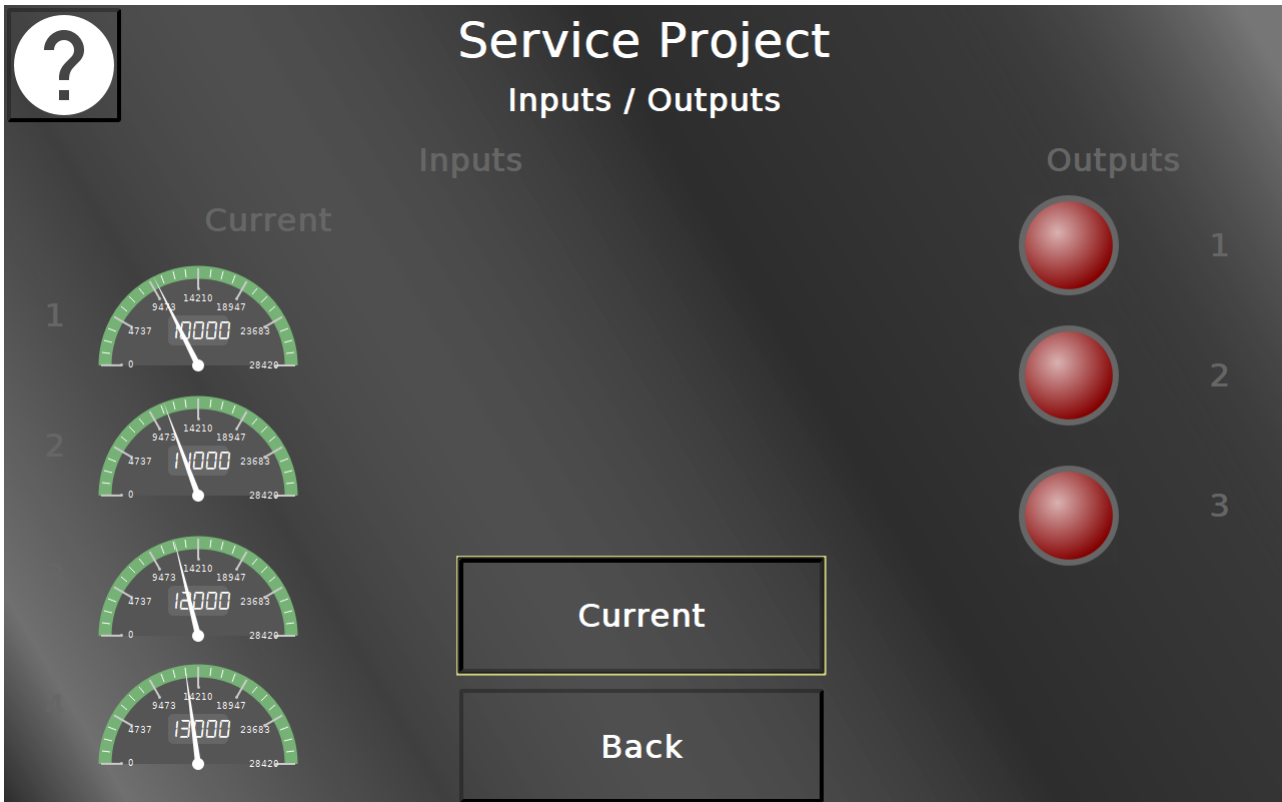
Please refer to the how-to [749] on digital outputs.

Back returns to the main menu.

Service Project -> Cameras
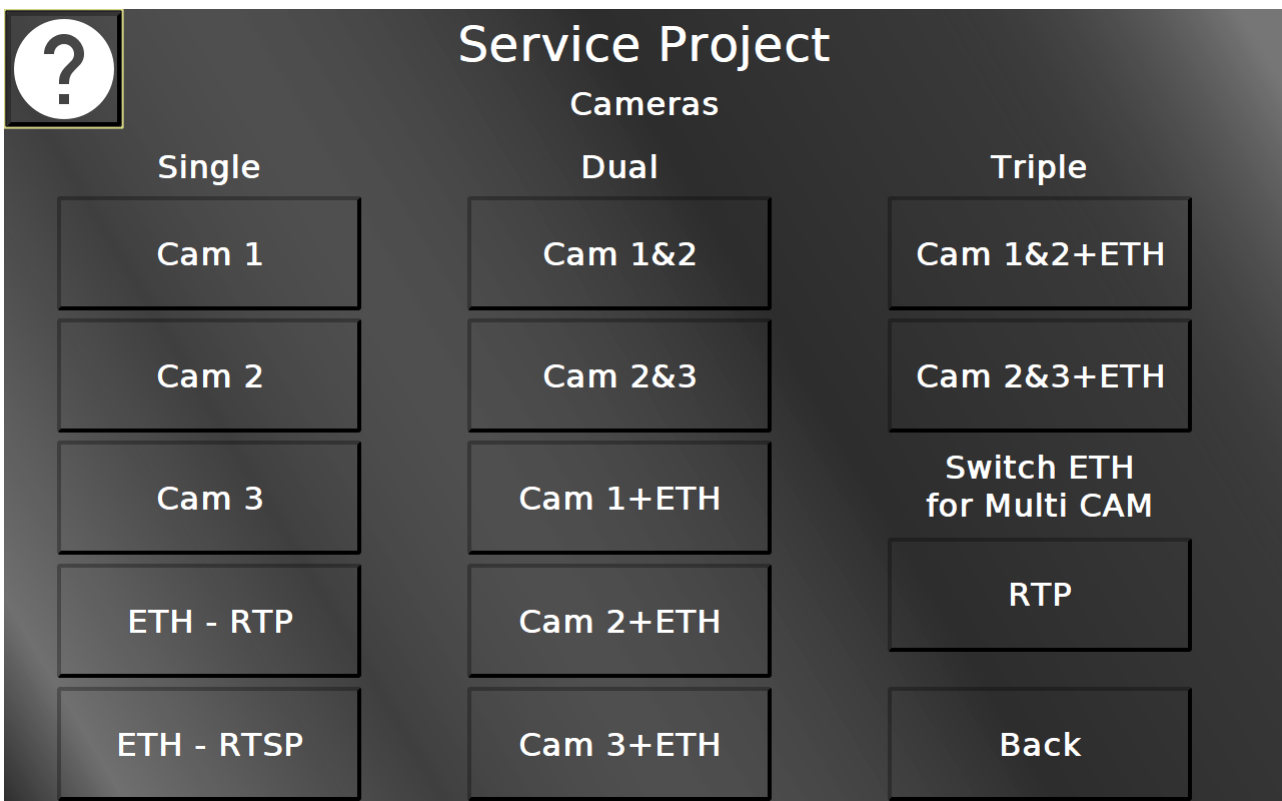
In this screen different camera screens can be chosen. Please note that only the available screens should be opened.
With the RTP button the Ethernet Camera input for the multi camera pages can be switched between RTP and RTSP.
Back returns to the main menu.

Service Project -> Cameras -> Single Camera Pages

The parameter drop down in the single camera view screens chooses between the available camera image manipulations (brightness, contrast, saturation, height, width, rotation, mirror, x-position, y-position).
The - (minus) and + (plus) buttons change the according value.
Rotate rotates the image in steps of 90 degrees. This will set a combination of rotation and mirror values.
Back returns to the camera main menu.

Service Project -> Cameras -> Ethernet RTP Camera

ETH Camera

?

UDP Port

5004

Modifier:

0x1

| - | + |

ETH settings

Back

In this screen an Ethernet camera stream can be viewed. Note that only RTP over UDP is supported in this screen.
The UDP port can be set with the - (minus) and + (plus) buttons. The Modifier dropdown chooses the modify value.
ETH settings jumps to the Ethernet settings to set an IP address.
Back returns to the camera main menu.

Service Project -> Cameras -> Ethernet RTSP Camera
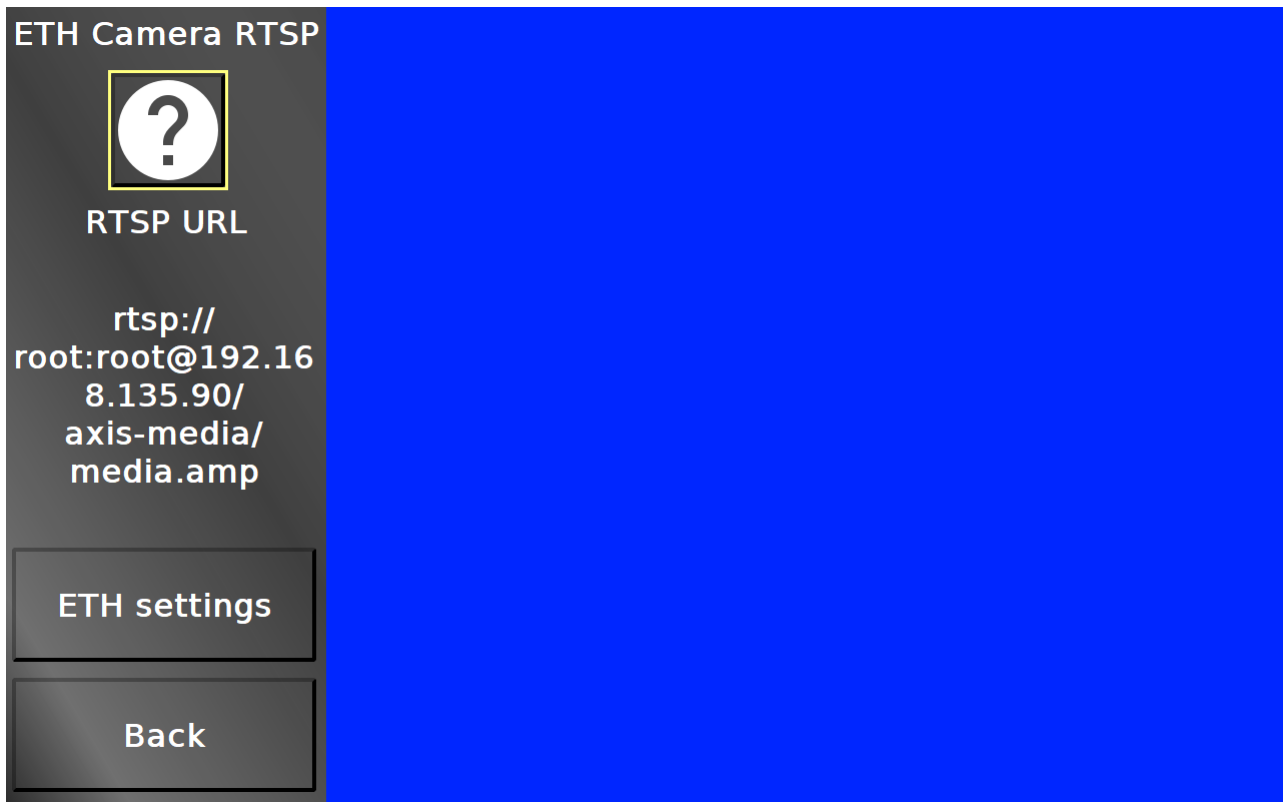
In this screen an Ethernet camera stream can be viewed. Note that only RTSP over UDP is supported in this screen.
The RTSP address can be set by selecting the String Field.
ETH settings jumps to the Ethernet settings to set an IP address.
Back returns to the camera main menu.

The other pages in the camera menu can show different multi camera settings. Note that not all combinations work with all device types. The combinations that are not possible, should be shown grayed out.

Service Project -> Clock
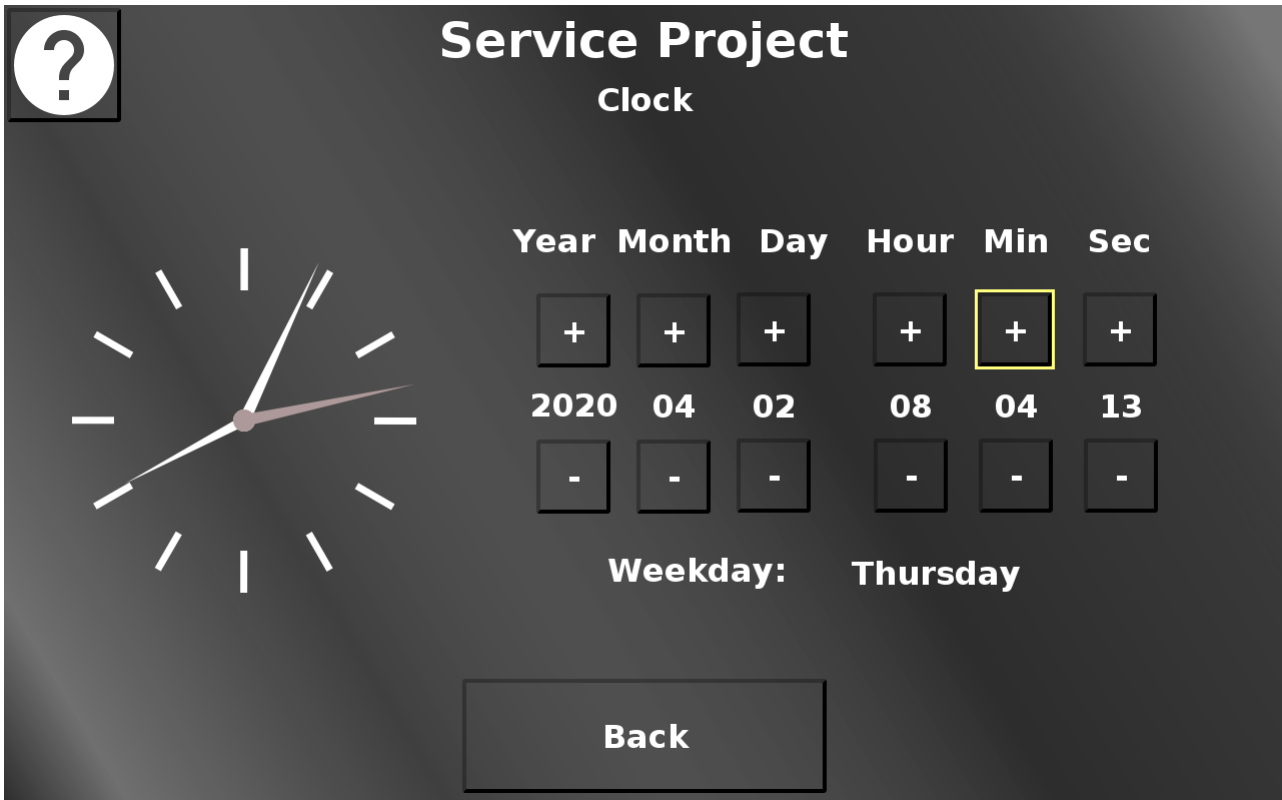
In this screen the date and time can be set. The - (minus) and + (plus) buttons change the according value.
The week day will be calculated automatically.
Back returns to the main menu.

Service Project -> Keys
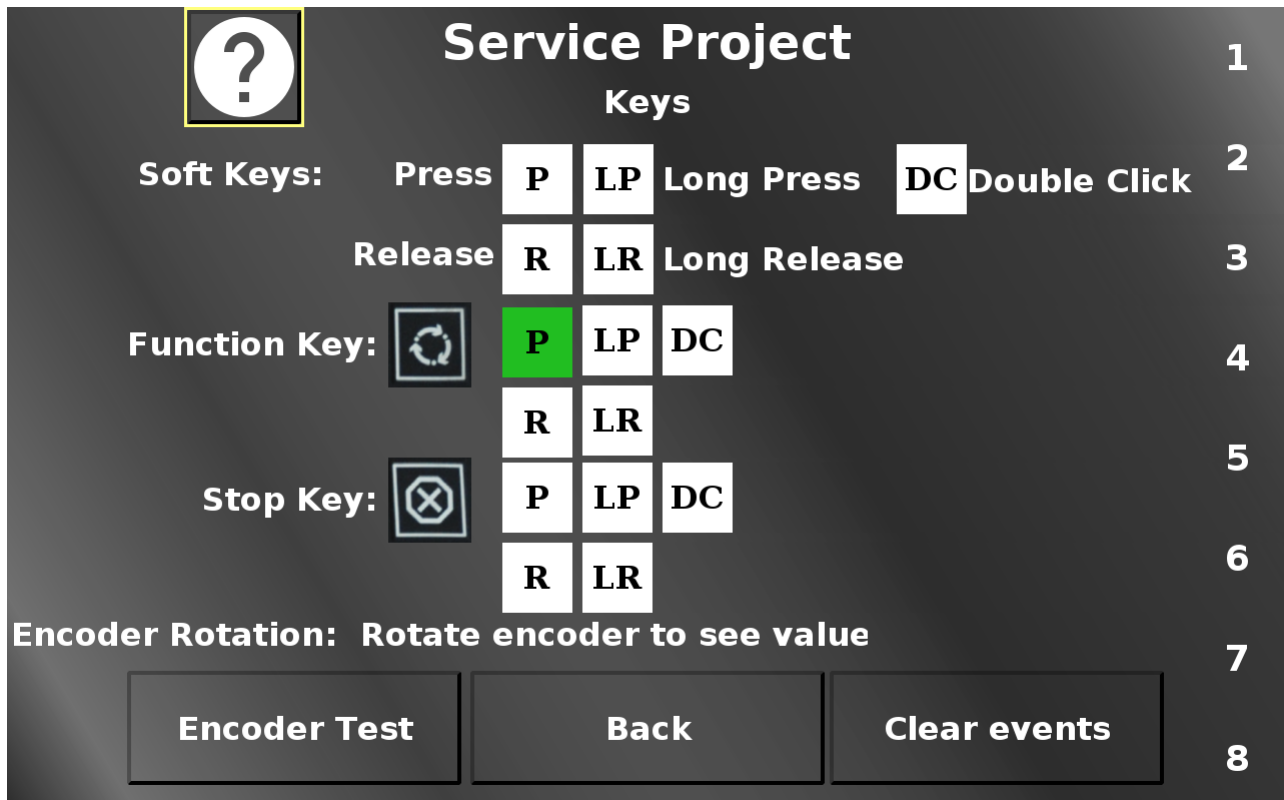
In this screen the soft- and hard keys as well as the encoder can be tested.
For the soft keys, both the actual physical key and the according touch region can be used to test.
For each event, the according field will be colored green (P - press, R - release, LP - long press, LR - long release, DC - double click).
The rotation of the Encoder is displayed in the Numeric Field. The Encoder Test button does nothing, it can be used to test the press and release events of the encoder without executing any unwanted action.
Clear events clears all event fields.
Back returns to the main menu.

Service Project -> Touch

Service Project

Touch

XPress 415   XRel 415
YPress 181   YRel 181

°Press 1   °Rel 1

> **Multi touch**
>
> **Clear**
>
> **Back**

In this screen the touch input can be tested. Each squared button will be colored green if it has been touched, so a well-working screen should be able to color all buttons, and a button should only be colored green if it actually has been touched.

Additionally, the touch coordinates for X and Y both for pressing and releasing are displayed, as well as a counter number for the presses and releases of the touch.

With the button Touch calibration the touch screen calibration can be started. 5 cross hairs will appear one after another (top left, top right, bottom right, bottom left, center). Press the cross hairs to calibrate the screen.

Please note that the PClient stays on during this operation, so the cross hairs will be painted over this screen.

Please note that the calibration button is available only for resistive touch screens. Devices with capacitive touch screens do not need a calibration.

The button Multi touch leads to the multi touch test page. Please note that the multi touch functionality is not available on OPUS A3 and OPUS A6 G1.

Back returns to the main menu.

Service Project -> Touch -> Multi Touch
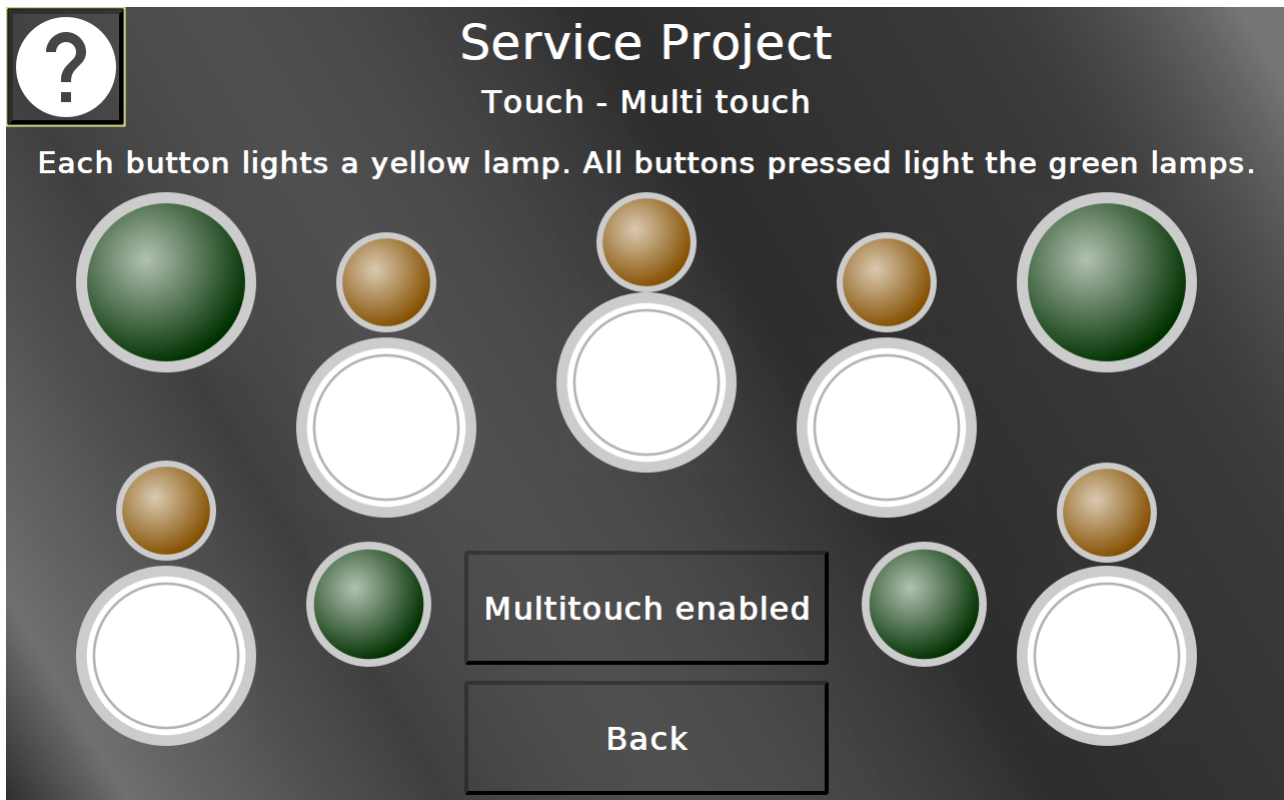
In this page the multi touch functionality can be tested. Depending on the capabilities of the hardware, 2 or 5 buttons, each with a lamp, will be displayed.
Each button lights up the according lamp. If all buttons are pressed at the same time, the green lamp lights up.
With the multi touch button the multi touch functionality can be disabled. After that the touch screen should only react upon the first finger press.
Please note that the multi touch functionality is not available on OPUS A3 and OPUS A6 G1.
Back returns to the main menu.

Service Project -> Audio / Video

Service Project
Audio / Video

Beeper

Media Player

Main Volume: -   40   +

Alarm Volume: -   100   +

Speaker Volume: -   100   +
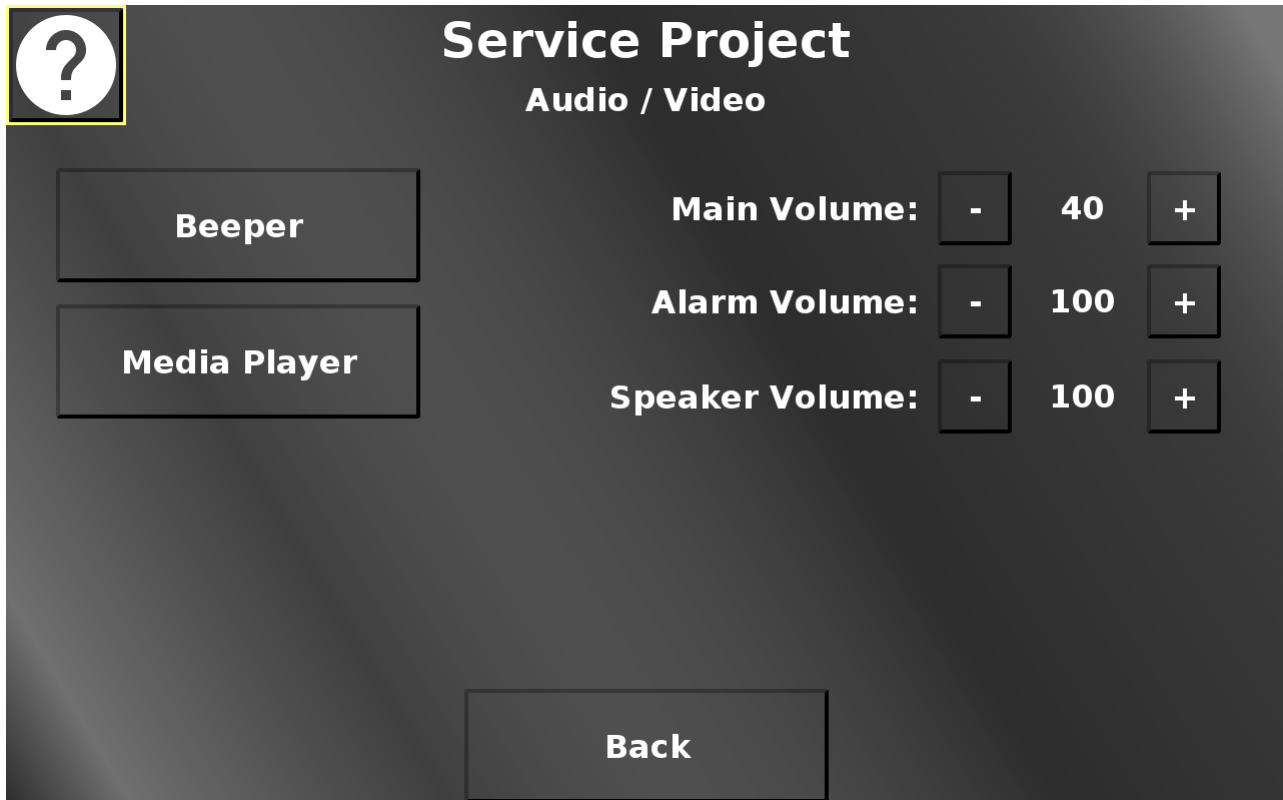
Back

In this screen some volumes for speaker / beeper can be set and different audio / video screens can be reached.
Back returns to the main menu.

Service Project -> Audio / Video -> Beeper

Service Project
Audio / Video - Beeper

Beep On Click On

Enables a beep on every touch press

Open Alarm

Opens an alarm and beeps an alarm sound. Alarm closes automatically.

Back
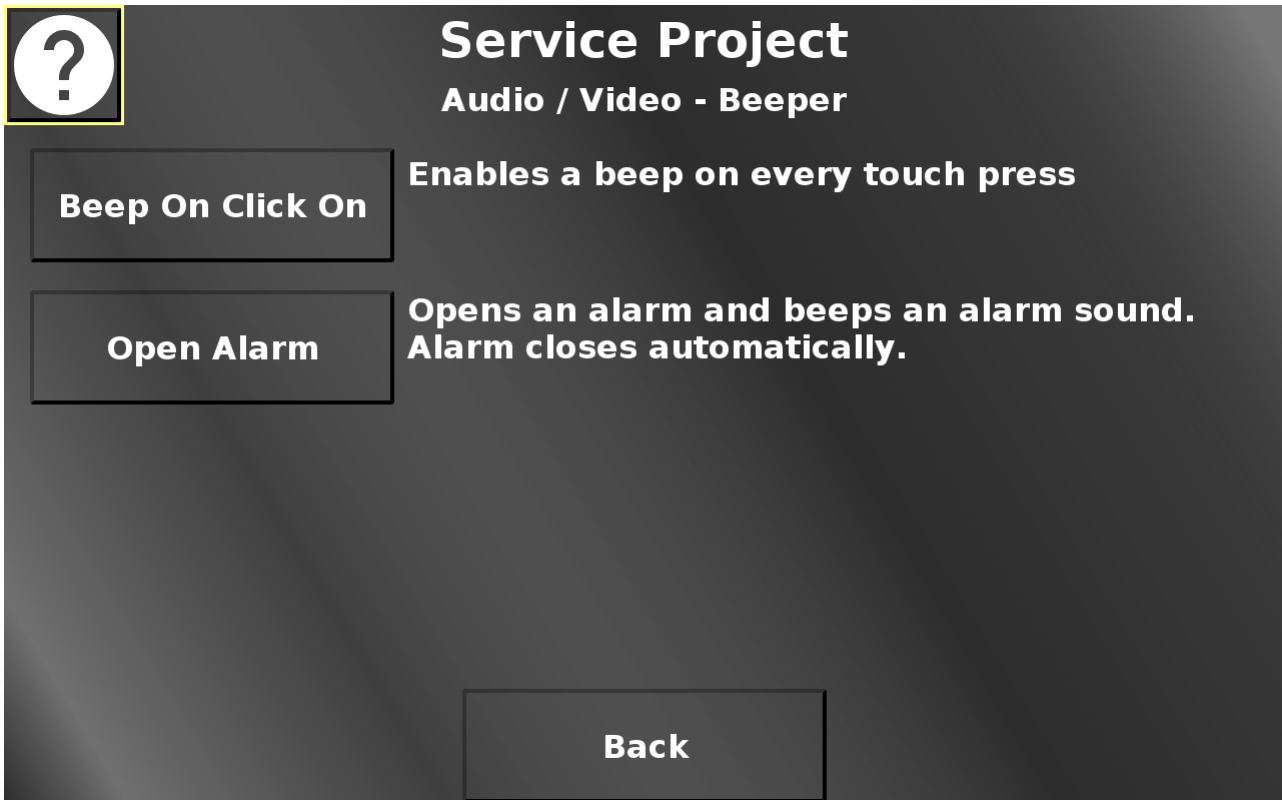
In this screen the beep on click function can be enabled / disabled. With this enabled, the device will beep whenever the touch screen has been pressed.
Additionally, an alarm can be opened which also triggers a beep. The alarm will close itself after 2 seconds.
Back returns to the Audio / Video menu.

Service Project -> Audio / Video -> Multimedia Player
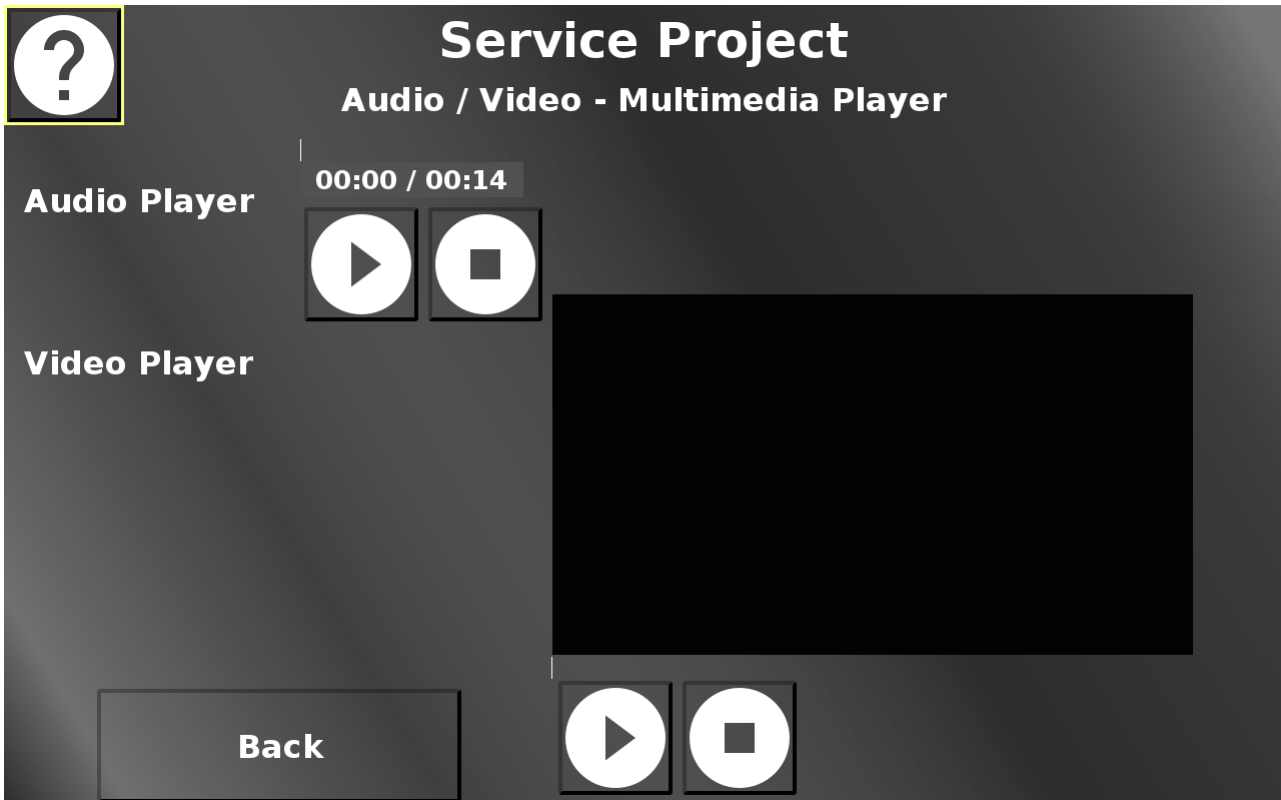
In this screen an exemplary audio and video file can be played.
Both players have a play / pause button and a stop button to control the playback.
Back returns to the Audio / Video menu.

11.3   Tutorial Videos

We have created a number of tutorial videos about the Projektor Tool and its usage. Some of them are linked within the topics of this manual.
For a complete list, please visit the Topcon Electronics Projektor Tool tutorial video page.

11.4   Crash, Issue, Feedback reports

With this version we are introducing the possibility to send crash reports and give us general feedback or report an issue directly through the program.

We want to ask you to send us reports, especially Crash Reports, but also your feedback, issues or feature requests.
We can guarantee you that your report will be looked at, carefully reviewed and treated seriously.
We are not a huge faceless company that doesn't care about its products' users.
We want to know your problems with and thoughts about our software, because we want to improve it.

The following types of reports can be sent:
- Crash reports (those are opened automatically when an exception occurs)
- Feedback (for general feedback about our program, positive or negative)
- Feature request (for ideas about new features)
- Issue/Bug (for problems that don't have to do with an exception directly)

Crash report

If a crash, or more correctly, an exception occurs, a dialog is opened automatically.

Good news first: if you want to send us that report, but you don't want to bother with all the things we ask, just click Send, the crash report will be accepted anonymously.

But it will be more helpful if you will out the fields in the report dialog.

Name - can be your personal name and/or company name (optional for crash reports). The name will be stored for future reports. You can delete the stored data in the [Options dialog](#)[65].

Email - if we have questions regarding your report, it would be good if we could reach out to you (optional for crash reports). The email address will be stored for future reports. You can delete the stored data in the <u>Options dialog</u>[65] .

Subject - is automatically filled with the exception name

Description - Anything you can tell us about what lead to the exception will be helpful (optional for crash reports)

You can attach files to the report.

Attach current project as zip-file to the report - You can choose to attach the current project to the report. Regardless of existing NDAs with your company we can assure you that we will use your project only to try to replicate the issue. We will never give customer projects to any third party. (optional)

Attach undo/redo history to help us to understand the circumstances of the error - You can choose to attach the last actions that were recorded in the undo/redo system. This can help us pinpoint what exactly happens. Additionally, some statistical information about the current project will be attached.

Such a workflow looks e.g. like this:

```
1:  ===================================
    Workflow => TS: 2021.11.24-14:30:45
    [Project] "EmptyProject_85" | Loaded | Duration: 3s
    Version                 : 2021.4.4.1
    Orig-Version             : 2021.4.4.1
    Versioning              : false
    Panel            :
    Warnings                : false
    Themes                  : [modern_bright]
    Languages               : [english, german, chinese,
russian, spanish]
    Language-/Unit Objects    : 0 / 0
    Day-/Night-Mode (Objects)      : false (0)
    Visibility (Objects)           : false (0)
    Original Images (unused)    : 10 (10)
    Unused Images           : [PowerSwitchObject_Pressed.svg,
PowerSwitchObject_Released.svg, PushSwitchObject_Pressed.svg,
PushSwitchObject_Released.svg, RockerSwitchObject_Pressed.svg,
RockerSwitchObject_Released.svg,
RockerSwitchHorizontalObject_Pressed.svg,
RockerSwitchHorizontalObject_Released.svg,
LampObject_Pressed.svg, LampObject_Released.svg]
    References              : 0
    Objects                 : 4
    --------------------------      :
      Frame                 : 1
      Page                  : 1
      Softkey Frame           : 2
    --------------------------      :
    JS-Files                : 0
    Variables               : 0
```

```
     Protocols                 : [No Protocols]
     CAN-Mappings               : 0
2:  ========================================
     DummyEdit => TS: 2021.11.24-14:30:45
     Page: Project
     Edit: Start: EmptyProject_85
3:  ========================================
     Workflow => TS: 2021.11.24-14:30:51
     [Action] "Report an issue or give feedback" |
Action_Performed
4:  ========================================
     Workflow => TS: 2021.11.24-14:30:51
     [Dialog] "Report an issue, request a feature or give
feedback" | Opened
5:  ========================================
     Workflow => TS: 2021.11.24-14:31:14
     [Dialog] "Report successful" | Opened
6:  ========================================
     Workflow => TS: 2021.11.24-14:31:16
     [Dialog] "Report successful" | Closed
7:  ========================================
     Workflow => TS: 2021.11.24-14:31:16
     [Dialog] "Report an issue, request a feature or give
feedback" | Closed |  => 'Cancelled'
8:  ========================================
     Workflow => TS: 2021.11.24-15:17:29
     [Action] "Open Project..." | Action_Performed
9:  ========================================
     Workflow => TS: 2021.11.24-15:21:28
     [TC_Page] "Home Page [35]" | Opened
10: ========================================
     Workflow => TS: 2021.11.24-15:21:29
     [TC_Page] "SPLASH SCREEN [16449]" | Opened
11: ========================================
     JSAddRemoveEdit => TS: 2021.11.25-07:44:20
     Page: JavaScripts
     Edit: Add
     Old Value:
     New Value: File: newEmpty_JavaScript.js
12: ========================================
     JSUpdateEdit => TS: 2021.11.25-07:44:21
     Page: JavaScripts
     Edit: Update
     Old Value:
     New Value: newEmpty_JavaScript.js
13: ========================================
     Workflow => TS: 2021.11.25-07:44:22
     [Dialog] "Hint" | Opened
14: ========================================
     Workflow => TS: 2021.11.25-07:44:23
```

```
   [Dialog] "Show this dialog again" | Action_Performed |  =>
'Selected: false'
15: =======================================
   Workflow => TS: 2021.11.25-07:44:24
   [Dialog] "Show this dialog again" | Action_Performed |  =>
'Selected: true'
16: =======================================
   Workflow => TS: 2021.11.25-07:44:25
   [Dialog] "Show this dialog again" | Action_Performed |  =>
'Selected: false'
17: =======================================
   Workflow => TS: 2021.11.25-07:44:25
   [Dialog] "Hint" | Closed |  => 'Cancelled'
18: =======================================
   Workflow => TS: 2021.11.25-07:51:53
   [Dialog] "Variable Manager" | Opened
19: =======================================
   Workflow => TS: 2021.11.25-07:51:59
   [Dialog] "Variable Manager" | Closed |  => 'Closed'
20: =======================================
   Workflow => TS: 2021.11.25-10:01:39
   [Action] "Help" | Action_Performed
21: =======================================
   Workflow => TS: 2021.11.25-10:01:57
   [Action] "Help" | Action_Performed
22: =======================================
   AddRemoveEdit => TS: 2021.11.29-09:04:03
   Page: Home Page
   Object(s): Button 1[45]
   Parent(s): Frame 1[36]
   Edit: Add
23: =======================================
   MoveEdit => TS: 2021.11.29-09:04:04
   Page: Home Page
   Object(s): Button 1[45]
   Edit: Move
   Old Value: POS [168, 46]
   New Value: POS [178, 19]
24: =======================================
   AddRemoveEdit => TS: 2021.11.29-09:04:09
   Page: Home Page
   Object(s): Numeric Field 1[46]
   Parent(s): Frame 1[36]
   Edit: Add
25: =======================================
   MoveEdit => TS: 2021.11.29-09:04:10
   Page: Home Page
   Object(s): Numeric Field 1[46]
   Edit: Move
   Old Value: POS [227, 125]
   New Value: POS [178, 97]
```

```
26: ====================================
   PropertyEdit => TS: 2021.11.29-09:04:21
   Page: Home Page
   Object(s): Numeric Field 1[46]
   Edit: Absolute Max Value
   Old Value: 100
   New Value: 1000
27: ====================================
   Workflow => TS: 2021.11.29-09:04:28
   [Action] "Report an issue or give feedback" |
Action_Performed
28: ====================================
   Workflow => TS: 2021.11.29-09:04:28
   [Dialog] "Report an issue, request a feature or give
feedback" | Opened
ct(s): Button 1[45]
   Edit: Move
   Old Value: POS [168, 46]
   New Value: POS [178, 19]
```

Attach system-information to help us understand the surroundings - You can choose to send us your system information. These will be all technical details of your hard- and software, as far as we can get this information. You can click the "system information" to see this information before sending it to us. The information looks e.g. like this:

OS info:
Name: Windows 10
Version: 10.0
Architecture amd64
Available processors (cores): 4

Memory info:
Free heap memory: 1.040.218
Allocated heap memory: 2.656.256
Max heap memory: 4.194.304
Total free heap memory: 2.578.266
Total physical memory: 8.006.560

Disk info:
File system root: C:\
Total space (bytes): 252974198784
Free space (bytes): 31004663808
Usable space (bytes): 31004663808

Attach the latest error-information - For the crash report this will be the most important option, which is why it cannot be de-selected. It is a stack trace of the internal exception. This will be the most valuable information to try to understand what just happened, and how we can prevent it from happening again. A stack trace looks e.g. like this:

Lastly occurred error [2020-02-20 13:02:03.1671769 (+01:00 Europe/Berlin)]:

java.util.ConcurrentModificationException
    at java.base/java.util.HashMap$HashIterator.nextNode(Unknown Source)
    at java.base/java.util.HashMap$KeyIterator.next(Unknown Source)
    at
org.netbeans.api.visual.model.ObjectScene.setSelectedObjects(ObjectScene.java:217)
    at
org..core.gui.sceneDesigner.PageTopComponent$3.resultChanged(PageTopComponent.java:328)
    at
org.openide.util.lookup.SimpleProxyLookup.checkLookup(SimpleProxyLookup.java:89)
    at org.openide.util.lookup.SimpleProxyLookup.lookup(SimpleProxyLookup.java:134)
    at
org.netbeans.modules.openide.windows.GlobalActionContextImpl.blickActionMapImpl(GlobalActionContextImpl.java:107)
    at
org.netbeans.modules.openide.windows.GlobalActionContextImpl.blickActionMap(GlobalActionContextImpl.java:81)
    at
org.netbeans.modules.openide.windows.GlobalActionContextImpl.propertyChange(GlobalActionContextImpl.java:180)
    at java.desktop/java.beans.PropertyChangeSupport.fire(Unknown Source)
    at java.desktop/java.beans.PropertyChangeSupport.firePropertyChange(Unknown Source)
    at java.desktop/java.beans.PropertyChangeSupport.firePropertyChange(Unknown Source)
    at java.desktop/java.awt.KeyboardFocusManager.firePropertyChange(Unknown Source)
    at
java.desktop/java.awt.KeyboardFocusManager.setGlobalPermanentFocusOwner(Unknown Source)
    at java.desktop/java.awt.DefaultKeyboardFocusManager.dispatchEvent(Unknown Source)
    at java.desktop/java.awt.Component.dispatchEventImpl(Unknown Source)
    at java.desktop/java.awt.Container.dispatchEventImpl(Unknown Source)
    at java.desktop/java.awt.Component.dispatchEvent(Unknown Source)
    at java.desktop/sun.awt.SunToolkit$1.run(Unknown Source)
    at java.desktop/java.awt.event.InvocationEvent.dispatch(Unknown Source)
    at java.desktop/java.awt.EventQueue.dispatchEventImpl(Unknown Source)
    at java.desktop/java.awt.EventQueue$4.run(Unknown Source)
    at java.desktop/java.awt.EventQueue$4.run(Unknown Source)
    at java.base/java.security.AccessController.doPrivileged(Native Method)
    at
java.base/java.security.ProtectionDomain$JavaSecurityAccessImpl.doIntersectionPrivilege(Unknown Source)
    at java.desktop/java.awt.EventQueue.dispatchEvent(Unknown Source)
    at org.netbeans.core.TimableEventQueue.dispatchEvent(TimableEventQueue.java:136)
    at java.desktop/java.awt.EventDispatchThread.pumpOneEventForFilters(Unknown Source)
    at java.desktop/java.awt.EventDispatchThread.pumpEventsForFilter(Unknown Source)
    at java.desktop/java.awt.EventDispatchThread.pumpEventsForHierarchy(Unknown Source)

at java.desktop/java.awt.EventDispatchThread.pumpEvents(Unknown Source)
at java.desktop/java.awt.EventDispatchThread.pumpEvents(Unknown Source)
at java.desktop/java.awt.EventDispatchThread.run(Unknown Source)

Don't show this error again - You can choose to not send this error again. If there happens to be an exception that occurs rather often in your work flow, it will be annoying for you to have the dialog pop up all the time, and receiving the same report over and over again will not be increasingly helpful for us, either. For that case you can disable to show the error that just occurred, again. You can reset this setting in the Options [65]dialog.

Don't show any errors - Please don't enable that. Normally it will not be helpful if you check that option. It will not be of direct harm to you, but if you don't send error reports, we have no chance of recognizing the underlying problem and no chance to repair it. You can reset this setting in the Options [65]dialog.

Send - This will send the report, close the dialog, and, when the transmission is finished, show a feedback dialog with the ticket number your report has created.

Report successful   ✕

ⓘ   Your request was sent to Topcon Electronics GmbH & Co. KG .
    Please use the following issue ID as reference for further communication: PRJEXT-181

                          OK
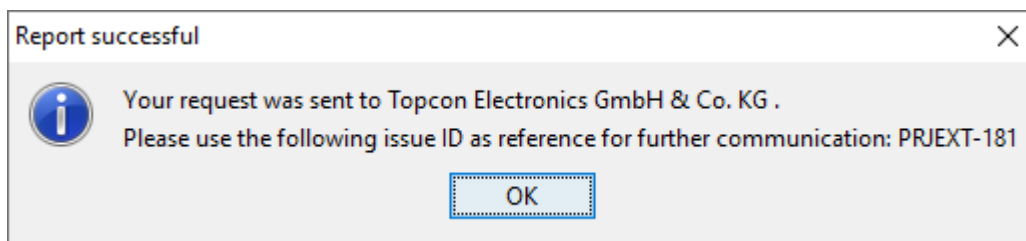
You can use this ID as a reference to contact us with more information, questions, etc.

Don't send - This will close the report dialog without sending the report. We hope that you will rarely choose that option. Once again, we are neither Microsoft nor Apple, we want to work with you on creating the best program possible and your report will not end up unread in the trash bin.

Feedback / Feature Request

The feedback dialog can be opened through the menu Help -> Report an issue or give feedback.

Choose Feedback in the Your concern drop down list to send a general feedback report. Choose Feature Request in the Your concern drop down list to send a feature request report.

The dialog for feedback and feature request are both very similar to the crash report with the following differences:

Name, Email, Subject - These fields are necessary fields now because the probability that we will want to discuss your report with you are much higher. The name and email will be stored for future reports.

Report Issue / Bug

The Report Issue / Bug dialog can be opened through the menu Help -> Report an issue or give feedback.

Choose Issue/Bug in the Your concern drop down list to send an issue / bug report.

The dialog for Issue/Bug is similar to the crash report, with the following differences:

Undo/redo history is not available.
Name, Email, Subject - These fields are necessary fields now because the probability that we will want to discuss your report with you are much higher.

Reported Issues View
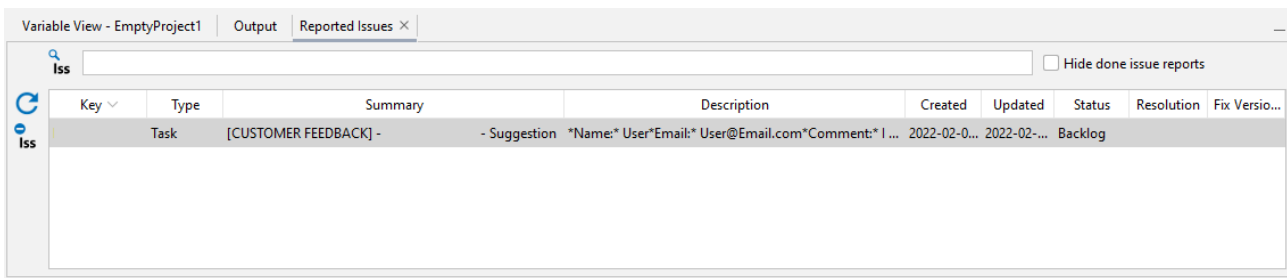
In this view all crash reports and feedbacks from this installment of the program are listed. The list can be sorted by either of the columns.
A double click opens the Reported Issue Details Window.
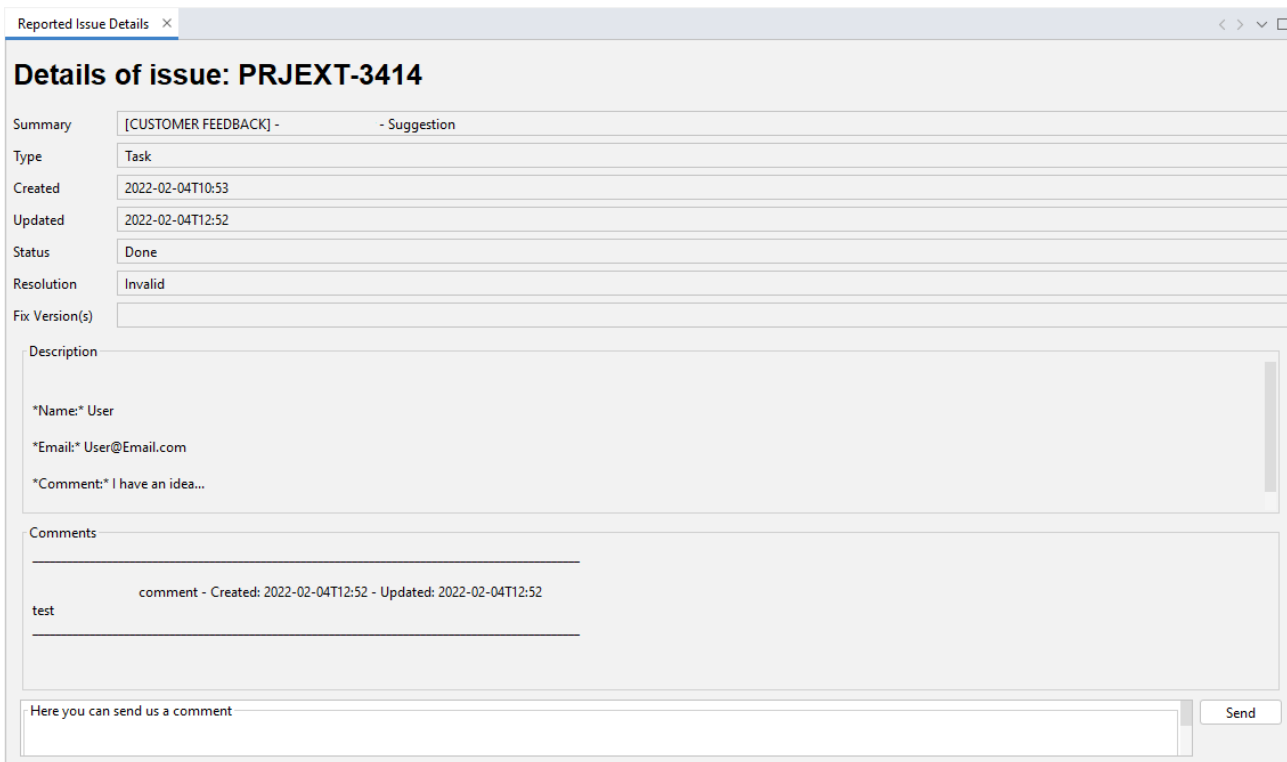
In this window the details about the issue can be viewed, including the current status and all comments, both by the user and by us.
A new comment can be added in this window.
If anything has changed in one of the issues, this symbol will be displayed in the lower right:

When the user clicks on the symbol, a bubble with information about the update is displayed. The blue underlined text can be clicked and opens the Reported Issue Detail window with this issue and the new information.
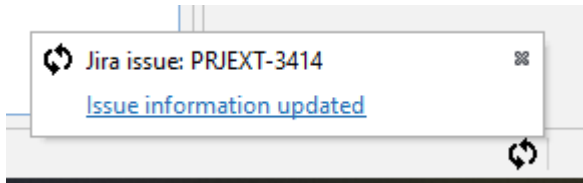
FAQ

Why shouldn't I just click "Don't send" mindlessly if a report pops up? Why should I care?
If you use the program regularly, you surely want the best experience possible when using it. We know that the program is not perfect (yet), but we want to make it as good as we can, and you can help make it a little better with your error reports and your feedback and ideas.

Will I get a feedback from you if I send reports?
Yes. With the new Issue View you have an overview of all your reports and the measures we take.

Will you do everything I suggest?
Probably not, except if you only have so very excellent ideas that we just need to implement them all. We will consider every suggestion though, and if we deem it a good idea for lots of customers, you may see your suggestion become reality in one of the next versions.

What will you do with my data and my project?
We will not use your email address for anything other than contacting you in case it is necessary to solve the issue at hand. We will only use your project to replicate the problem that you have. Neither will ever go to a third party.

How is my report sent to you?
We use a JIRA server to process reports. Every report creates a ticket in JIRA, which will then be looked at and processed by us. The ticket number will be displayed in a dialog after you have sent the report. You also have an overview of all your reports in the Issue View.

The undo/redo history looks like you are recording all my activities - are you spying on me?
No. Yes, we are recording your activities in the current session, but only for the purpose of being able to understand what you did before an exception occurred. You can see the complete history that you send to us beforehand. There is no sensitive data that could reveal any kind of intellectual property or sensitive information.

My report wasn't sent - why?
We see the following possible reasons why a report couldn't be sent:
- You don't have an Internet connection - nothing we can do about that, but when you reconnect you can send your reports again, see below.
- Our server is down - nothing you can do about that. Hopefully it will be back up quite soon. If in doubt, contact the technical support to find out what happened
- Computer says no - I hope you laughed now. This is a little easter egg, and by the way, thank you for reading our manual so deeply! Let's stay serious now and continue.

- The report is too big - There may only be short of 30.000 characters in a report. If you wrote a LOT in the description, maybe shorten it a bit and send the rest as a comment through the Issue View. If it happens that the exception stack trace is too large, please send it as a comment later instead.

My report wasn't sent because I don't have Internet connection. Is it lost now?
No, we buffer unsent reports until our JIRA server can be reached again. They will be sent the next time a report is successfully sent. We buffer up to 10 MB of report data (excluding attachments and project data).

How To

12   How To

Here you will find some quick short solutions for common use cases. Feel free to contact us if something you think should be there is missing.

12.1   How to create switch buttons with changing colors

You want to create buttons that activate and deactivate certain settings (like a switch). The background color of the button should indicate if the setting is activated or not.

- Create a button object
- Set the property Latching to true
- If applicable, put a String Field or a Picture Graphic object into the button (set them to transparent to still see the button background)
- Put a script with the following code into the OnPress event of the button:
  ```
  setProperty(<button_ID>, "Background Color", <"pressed" color
  in [r,g,b,a] array format>);
  ```
- Put a script with the following code into the OnRelease event of the button:
  ```
  setProperty(<button_ID>, "Background Color", <"released" color
  in [r,g,b,a] array format>);
  ```

If the button background color shouldn't change directly when the button is pressed but instead after a confirmation message from the CAN bus, put the script on the OnValueChangeByOwner event of the related variable.

12.2   How to use one script for several buttons

You want to have a set of buttons that should basically perform a similar action, but a bit different depending on which of the buttons has been pressed.

- Create some buttons
- Create a variable "button_id"
- On the OnPress event of each button, set the variable button_id to a certain value (e.g. on button 1 to 1, on button 2 to 2 and so on)
- Create one JavaScript
- Put the script file on the OnRelease event of each button
- In the script use a switch-case statement to differentiate what should happen with which button

New since version 2019 of the program:
We have a function called <u>GetCallerID</u>[552] now which tells you in a script from which object the script was executed. So alternatively to the solution above this function could be used in a generic script to find out which button was being pressed.

12.3   How to create a drop down menu

You want to create a drop down menu where the user can select one of several options.

- Create a list object
- Reference a variable menu_id with the list object
- Put e.g. 10 String Fields (or images) into the list object (every object is one option in the menu)

- Activate the Set as Input property of the list object
- Set the Items Visible (Editing Mode) to e.g. 5
- Put a JavaScript to the event OnEnterValue of the list object
- In the script, read the variable menu_id to find out which option in the menu was selected and perform the according action (e.g. jump to a certain page)

12.4   How to use the automatic backlight dimming

You want the device to change the backlight intensity automatically depending on the ambient light.

- Set the variable @DispBacklightIntensity to a value outside of its normal value region 0 to 100, e.g. -1 or 101
- To disable the automatic backlight mode, set the variable @DispBacklightIntensity back to a value between 0 and 100
- The ambient brightness is readable in the variable @DispBacklightAmbient-LowPassed
- The brightness at different ambient brightness levels can be set with the variables @DispBacklightIntensity0 to @DispBacklightIntensity100. You set e.g. @DispBacklightIntensity0 to 20, then the display brightness will be set to 20 when @DispBacklightAmbient-LowPassed has a value of 0 (%). Set @DispBacklightIntensity10 to 30, then the display brightness will be set to 30 when @DispBacklightAmbient-LowPassed has a value of 10 (%). Values in-between will be interpolated linearly
- The variable @DispBacklightAmbient-LowPassed will be calculated from @SensorAmbientLight using @DispBacklightFilterFactor. The formula is:

$$\frac{(@DispBacklightAmbientLowPassed * @DispBacklightFilterFactor) + (@SensorAmbientLight * (1000 - @DispBacklightFilterFactor))}{1000}$$

In words, the lower @DispBacklightFilterFactor is, the faster changing ambient brightness values will influence the display brightness.
This formula will be calculated every @DispBacklightTimeConstant milliseconds.

12.5   How to set up SDO communication

You want to
a) request a value from an ECU / sensor
b) read a value with the ECU from the display device
c) write a value with the ECU to the display device

- include the CANopen protocol[328] for the CAN port you want to use[307]
- set up IDs for PClient SDO Server Channel and PClient SDO Client Channel in the Terminal and Owner ECU Configuration[328] dialog
- so make it clear:
  COB-ID Client ->Server (rx) is the ID that the ECU uses to read / write a value from / to the display device (used for b) and c))
  COB-ID Server->Client (tx) is the ID that the display uses to respond a read / write request from the ECU (used for b) and c))

  COB-ID Client ->Server (tx) is the ID that the display device uses to read a value from the ECU (used for a))
  COB-ID Server->Client (rx) is the ID that the ECU uses to respond a read request from the display device (used for a))

a)
- create the variable you want to read from the ECU (data type must match!)
- open the [Variable Manager](#)¹⁶⁵
- set the ECU as the owner of the variable
- set the ECU index and ECU sub index of the variable according to the index / sub index the variable has in the ECU
- use the JavaScript function [requestVariableValueFromOwner](#)⁵⁴⁸ to request the value of the variable from the ECU

b)
- create the variable that the ECU wants to read (here the internal index / sub index are used)
- the ECU needs to send a read request message for the variable

c)
- create the variable that the ECU wants to read (here the internal index / sub index are used)
- the ECU needs to send a write request message for the variable

12.6 How to remember variable values when the device is powered off

You want to remember variable values when the device is shut down and use them when it is started again.

The most important setting is the property **Remanent** in the [Variable Manager](#)¹⁶⁵ .

The property can only be activated for user-defined variables with the PClient as the [owner](#)¹⁵⁴ .
For pre-defined variables or variables with an owner other than the PClient this setting is not possible.

Remanent variable values will be saved on the file system automatically, whenever their value changes.
Removing power completely (disconnect clamp 15) directly after a value change might lead to the loss of the last change.

Please note: When a project update with the same name is downloaded to the device, the remanent variable values will be kept.
If a project with a different name is installed on the device, the remanent variable values from the old project are not used. However, they will still be stored on the device, so if the old project (i.e. a project with the old name) is installed, the remanent variable values are used.

12.7 How to use the power management

You want to be able to use the power management in your system to save power and / or shut down the device safely.

All devices have an integrated power management.
There are 4 power modes, **On**, **Low Power**, **Sleep** and **Off**.
When the device starts up, it starts in the **On** mode.
The **Low Power** mode is reached by disconnecting clamp 15 (ignition pin).
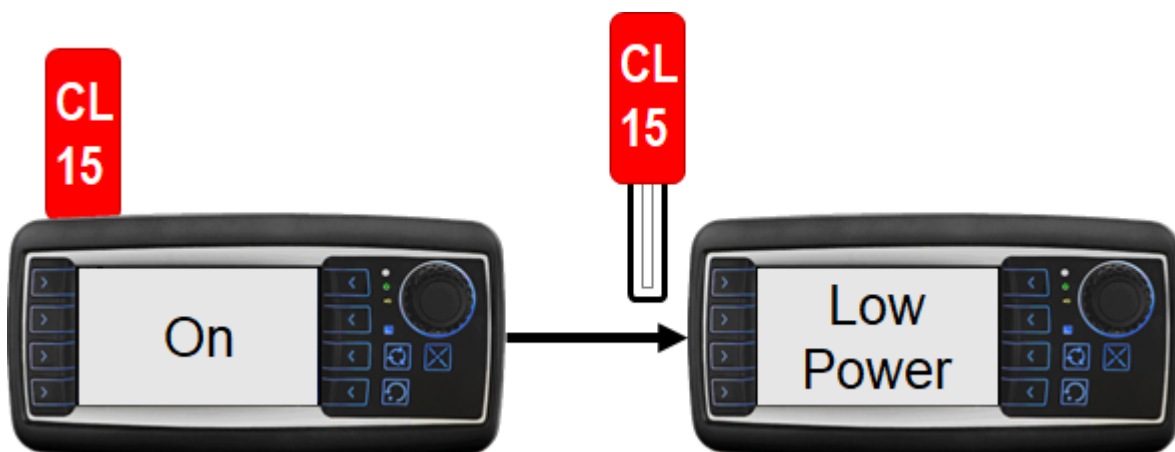
In **Low Power** mode the device is basically functional, all scripts, CAN messages etc. are still running.

Some elements of the devices can be turned off to save power and / or give the user the illusion that the device is already off:

- Display function (@PWR_LowPowerDisplayState)
- Display backlight (@PWR_LowPowerDisplayBacklightState)
- Key backlight (@PWR_LowPowerKeypadBacklightState)
- Key function (@PWR_LowPowerKeypadState)
- Encoder function (@PWR_LowPowerEncoderState)
- Digital outputs (@PWR_LowPowerOutputXState)
- Multicolor LED (@PWR_LowPower_RGB_LED_State)
- Beeper / Speaker (@PWR_LowPowerBeeperState)
- Touchscreen (@PWR_LowPowerTouchState)
- System console (@PWR_LowPowerConsoleState)
- Video processing (@PWR_LowPowerVideoState)
- Camera output voltage (@PWR_LowPowerCamOutputState)

By default, all but the console are turned off.

After a time that can be set with @PWR_TimeToSleepMode (anywhere from 0 to 2332800 seconds (27 days), time will be rounded up to multiples of 60s for EGScore devices) the device goes into **Sleep** mode.

In **Sleep** mode the device is basically not running anymore, no CAN, no scripts, nothing.

After another time, set with @PWR_TimeToPowerOff (again between 0 and 2332800 seconds), the device turns **Off**.
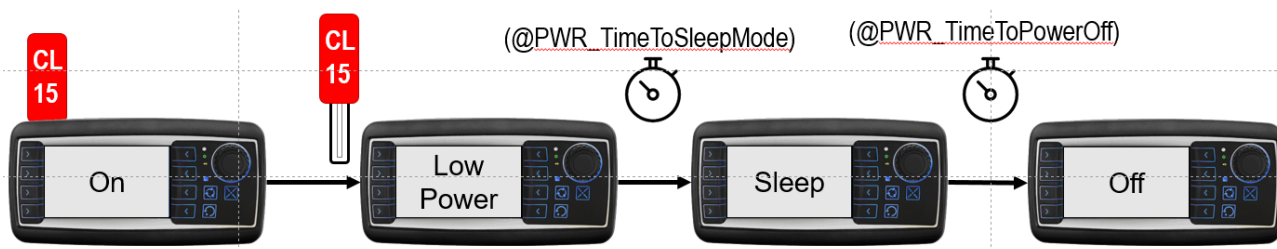
Here is a rough overview how much power the devices draw in which mode.

	4.7" iMX.35	7" iMX.25	7" iMX.6	12" iMX.6
On	430	900	1000	1600
Low Power	160	200	200	300
Sleep	≤ 90	≤ 100	≤ 100	≤ 200
Off	< 3	< 3	< 3	< 3

Only in **Sleep** or **Off** mode it's safe to remove clamp 30 (power plus). Disconnecting clamp 30 in **On** or **Low Power** is not good and can damage data.
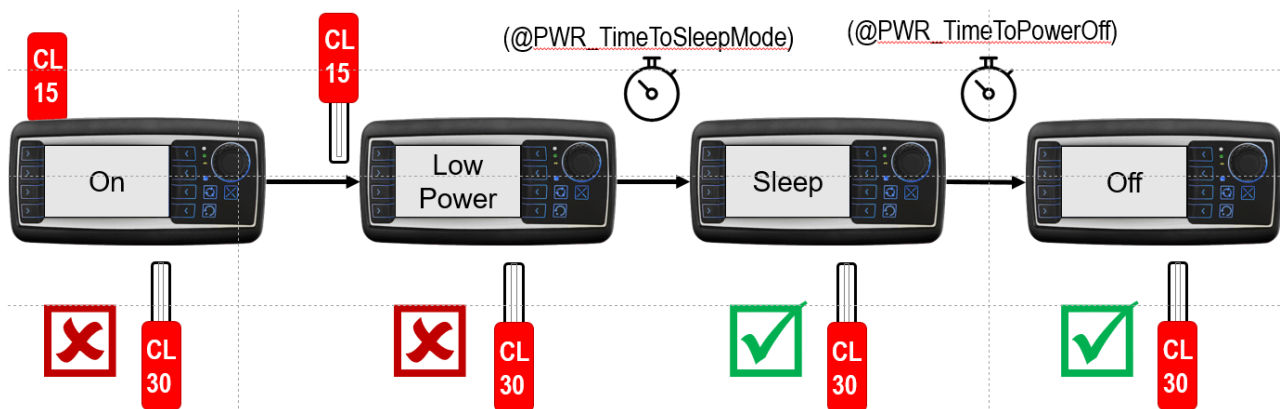
When returning to **On** mode, the starting time is highly reduced when starting from **Low Power** or **Sleep** mode.
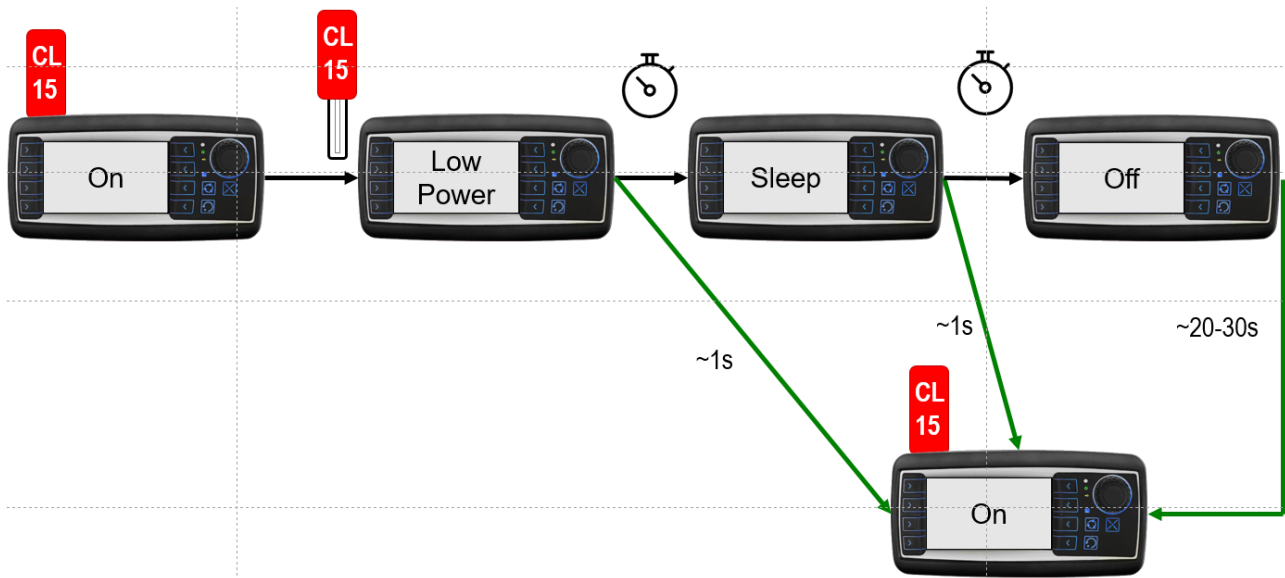
When returning from **Low Power** or **Sleep** mode, the device will continue where it was instead of starting freshly.

If disconnecting clamp 15 is not possible, consider using the variable @PWR_CurrentPowerMode.
You can shut down by setting it to 0x4A17.

12.8   How to use the digital outputs

The digital outputs that are available on some of the variants of our displays are open-drain outputs.
This means that they are not actively pushing any voltage.
Instead, they connect an external power source to ground, closing the circuit and thus enabling the external element.
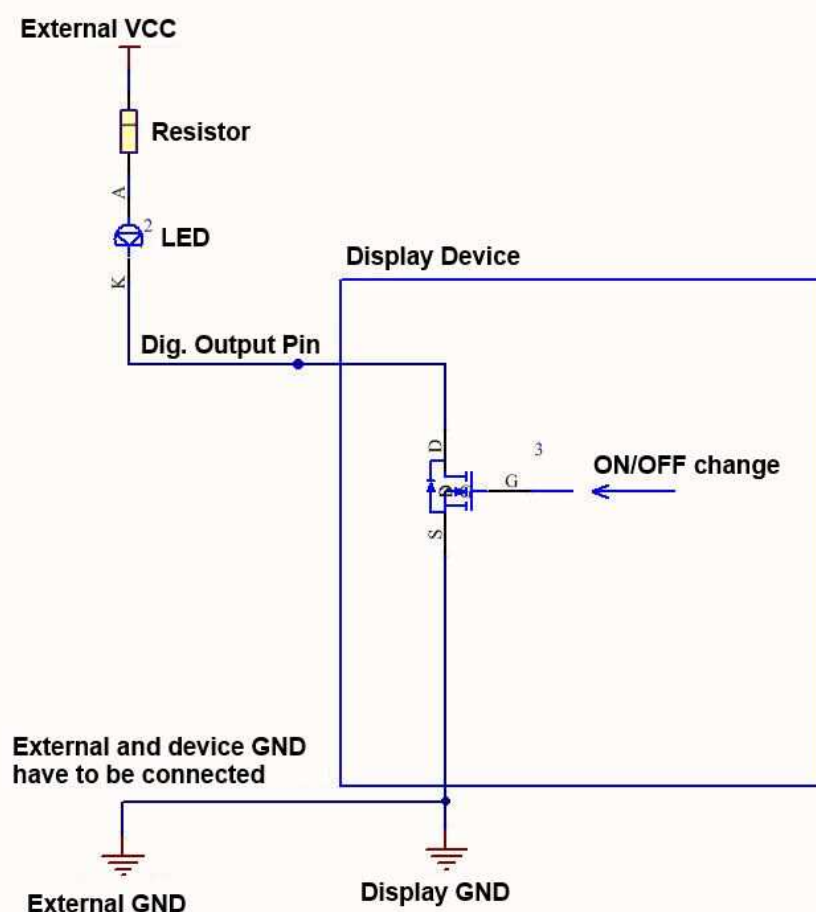
The external VCC and the power source of the display device can be the same.
The grounds have to be connected (i.e. have to be the same).
The LED in the image is only an example, it is not limited to LEDs.

12.9 How to process CAN messages with a multiplexer bit

You need to process CAN messages from an ECU that uses one CAN-ID and a multiplexer bit to differentiate the data.
The best way to do that is to create the "same" CAN mapping several times, i.e. with the same CAN-ID, length etc.
In the Visual CAN Mapping, put a constant into the multiplexer byte and set this constant to the first relevant value, e.g. 1.
The other bytes you fill with variables that should contain the data from the CAN message with multiplexer byte set to 1.
The next mapping with identical settings you put a constant into the multiplexer byte and set this to 2, and you put in the variables that should receive those data, and so on.

12.10 How to create dynamic icons

You want to put 40 icons in your UI, but you only have space for 10, so you want to display them dynamically.

- Create a list object
- Put a String Field in as the first child, delete its text and make it transparent. This will be shown when no icon needs to be displayed.
- Put all images into the list object
- Create 9 shallow copies of the list object. Now you have the 10 icon "spaces". Put them where they should be in the UI.
- Create 10 variables and reference one of them to each list object. The variables will switch the shown icon in each list object.
- Create another variable, "icon_event_flag". This variable will be used as a trigger to refresh the icons.
- Create a script for the event OnProjektInit. Fill it with this code:

```
//Definition of the variable names for the listen objects
icon_variable_names = ["New Variable_1","New Variable_2","New Variable_3","New
Variable_4",<until 10>];

num_icons = 40;//global variable with number of icons
num_lists = 4;//global variable with number of list objects
icon_enable = [];//global array to switch each icon visible

//initialize all icons with 0
for (var i = 0; i < num_icons; i++)
{
    icon_enable[i] = 0;
}
```

- Create a script show_icons.js with this code:

```
//always start with the first list object
var list_position = 1;

//set all list objects transparent first
for (var i = 0; i < num_lists; i++)
{
    setVariableValue(icon_variable_names[i],0);
}

//go through icon list and show icon if necessary
for (var i = 0; i < num_icons; i++) {//go through all icons
    if (icon_enable[i] !== 0)//check if icon i should be shown
    {
        setVariableValue(icon_variable_names[list_position-1],(i+1));//show icon i at
position "list_position"
        list_position = list_position + 1;//increase list_position to next value
    }
}
```

- Put this script to the event OnValueChange of the variable "icon_event_flag".
- For each event that should show a certain icon, put this code in a script:

```
icon_enable[<n>] = 1;//show Icon n

//change flag
var flag = getVariableValue("icon_event_flag");
flag = flag + 1;
setVariableValue("icon_event_flag", flag);
```

- For each event that should remove a certain icon, put this code in a script:

```
icon_enable[<n>] = 0;//hide Icon n

//change flag
var flag = getVariableValue("icon_event_flag");
flag = flag + 1;
setVariableValue("icon_event_flag", flag);
```

- To test this without actual events, you can put latching buttons on the screen with the above 2 scripts on OnPress and OnRelease, respectively. Each button should shows/hides one icon, so you need to vary <n> in each script.

Please note:

There are no safety procedures in place here. If more than 10 events can occur at once, this needs to be considered, e.g. by discarding the oldest entries.
The icons will always be sorted in the order that they are put in the list. If there should be a FIFO or LIFO or any kind of other order, a second array with the position within the list objects would be needed.

FAQ

13   FAQ

Can I rearrange the pages in the tree?

You can change the order of pages in the project tree by selecting a page and pressing the keyboard shortcut <ALT> + <Page Up> / <Page Down>. You can bring them to the top / bottom with <ALT> + <Pos1> / <ALT> + <End>.
Note that this order has no meaning for the project on the device.

Can I load projects from older versions?

With this version you can load projects as old as from version 1.6.x (or IDE version 3.6.x). Any older version should be opened with version 1.12.1 first.

I have a terminal with CoDeSys and would like to work with the PClient now. What should I do on the terminal side?

Just install the PClient with the normal update procedure. Codesys will be removed with this. If you want to work with Codesys again, you can re-install it.

Is it possible to shorten the startup time of the terminal?

The starting time of the terminal is caused by the 32 bit Linux operating system involved that is similar to a normal PC. Therefore it is not possible to shorten it directly. But it is definitely possible to make the process more suitable using the power management[746] . This contains two variables:
@PWR_TimeToSleepMode – defines the time in seconds after which the terminal should go into sleep mode (maximum is 2332800 s - 27 days);
@PWR_TimeToPowerOff – defines the time after which the terminal should switch off completely (maximum is 2332800 s - 27 days).

How can I adjust Power Management settings?

The power management[746] settings can only be configured using variables from the Power Modes group (Index 0x2030). Using these variables, the timeouts can be configured (when going into sleep mode and when switching off completely) and what components shall be switched on/off when entering low power mode.
Please refer to the power management variables[174] for more information.
PLEASE NOTE: The configuration file /opt/etc/pm_state.conf is not considered anymore. Only the settings made with the variables will apply.

How can I execute actions when entering different Power Modes?

Power Management variables are available, please refer to the pre-defined variable list[174] .
PLEASE NOTE: The maximum value for a timeout setting is 27 days.

The devices support four power modes:
• Off: The device is completely off, no power is consumed
• On: The device is fully running
• LowPower: Ignition pin (clamp 15) was pulled. The core program is running, but the device switches off configured modules to consume less power.

• Sleep: When the device is in LowPower mode and a timeout expires (@PWR_Time_To_Sleep_Mode), the device enters sleep mode. In this mode only the RAM is supplied with power. Power consumption is very low.

Three events are defined for the project to which actions can be assigned: OnPowerOn, OnLowPower and OnSleep. Actions can be assigned to these events by clicking on the project name in the project tree and then select "Events" in the property pane.

The OnPowerOn Event is executed as soon as the device wakes up from sleep mode or from LowPower mode (Ignition is plugged in again), but NOT when starting with a cold boot.
The OnLowPower event is executed as soon as the ignition is pulled.
The OnSleep event currently gets executed as soon as the device wakes up from sleep mode (before the OnPowerOn event). This may change in future so that the event is executed as soon as sleep mode is entered.

To each of these events one of the following actions can be assigned:
• Execute Script
• Set Value

Some ideas for what these events can be useful:
• Jump to the home (start) page when entering LowPower mode by setting the Page ID to variable @ProjektCurrentPage.
• Initialize variables with default values.
• Mimic the same behavior like a cold boot when the device returns from LowPower or Sleep Mode.
By using a script, the ideas mentioned above can be combined.

How can I show a screen with progress bar while the project is loaded?

The loading screen image should have the same resolution as the target device (e.g. 480x272, 800x480 or 1024x768)

Only PNG images are allowed as loading screen image.

The selected image will be renamed to 'loading_screen.png' and placed in the 'terminal_files/additional_files' directory in the project folder.

PClient will display the loading screen image as soon as it starts on the device.

How to show a progress bar:
PClient automatically draws a progress bar BEHIND the loading screen image. That means if you want to show the progress bar, you have to make parts of the loading screen transparent. Through this transparent area the progress bar will be shown.
The progress bar runs from left to right of the display. It starts at 10% of the display resolution and ends at 90% of the display resolution. For a device with 480x272 resolution that would mean that it runs from X = 48 to X = 432. The height of the progress bar is the complete display resolution. If no loading screen image is selected, no progress bar will be shown.

CANopen: Can I change the Node-ID of the device at runtime?

The LSS services used for this purpose are currently not supported by the terminals.

How can I change the default backlight intensity at startup?

The default intensity for the display and the keypad backlight can be adjusted with the following variables:

@DispBacklightStartupIntensity
@KeyBacklightStartupIntensity

Both variables take values from 0 to 100, representing the percentage of the backlight intensity. For security reasons the lowest accepted value for the display backlight inten-sity is 5%.
Please make sure to not write too frequently to these variables since the value is directly stored in the devices EEPROM.

How can I use the multi color LED?

To turn on the multicolor LED in the desired color, set the variable @LED_MultiColor to the desired color value (in hexadecimal format, 0xRRGGBB, e.g. red 0xFF0000, blue 0x00FF00, green 0x0000FF.
To let the LED blink, set the @LEDMultiColorRepetitions to the desired value, then the color as explained above, and lastly set the @LEDMultiColorPattern to the desired value. Here every bit of 20 bits stands for 100ms, so setting the variable to 0xF000F will light up the LED for 400 ms, then it's 1200ms off and then again 400ms on.
When this variable is set, the repetitions are counted down to 0 and the color will be set alternating to 0 and the chosen color. At the end both the repetitions and the color will be set to 0.
Example project: Download this example to see how to use the multi color LED

Why is my device crashing when using .svg images?

First you have to make sure that you are using the tiny svg format. A second cause can be that you are using shadows. If so please disable them and try again.

How can I use the beeper in my project?

There are 3 ways to use the beeper. First of all, the general beeper volume (@BeeperMaxVolume) has to be set. Then you can use:
- alarm beeper (@BeeperAlarmVolume)
- key beeper (@BeeperClickVolume)
- general event beeper (@BeeperEventVolume)
The alarm beeper will sound when an alarm with a configured beeper pattern is opened on the display.
The key beeper will beep whenever the touch screen is touched by the user.
The general event beeper can be used freely for any chosen events. If @BeeperEventOn is set to 1, the beeper will beep, if it is set to 0, it won't.
You can also program a pattern here with @BeeperEventPattern and @BeeperEventRepetitions.

How can I use the hard keys in my project?

Encoder: The encoder is automatically used for selection and editing when you set the "Set as Input" property for objects. Additionally, you can use encoder events directly by configuring the events of the encoder. You can do this for every page.
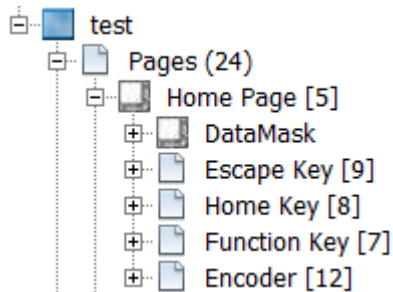
Function Key (three arrow key): You can use the events of that key for every page. When using ISO-VT, this key is fixed to switch between the ISO-VT pool(s) and PClient project.

Escape Key: This key has a fixed function to go back one page in the project.

Home Key: This key has a fixed function to go back to the home page in the project.

Power Key: This key can be configured in general in the [Project Properties](#)[76]. Alternatively, specific settings can be made on each page.

Please note that if ANY event is configured for a specific page, NO action from the project properties will be executed while on that page, i.e. if you want a specific action OnPress but the general action OnRelease, you need to put that action into the page event configuration for the Power Key, as well.

Please note that not all keys are available on all devices. Please refer to the corresponding device manual.

How can I change the page with JavaScript?

The currently shown page can be changed with JavaScript by changing the value of the variable @ProjektCurrentPage. The variable needs to be set to the ID of the page. The ID can be seen in the project tree behind the page name.

How can I convert a project?

This is described in the [project conversion](#)[40] chapter.

Audio, Video and Multimedia

For multimedia playback questions see the [FAQ of the MultimediaPlayer](#)[484]. For a description of event and alarm sound configuration see [Events and Actions](#)[430].

Everything about beeper and audio volume

In general all volume settings are made with the [beeper volume variables](#)[174].

General volume settings

The main variable to control the general volume level is *@BeeperMaxVolume*. This variable can be seen as the main volume level. It has an influence on all other volumes. The next two general volume variables are *@VolumeLevelSpeaker* and *@VolumeLevelLineOut*. These variables can be seen as mixer volume levels. If the device is equipped with an internal speaker and an audio line out, the volume level of these outputs can be adjusted (they have no influence on a device that is only equipped with a beeper). These variables directly accesses the hardware volume settings so that all other volumes will be raised or lowered when their value changes.

Alarm Volume

The following variables have an influence on the volume of an alarm sound pattern or audio file:

- @BeeperMaxVolume
- @BeeperAlarmVolume
- @BeeperMinEmergencyAlarmVolume

Sound Pattern

The beeper volume when playing an alarm sound pattern is calculated like this:

```
beeperVolume = (@BeeperAlarmVolume * @BeeperMaxVolume) / 100
if (@BeeperMinEmergencyAlarmVolume > beeperVolume)
{
    beeperVolume = @BeeperMinEmergencyAlarmVolume
}
```

Audio Output

The sound volume when playing an alarm audio file is calculated like this:

```
soundVolume = (configuredVolume * ((@BeeperAlarmVolume *
@BeeperMaxVolume) / 100)) / 100
if (@BeeperMinEmergencyAlarmVolume > soundVolume)
{
    soundVolume = @BeeperMinEmergencyAlarmVolume
}
```

While *configuredVolume* is the volume that was configured in the <u>Audio Output settings</u>[432] . Note that the final volume also still depends on *@VolumeLevelSpeaker* and *@VolumeLevelLineOut* (see <u>above</u>[758]).

Event Volume

The following variables have an influence on the volume of an event sound pattern or audio file:

- @BeeperMaxVolume
- @BeeperEventVolume

Sound Pattern

The beeper volume when playing an event sound pattern is calculated like this:

beeperVolume = (@BeeperEventVolume * @BeeperMaxVolume) / 100

Audio Output

The sound volume when playing an event audio file is calculated like this:

```
soundVolume = (configuredVolume * ((@BeeperEventVolume *
@BeeperMaxVolume) / 100)) / 100
```

While *configuredVolume* is the volume that was configured in the [Audio Output settings](#)[432] .
Note that the final volume also still depends on @*VolumeLevelSpeaker* and
@*VolumeLevelLineOut* (see [above](#)[758]).

Click Volume

The following variables have an influence on the click volume:
- @BeeperMaxVolume
- @BeeperClickVolume

The beeper/sound volume when touching the touch screen is calculated like this:

```
volume = (@BeeperClickVolume * @BeeperMaxVolume) / 100
```

Note that on devices with an internal sound card the final volume also still depends on
@*VolumeLevelSpeaker* and @*VolumeLevelLineOut* (see [above](#)[758]).

Multimedia Volume

The following variable has an influence on the volume when playing back multimedia files
with the [MultimediaPlayer](#)[484] object:
- @BeeperMaxVolume

The sound volume when playing back multimedia contents is calculated like this:

```
volume = (currentMediaPlayerVolume * @BeeperMaxVolume) / 100
```

While *currentMediaPlayerVolume* is the volume that is currently set for the
[MultimediaPlayer](#)[484] object.
Note that the final volume also still depends on @*VolumeLevelSpeaker* and
@*VolumeLevelLineOut* (see [above](#)[758]).

How can I calibrate the touchscreen of my device?

The touchscreen only needs to be calibrated if it is a resistive touchscreen. Capacitive
touchscreens do not need any calibration.
To start the touchscreen calibration use the JavaScript function
[startTouchScreenCalibration](#)[548]

How can I let objects blink?

Visual objects can be configured to blink in a certain frequency.
To do this, the JavaScript [setProperty](#)[538] function has to be used.
The properties to configure are "Flashing" and "Flashing Cycle".
Flashing enables or disables the blinking, while Flashing Cycle takes a time in milliseconds
from 100 to 5000.
The Flashing Cycle time can be changed dynamically while the object is blinking. It will then
blink with the newly set frequency.

How can I copy or move files on the device file system?

To copy or move files on the device file system (or to / from the USB stick), you need to use
4 variables in the following form and order (example to move an alarm log file to the trash,
i.e. to a folder that will be deleted upon restart):

```
//Set the current directory to where the log file is situated
```

```
setVariableValue("@CurrentDirectory",
"/opt/pclient/logfiles/");
//Set the current file name to "alarm.log", the name of the
file
setVariableValue("@CurrentFile", "alarm.log");
//Set the destination to a temporary folder which will be
deleted after restart
setVariableValue("@DestinationDirectory", "/dev/null/");
//Execute the copy/move process
setVariableValue("@CopyFileOrFolder", 2);
```

Please note:
- no slash before or after the file name
- @CurrentDirectory needs to be set before @CurrentFile

How can I rename my project?

There are two ways to rename a project. One is by performing a Save As action. This menu entry can be found in the menu File -> Save As
Please note that the project will be stored in the path that is written in the **Folder name** field, so the new project name has to be added after the existing path with a \ in front, e.g.: C:\projects is the existing path, then the Folder name field needs to be C: \projects\new_project_name.
The second way is to right-click on the project name in the project tree and select **Rename** in the context menu. This will keep the project location and just change the project name (and thus the project folder).
Please note: Manually changing the project folder name does not rename the project, and it will lead to problems. Only use one of the methods described here.

How can I change the order in which the Encoder navigates between objects?

The order is dependent on the Z-Order[151] of the objects. By changing the Z-Order you can modify in which order objects are selected.

How can I create landscape / portrait projects?

All devices can be used in landscape (widescreen / horizontal) or portrait (vertical) mode. The distinction is made when creating a project. For every device type, a landscape or portrait mode project can be created. This cannot be changed later (only by converting[40]).

Please note: For i.MX 6 device projects, when changing from a landscape to a portrait project on the device, a restart is necessary to apply the setting.

What can I do if a font file can't be imported?

A font file can only be imported if there isn't already a font file with the same name imported in the project. If a font comes with several files for bold, italic etc, it is possible that they have the same name in their meta data. If this is the case, give the fonts unique names and make sure that all name fields have the same name. Please refer to the passage Imported Fonts[76] in the chapter Project Properties[76].

How can I calculate with numbers in unsigned32 in JavaScript

You can force the interpretation of a number in JavaScript in unsigned32 by executing this code:
var number_us32 = number_original >>> 0;

How can I set the IP of the device?

To change the IP of a device while it is running, use the variable @Ethernet0_IPAdress[174].
But this is not a permanent setting!
The IP address that will be set on startup can and has to be defined in the project updater[618]. The Ethernet port will be available as soon as the address is set.

I connected an Ethernet camera and did everything right, but why don't I see an image?

1. The camera needs to be set to H.264
2. The UDP output port of the camera needs to be set in the video frame/page properties
3. Even when all camera settings are correct and the project is set up correctly, it's important to remember that Ethernet needs to be activated, this does not happen automatically with an Ethernet Camera configured and connected.

How can I make changes to the file system when loading the project?

Sometimes you may want to make certain changes to the file system upon project installation. An example would be the creation of the path /opt/user_files or to copy some external files somewhere. You can add files in the project update process[639]. There you can also edit the post-installation script which can be modified to add necessary commands (e.g. mkdir /opt/user_files).

How can I change the line appearance in the 2D Graph?

Line attributes define the look of the lines in the 2D Graph[451]. They can be modified and new attributes can be created in the project properties[76].

How can I display languages with other characters?

The possible characters depend on the font supporting them.
Please find free fonts for the language you need and import them in the project properties[76].
For asian characters e.g. consult this list: https://en.wikipedia.org/wiki/List_of_CJK_fonts

Why is the save button disabled?

The save button is only enabled if any change to a project has been made. No changes - no need to save the project.
A consequence of that is that if a project has been saved and the program is closed, no saving dialog will pop up, the program will just close.

(How) Can I make screenshots of my project on the device?

You can save screenshots of your project (i.e. what is currently shown on the screen) by using the variable @ProjektTakeScreenshot[174]

Why aren't my GIF files displayed in the project?

We support the GIF file format mainly because of its animation possibility. GIF files are only supported for the Image to Display property of Picture Graphics. For all other places and for inanimated images, please use one of the other supported image formats.

Can I show a PDF file on the device?

Yes! We have a PDF reader included in the PClient. You can open a PDF file on the device by using button events[442] or by using the JavaScript function openPDFReader[556] . Note that the PDF reader doesn't work in the PClient simulation.

OPUS devices update manual

14 OPUS devices update manual

14.1 Overview

OPUS devices run different kinds of software, depending on the ordered product variant. Here is an overview of the different software types:

- **OS** – Operating System (before: BSP - Board support package)
  This is the operating system of the device, a customized embedded Linux. It is installed on every OPUS A3 / A6 / A6 G2 / A8 device. If you are programming with C / C++ / Qt, you only need to install the OS files.

- **PClient** – Projektor Client
  This software is installed on devices that were ordered as CAN devices (e.g. article number OPUSA3EN1**CAN**F000). The PClient is the counterpart of the Projektor Tool on the PC and part of the OPUS Toolchain.

- **Codesys Runtime**
  This software is installed on devices that were ordered as CDS devices (e.g. article number OPUSA3EN1**CDS**F000) or Codesys Webvisu devices (CWV, e.g. article number OPUSA3EN1**CWV**F000). It is the counterpart of the Codesys development environment on the PC. The Codesys runtime contains a per-device license and runs only on devices with CDS in the article number.
  Please note that you can also install the PClient on Codesys devices.

- **ISO-VT**
  This software is installed on devices that were ordered as ISO devices (e.g. article number OPUSA3EN1**ISO**F000). ISO-VT (Virtual Terminal) is part of the standard ISOBUS. ISO-VT contains a per-device license and runs only on devices with ISO in the article number.
  Please note that you can also install the PClient on these devices.
  Please note that this software is not developed further.

- **ISO-V2**
  This software is installed on devices that were ordered as ISO V2 devices (e.g. article number OPUSA8SN1**UTL**F000). ISO-V2 is part of the standard ISOBUS. The ISO V2 software only runs on devices with UT in the article number. Upgrades can be bought to extend the functionality.

> **To be sure you have the latest software running on the devices, please always perform a software update on devices you purchased.**

If you have any problems during the update, please contact the Topcon Electronics support team:

opus-support@topcon.com
+49 6722 4026 888

14.2 Contents of the software update

In this chapter, the files that are used for the different update parts are described.

14.2.1 OS 1.x

The OS 1.x contains the following files:

Redboot_RAM.bin – a binary image with the bootloader
zImage – a binary image with the Linux kernel
zImage.service– a binary image for the service mode
rootfs.jffs2 – a binary image with the JFFS2 root file system
logo.png (optional) – a boot logo which is displayed at every system start

Additionally, there is an md5 checksum file for each of the above files which are used to check the integrity of the files in the installation.

OS 1.x is not developed any further and should not be used for new development projects.

14.2.2 OS 2.x

The OS 2.x contains the following files:

Boot_opusaX_2.X.bin – a binary image with the bootloader
Redboot_RAM.bin – a binary image with the bootloader for an update from an older (1.x) OS version (only necessary when updating from a BSP < 2.x)
linux_opusaX_2.X.bin – a binary image with the Linux kernel
service_opusaX_2.X.bin – a binary image for the service mode
zImage.service – a binary image for the service mode for an update from an older (1.x) OS version (only necessary when updating from a BSP < 2.x)
rootfs_opusaX_2.X.squashfs – a binary image with the squash root file system
logo_opusaX_2.X.png (optional) – a boot logo which is displayed at every system start

Additionally, there is an md5 checksum file for each of the above files which are used to check the integrity of the files in the installation.

14.2.3 OS EGSPRO

The OS EGSPRO contains the following files:

boot_egspro_X.X.X.bin – a binary image with the bootloader
pmic_egspro_X.X.X.sud – a binary file with the firmware for the power management circuit
rootfs_egspro_X.X.X-we-x11-qt5.tar.gz – Root file system package including the Linux kernel
dev-tree_egspro_X.X.X.tar.tar.gz – Device tree file to install the correct device configuration into the device
uImage – update RAMDISK image (optional, only if a 0.x OS is installed before)
logo_egspro_a6-we-landscape.bmp (optional) – boot logo for A6 for normal start
recovery_egspro_a6-we-landscape.bmp (optional) – boot logo for A6 for update mode
logo_egspro_a8-we-landscape.bmp (optional) – boot logo for A8 for normal start
recovery_egspro_a8-we-landscape.bmp (optional) – boot logo for A8 for update mode

update.img (only for upgrade from 1.x to 2.x **and from 1.x or 2.x to 3.x**) – updater image file

Additionally, there is an md5 checksum file for each of the above files (except for the uImage file) which are used to check the integrity of the files in the installation.

14.2.4 OS EGSCORE

The OS EGSCORE contains the following files:

boot_egscore_X.X.X.bin – a binary image with the bootloader
rootfs_egscore_X.X.X-we-x11-qt5.tar.gz – Root file system package including the Linux kernel
dev-tree_egscore.tar.gz – Device tree file to install the correct device configuration into the device
comic_egscore_X.X.sud – Co-processor update file
service_egscore_x.bin – Service kernel
logo_egscore_b3-te-landscape.bmp (optional) – boot logo for normal start
recovery_egscore_b3-te-landscape.bmp (optional) – boot logo for A6 for update mode

update.img – updater image file

Additionally, there is an md5 checksum file for each of the above files which are used to check the integrity of the files in the installation.

14.2.5 OPUS Toolchain – PClient

The PClient consists only of one file:

user.tar.gz – zipped package with PClient executable, additional files and standard project (**user_opusax_x.y.tar.gz** for OS 2.x, **user_egspro_x.y.tar.gz** for EGSPRO OS, **user_egscore_x.y.tar.gz** for EGSCORE OS)

Additionally, there is an md5 checksum file for this file which is used to check the integrity of the files in the installation.

> ## In Windows, the user.tar files are often shown as zip files. Never unzip these files or the update won't work!

14.2.6 Codesys

The Codesys runtime consists only of one file (OS 1.x):

user.tar.gz – zipped package with Codesys executable, additional files and service project

For OS 2.2.4 and newer, there are two files:

user_opusaX_x.y.tar.gz – zipped package with Codesys executable and additional files
user2_opusaX_x.y_Service.tar.gz – zipped package with service project

For OS EGSPRO there are two files:

user_egspro_x.y.tar.gz – zipped package with Codesys executable and additional files
user2_egspro_x.y_Service.tar.gz – zipped package with service project

For OS EGSCORE there are two files:

user_egscore_x.y.tar.gz – zipped package with Codesys executable and additional files
user2_egscore_x.y_Service.tar.gz – zipped package with service project

Additionally there is an md5 checksum file for these files which are used to check the integrity of the files in the installation.

> **In Windows, the user.tar files are often shown as zip files.**
> **Never unzip these files or the update won't work!**

14.2.7   ISO-VT

The ISO-VT program consists only of one file:

user.tar.gz – zipped package with PClient and VT (virtual terminal) executable, additional files and ISO-VT settings project (**user_opusax_2.x.tar.gz f**or OS 2.x, **user_egspro_x.x.tar.gz f**or EGSPRO OS)

Additionally there is an md5 checksum file for this file which is used to check the integrity of the files in the installation.

> **In Windows, the user.tar files are often shown as zip files.**
> **Never unzip these files or the update won't work!**

14.2.8   ISO-V2

The ISO-V2 program consists of these files:

egspro/egscore-x-rootfs.img – package with the OS and ISO application
egspro/egscore-x-install.sh – install script for the update
uImage – updater image that installs the update
pkginfo.sign – checksum file
pkginfo.json – checksum file
hostnames – reference file for device MAC addresses
egspro-x-extras.img – additional install packages
Console.ini – settings file

There are currently three types of this release (substituting the **x** in the file name):

A6_TELE: only for OPUS A6 Generation 2 Dual Core devices (e.g. OPUSA6EN2UTX**D**000)
A8_TELE: only for OPUS A8 Quad Core devices (e.g. OPUSA8EN2UTX**F**000)
B3_TELE: only for OPUS B3 devices (e.g. OPUSB3EN1ISOF000)

> **Please note that no other OS files from other releases are needed in addition to the files listed here! These files contain everything**

Please note that the boot logo cannot be changed with these devices like it can for OPUS devices. If you need to change the boot logo, please contact opus-support@topcon.com

14.3 Updating the device

In this chapter, the update process for the different device types is described.

14.3.1 OPUS A3 / A6 G1

The first step to update the device is to **copy the update files on a USB stick**. The USB stick has to be **FAT32 formatted** with only **one partition**.
Copy the OS and/or the files for the application that should run on the device into the root folder of the USB stick. The USB stick may contain other data, they will be ignored.
You can find the update files in the extracted zip file that you downloaded / got on your USB stick.

> **In Windows, the user.tar files are often shown as zip files. Never unzip these files or the update won't work!**

For **OPUS Projektor**: Starting from OPUS / WE Toolchain 1.11, the update files for the devices are included in the Projektor Tool. Go to the menu Update -> PClient / OS Update. In the dialog you can choose for which device you want to copy the OS and PClient files on the USB stick.
For older versions, check the downloaded zip file:
\PClient\OPUSAx\OPUSA3L(andscape)_BSP_PClient_Update (for landscape, for portrait it's OPUSA3P(ortrait)_BSP_PClient_Update)

For **ISO-VT**: \OPUS Ax

For **Codesys**: \CustomerFiles\Device\OPUSAx

Power off the device and connect the USB stick with the update files with the USB port of the device (both the front USB slot and the USB cable from the main connector can be used).
There are **two ways to start the device in update mode**, you can use either one of them (but you don't need to do both!):

   a) **Connect the clamp "Serv_EN"** (service enable) **with clamp 15 and 30** and then **power on the device**.

After some seconds a screen with "Maintenance" will be displayed. At this point please **disconnect the clamp "Serv_EN"**, otherwise the update will be done again after restart.

b) **Press the left keys 1 and 2** and then **power on the device** while holding the keys (works only if you have a product version with keys).

After some seconds a screen with "Maintenance" will be displayed. At this point you can let go of the keys.

Sometimes it happens that the update doesn't start initially and asks for a USB stick. If the update process didn't start for about a minute, please disconnect the USB stick and connect it again. Then the update will start.

Let the update run through completely. Don't disconnect the power when the reboot countdown is displayed

Once the device restarted, there will be a touch screen calibration process, if the device has a touch screen. You can read about that process here.

14.3.1.1   Updating from OS 0.x / 1.x to 2.x

Warning: All files in /opt will be deleted with this update.

You can't update from OS 0.x to 2.x directly, please update to 1.x first

If you still have an OS 0.x on your device, please first update to a 1.x OS. You can't directly update from OS 0.x to OS 2.x!

Due to a change in the file system the update from 1.x to 2.x will be done in **two steps**. In the first step only the two files redboot_RAM.bin and zImage.service will be installed.
After that the device is restarted and the other files are installed.
The second start can take up to 3 minutes because the mass storage has to be initialized with the new file system.

14.3.1.2   Downgrading from OS 2.x to 1.x

Warning: All files in /opt and /diag will be deleted with this update.

Due to a change in the file system the **downgrade from 2.x to 1.x has to be done in two steps**. In the first step, the updater will only install the following files from the OS 1.x update:

- redboot_RAM.bin
- zImage
- zImage.service
- rootfs.jffs2

There **MUST NOT** be any files from an OS 2.x on the USB stick, because then the updater will try and install these files and ignore the 1.x update files. You can put all OS 1 files on the stick, though.

After this downgrade has been installed, a second update can be done which includes the boot logo and the desired application (user.tar.gz file).

14.3.2   OPUS A6 G2 / OPUS A8

The first step to update the device is to **copy the update files on a USB stick**. The USB stick has to be **FAT32 formatted** with only **one partition**.
Copy the OS files and/or the files for the application that should run on the device into the root folder of the USB stick. The USB stick may contain other data, which is okay, they will be ignored. You can find the update files in the extracted zip file that you downloaded / got on your USB stick.

Warning: When upgrading to OS EGSPRO 2.x, a downgrade to EGSPRO 1.x is NOT POSSIBLE!

In Windows, the user.tar files are often shown as zip files. Never unzip these files or the update won't work!

To update from EGSpro 1.x to EGSpro 2.x, the additional files update.img and update.img.md5 must be included on the update USB stick.

To update from EGSpro 1.x or 2.x to EGSpro 3.x, the additional files update.img and update.img.md5 must be included on the update USB stick.

For **OPUS Projektor**: Starting from OPUS / WE Toolchain 1.11, the update files for the devices are included in the Projektor Tool. Go to the menu Update -> PClient / OS Update. In the dialog you can choose for which device you want to copy the OS and PClient files on the USB stick.
For older versions:
\PClient\EGSPRO\OPUSA3L(andscape)_BSP_PClient_Update (for landscape, for portrait it's OPUSA3P(ortrait)_BSP_PClient_Update)

For **ISO-VT**: \EGSPRO

For **Codesys**: \CustomerFiles\Device\EGSPRO

Power off the device and connect the USB stick with the update files with the USB port of the device (both the front USB slot and the USB cable from the main connector can be used). **Connect the clamp "Serv_EN"** (service enable) **with clamp 15 and 30** and then **power on the device**.
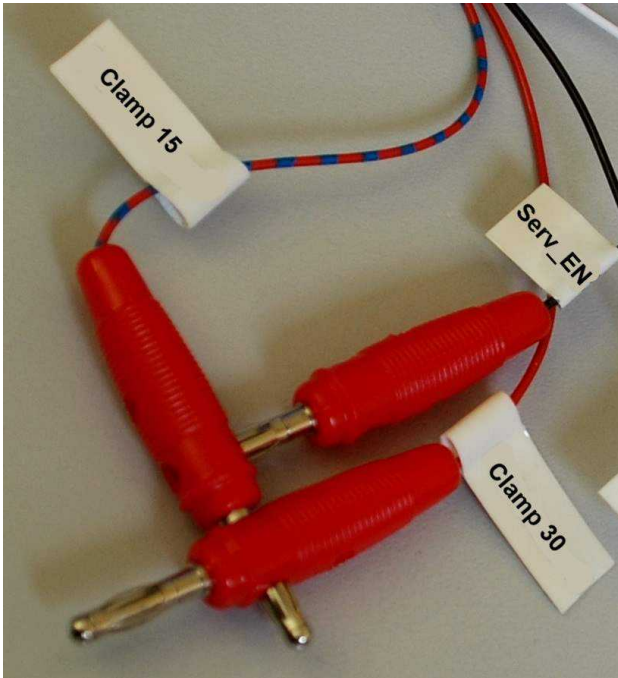
Alternatively, you can also press left keys 1 + 2 for A6 G2 Standard, keys 1 and stop for OPUS A8 Standard, during power on.

After some seconds a screen with "Maintenance" will be displayed. At this point please **disconnect the clamp "Serv_EN"**, otherwise the update will be done again after restart.

14.3.3 OPUS B3

The first step to update the device is to **copy the update files on a USB stick**. The USB stick has to be **FAT32 formatted** with only **one partition**.
Copy the OS files and/or the files for the application that should run on the device into the root folder of the USB stick. The USB stick may contain other data, which is okay, they will be ignored. You can find the update files in the extracted zip file that you downloaded / got on your USB stick.

In Windows, the user.tar files are often shown as zip files. Never unzip these files or the update won't work!

For **OPUS Projektor**: Go to the menu Update -> PClient / OS Update. In the dialog you can choose for which device you want to copy the OS and PClient files on the USB stick.
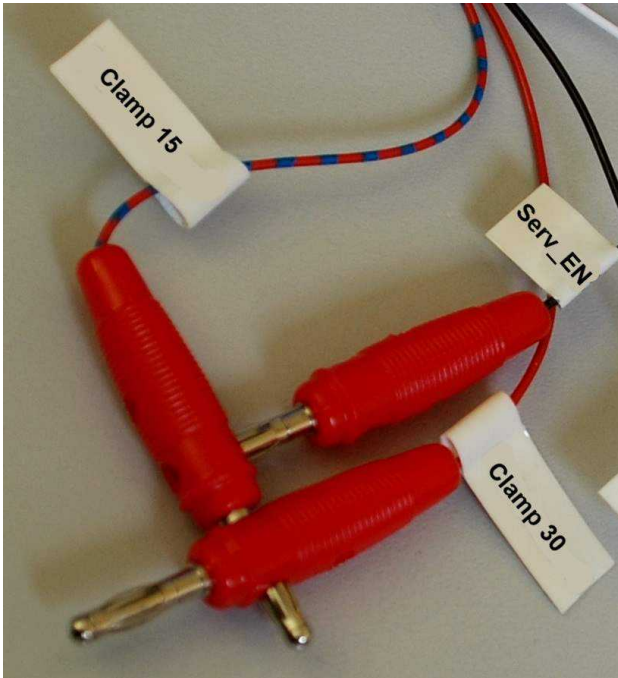
For **Codesys**: \CustomerFiles\Device\EGSCORE

Power off the device and connect the USB stick with the update files with the USB port of the device (both the front USB slot and the USB cable from the main connector can be used). **Connect the clamp "Serv_EN"** (service enable) **with clamp 15 and 30** and then **power on the device**.

After some seconds a screen with "Maintenance" will be displayed. At this point please **disconnect the clamp "Serv_EN"**, otherwise the update will be done again after restart.

14.3.4   ISO-V2

The first step to update the device is to **copy the update files on a USB stick**. The USB stick has to be **FAT32 formatted** with only **one partition**.
Copy the correct update files for the device into the root folder of the USB stick. The USB stick may contain other data, which is okay, they will be ignored. You can find the update files in the extracted zip file that you downloaded / got on your USB stick.

There are two ways to start the update process.

1. Via Setup menu (Please use this method)

Insert the USB stick with the update files into the running device.
On the device, go to the menu
User -> Utilities or **System -> Utilities** (depending on the currently installed version)
and you will see an entry **CONSOLE SOFTWARE UPGRADE**. Please select this entry.
There will be a warning dialog "Operation Requires Restart". Confirm the dialog by pressing the green checkbox button.
Then the device will restart and the update will be installed.
Please note that the menu entry is only active if valid update files are on the USB stick.

2. Via Service mode (Please DON'T use this method, as licenses and settings can be deleted)

Power off the device and connect the USB stick with the update files with the USB port of the device (both the front USB slot and the USB cable from the main connector can be used).
There are **two ways to start the device in update mode**, you can use either one of them (but you don't need to do both!):

a) **Connect the clamp "Serv_EN"** (service enable) **with clamp 15 and 30** and then **power on the device**.
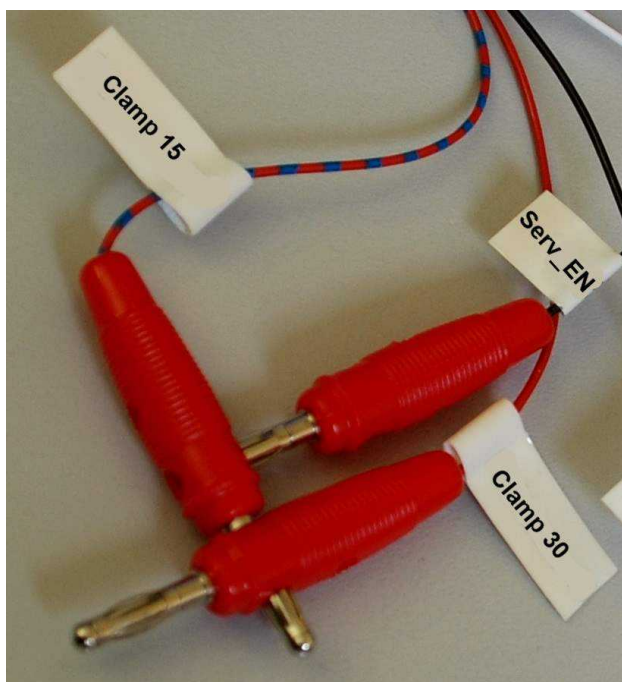
After some seconds please **disconnect the clamp "Serv_EN"**, otherwise the update will be done again after restart.

b) **Press the left keys 1 and 2** and then **power on the device** while holding the keys (works only if you have a product version with keys). After some seconds please let go of the keys.

When the update starts, you will see a small bar graph in the bottom of the screen. Once the bar graph is filled, the device will restart and the update was successful.

14.3.5 Update via partition

It is possible to update OS and application without a USB stick (not valid for ISO-V2).
For this, the files have to be put into a special folder that might have to be created at first.

For A3/A6G1: **/opt/var/update**
This folder might have to be created first:
```
mkdir /opt/var/update
```

For A6g2/A8/B3/B4: **/var/shadow/update** (without update images or uImage files)
For A6g2/A8/B3/B4 you can also put a uImage file into: **/var/shadow/boot**

To use the partition, it needs to be remounted as read-write first:
```
mount -o remount,rw /var/shadow
```

14.4 Additional information

14.4.1 Connecting to the serial console

You can connect to the serial console of the device to change the boot logo, read debug and error messages and copy files and folders.
You need to connect your computer to the RS232 interface of the device, either directly with an RS232 cable, if your computer has the according connector, or with a RS232 <-> USB cable.

You can use any shell program, such as HyperTerminal (only up to Windows XP), PuTTY or TeraTerm VT (both free and open source).
These are the settings you need to make in your software:
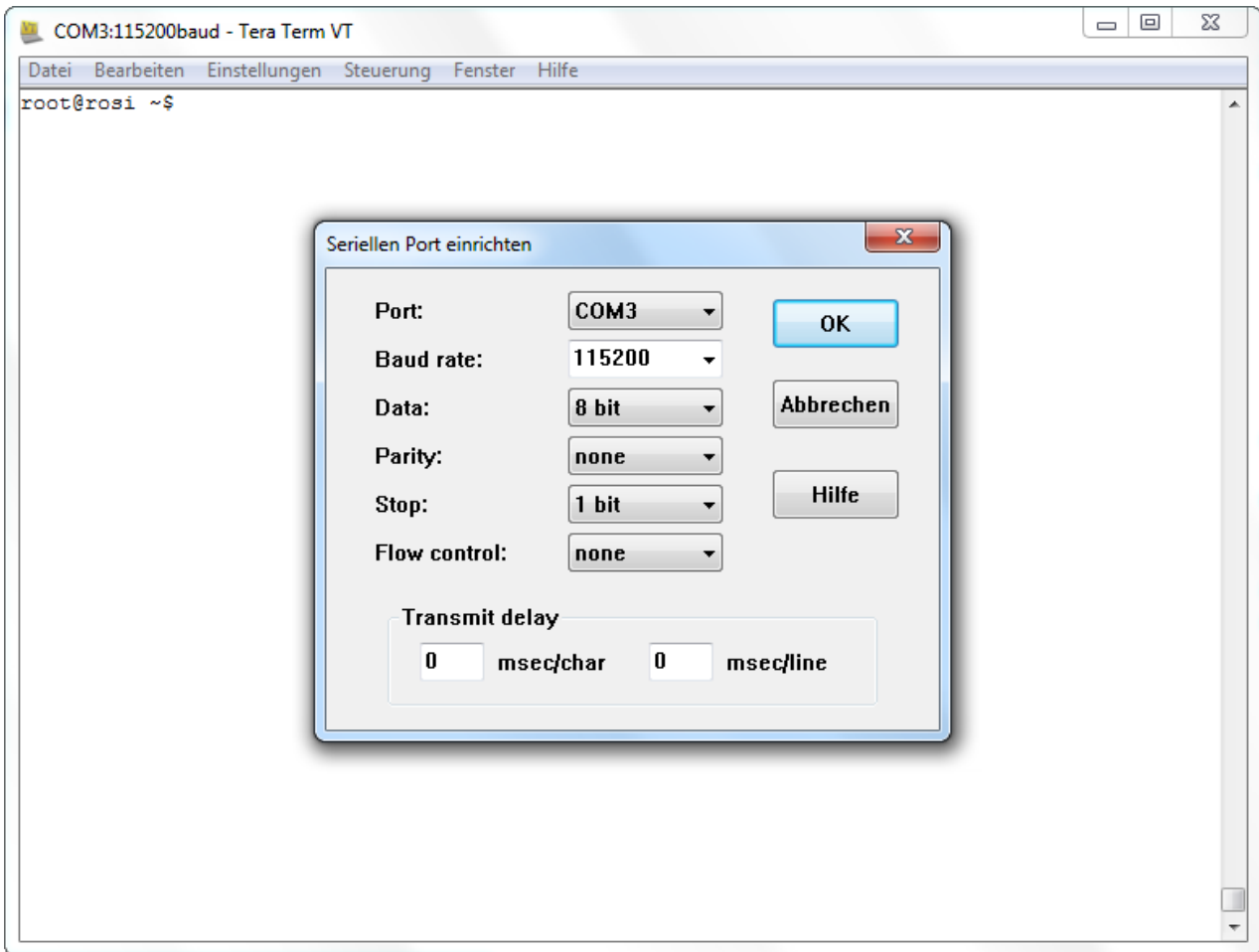
The COM-Port can differ depending on your system.

If you start up the device with the console connected, you will see a lot of startup messages. If you connect while the device is running, you might not get any messages at first. Just press enter to get the login dialog:
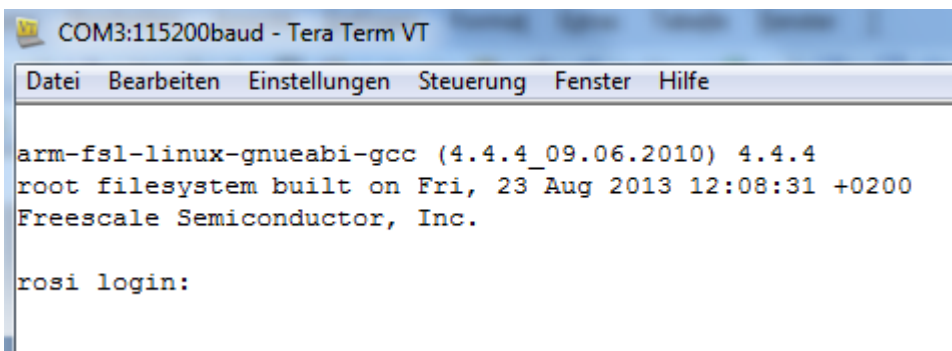
Here are the login parameters:

login name: **root**
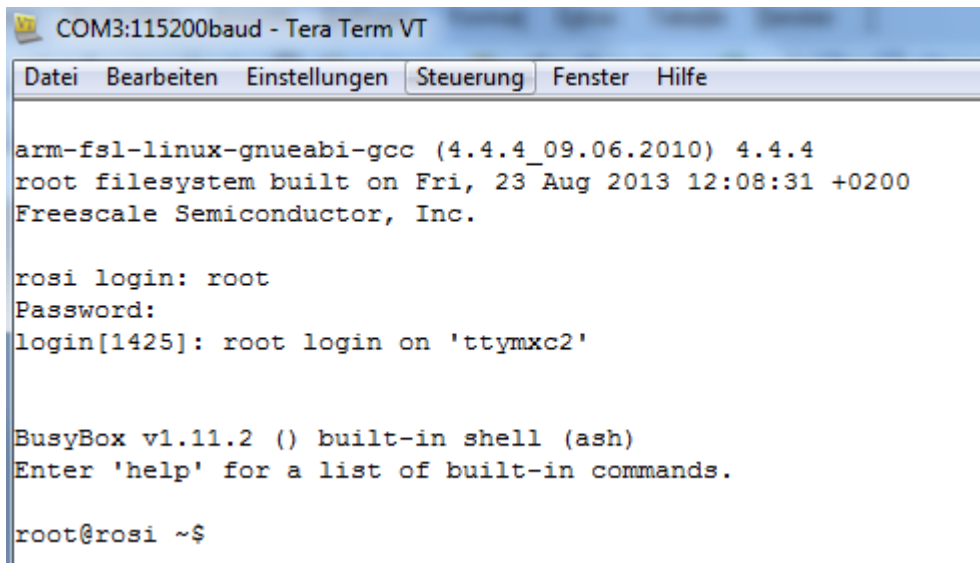
password:
for A3 -> **opusa3**
for A6 -> **opusa6**
for A8 -> **egspro**
for A6 G2 -> **egspro**
for B3 -> **egscore**

After logging in you will see a screen similar to this (it differs from device to device):


```
COM3:115200baud - Tera Term VT
Datei  Bearbeiten  Einstellungen  Steuerung  Fenster  Hilfe

arm-fsl-linux-gnueabi-gcc (4.4.4_09.06.2010) 4.4.4
root filesystem built on Fri, 23 Aug 2013 12:08:31 +0200
Freescale Semiconductor, Inc.

rosi login: root
Password:
login[1425]: root login on 'ttymxc2'


BusyBox v1.11.2 () built-in shell (ash)
Enter 'help' for a list of built-in commands.

root@rosi ~$
```

Now you're logged into the Linux system on the device.

14.4.2   Using a different boot logo image

> **Besides the processes described below, both in Codesys and OPUS Toolchain the boot logo can be changed within the project.**

To use your own boot logo image you need to prepare it for the update process. The image file(s) need to be in the correct solution:

- A3: 480x272
- A6: 800x480
- A6 G2: 800x480
- A8: 1280 x 800

These are the names and formats you need to save the files in to be used in the update:

- **logo.png** for OS 1.x

- **logo_opusaX_2.X.png** for OS 2.x (exchange X with the appropriate numbers)

- **logo_egspro_a6-we-landscape.bmp** for OS EGSPRO with A6
- **recovery_egspro_a6-we-landscape.bmp** for OS EGSPRO with A6 (update screen)

- **logo_egspro_a8-we-landscape.bmp** for OS EGSPRO with A8
- **recovery_egspro_a8-we-landscape.bmp** for OS EGSPRO with A8 (update screen)

- **logo_egscore_b3-te-landscape.bmp** for OS EGSCORE with B3
- **recovery_egscore_b3-te-landscape.bmp** for OS EGSCORE with B3 (update screen)

Additionally, you need to create the MD5 checksum file.

The checksum in the .md5 file needs to contain lower case letters and numbers only!

If you have an existing check sum file (e.g. from our boot logo), you can create the check sum online, e.g. on https://emn178.github.io/online-tools/md5_checksum.html and replace the existing checksum in the .md5 file with the newly created one.

You can also use a Windows tool like WinMD5sum: http://www.nullriver.com/products/winmd5sum/

Lastly, you can use the Linux system on our device to create the checksum file. Follow these steps to do that:

1. Create the image file as described above
2. Put the image on a FAT32 formatted USB stick
3. Log in to the serial console (login: root, password: A3: opusa3, A6: opusa6, A6 G2 & A8: egspro)
3b. Connect the USB stick to the running device
4. Check that the USB stick is mounted to /disk/usbsda1 by executing the command `df`
5. Note the path of the USB stick
6. Write and execute the command:

```
md5sum /disk/usbsda1/logo.png > /disk/usbsda1/logo.png.md5
```
(**for OS 1.x**)

```
md5sum /disk/usbsda1/logo_opusaX_2.X.png
> /disk/usbsda1/logo_opusaX_2.X.png.md5
```
(**for OS 2.x**)

```
md5sum /disk/usbsda1/logo_egspro_a6-we-landscape.bmp
> /disk/usbsda1/logo_egspro_a6-we-landscape.bmp.md5
```
(**for OS EGSPRO**)

```
md5sum /disk/usbsda1/logo_egscore_b3-te-landscape.bmp
> /disk/usbsda1/logo_egscore_b3-te-landscape.bmp.md5
```
(**for OS EGSCORE**)

(Change the USB path if the stick was mounted in a different place)

7. Check if the file was created by executing `ls /disk/usbsda1` and, optionally, `cat /disk/usbsda1/<filename>.md5`

The MD5 file has to be encoded in ANSI or UTF8 without BOM. UTF8 doesn't work.

The letters in the checksum have to be lower case.

The line endings in the .md5 file have to be Linux line endings (LF only)

You can use these two files in the update process described above now and the boot logo will be changed to that image. Be aware that you always have to re-create the md5 checksum file when you change the boot logo image.

There is a second way to change the boot logo directly on one device (quicker for one device, but not suitable for several devices):

1a . (**OPUS A3/A6 G1**). Create a **png** image in the appropriate resolution (A3: 480x272, A6: 800x480)
1b . (**OPUS A6G2/A8**). Create a **bmp** image in the appropriate resolution (A6 G2: 800x480, A8: 1280x800)
1c . (**OPUS B3**). Create a **bmp** image in the appropriate resolution (B3: 800x480)

2. Put the image on a FAT32 formatted USB stick
3. Log in to the serial console (see chapter 4.1)
3b. Connect the USB stick to the running device
4. Check that the USB stick is mounted to /disk/usbsda1 by executing the command `df`
5. Note the path of the USB stick
6 a. (**OPUS A3/A6 G1**). Write and execute the command:
setbootlogo /disk/u-sbsda1/imagename.png (change "imagename" to the actual name of the image, and change the USB path if necessary)
6 b. (**EGSPRO/EGSCORE devices**). To copy the boot logo for the normal start, please use the following command (<image.bmp> is the path to the image file):
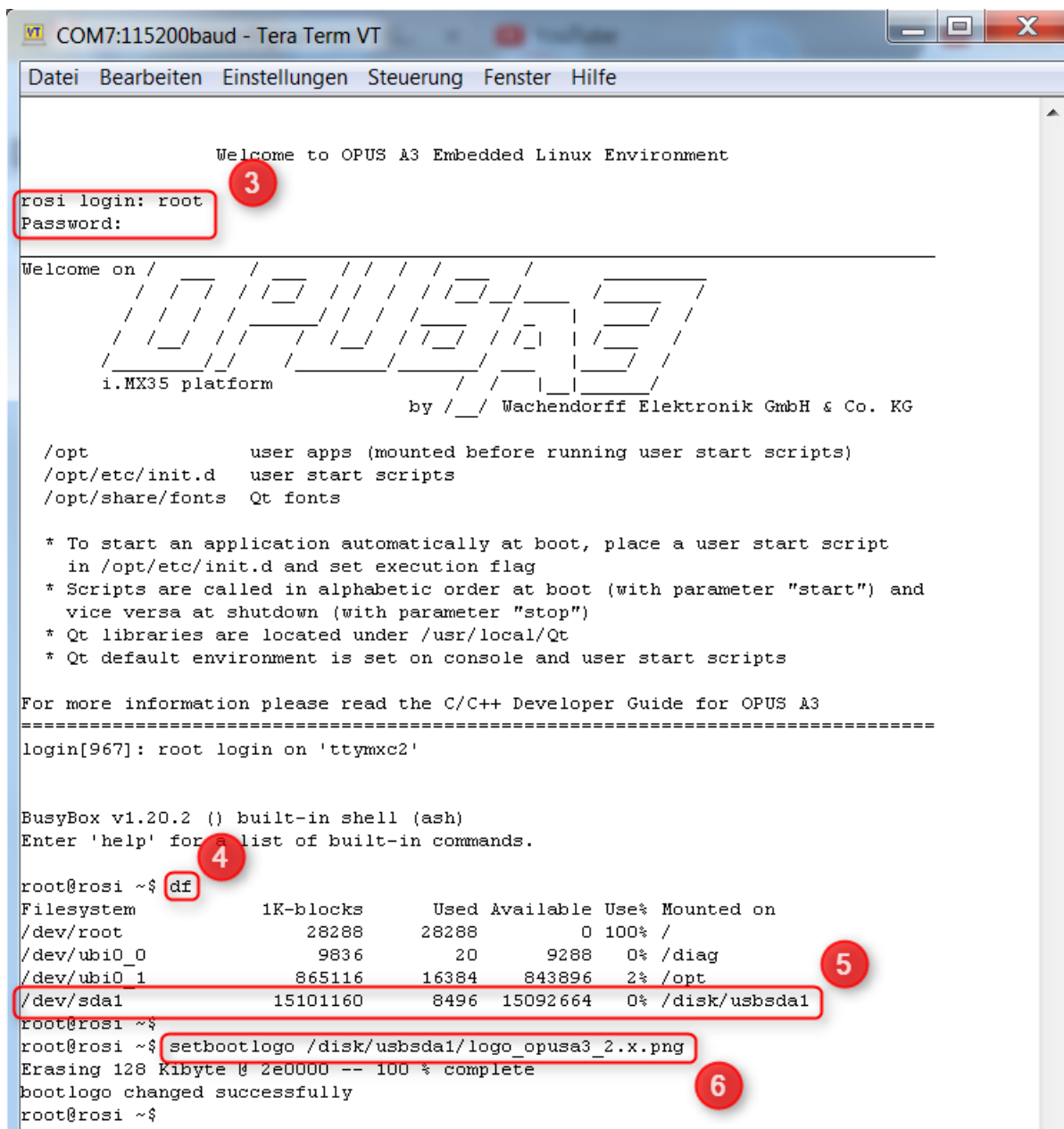cp <image.bmp> /boot/logo.bmp
To change the recovery boot logo, use this command:
cp <image.bmp> /boot/recovery.bmp
Afterwards, execute the command
sync
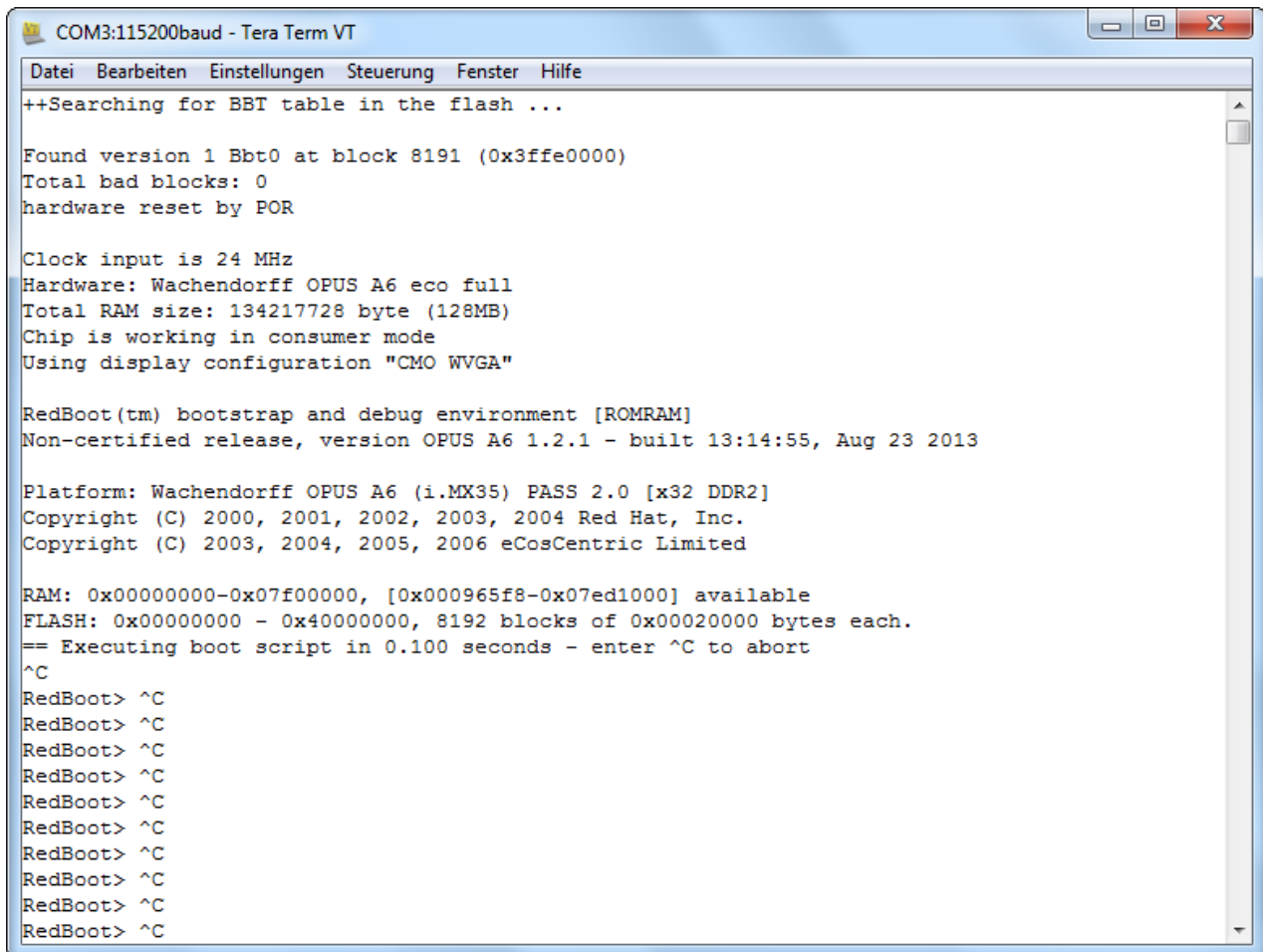
14.4.3 Manual update of the boot loader and service ramdisk (OS 1.x)

If the wrong boot loader and/or service ramdisk have been installed (e.g. A6 files on A3 or vice versa), you need to install the correct boot loader / service ramdisk file manually.
For OS 2.x this doesn't work anymore because the filenames containing the display name prohibit a wrong installation.

14.4.3.1 Manual update of the boot loader

1. **Connect to the serial console** of the powered device (see chapter 4.1). Be aware that this **doesn't work with the program PuTTY**! Hyperterminal and Tera Term will work.
Then unplug the power from the device and be sure the shell program is in focus on your PC.

2. **Press CTRL + C** on your keyboard, hold it pressed and then **power on the device**. If you were successful you will see the following screen:

```
COM3:115200baud - Tera Term VT

Datei  Bearbeiten  Einstellungen  Steuerung  Fenster  Hilfe

++Searching for BBT table in the flash ...

Found version 1 Bbt0 at block 8191 (0x3ffe0000)
Total bad blocks: 0
hardware reset by POR

Clock input is 24 MHz
Hardware: Wachendorff OPUS A6 eco full
Total RAM size: 134217728 byte (128MB)
Chip is working in consumer mode
Using display configuration "CMO WVGA"

RedBoot(tm) bootstrap and debug environment [ROMRAM]
Non-certified release, version OPUS A6 1.2.1 - built 13:14:55, Aug 23 2013

Platform: Wachendorff OPUS A6 (i.MX35) PASS 2.0 [x32 DDR2]
Copyright (C) 2000, 2001, 2002, 2003, 2004 Red Hat, Inc.
Copyright (C) 2003, 2004, 2005, 2006 eCosCentric Limited

RAM: 0x00000000-0x07f00000, [0x000965f8-0x07ed1000] available
FLASH: 0x00000000 - 0x40000000, 8192 blocks of 0x00020000 bytes each.
== Executing boot script in 0.100 seconds - enter ^C to abort
^C
RedBoot> ^C
RedBoot> ^C
RedBoot> ^C
RedBoot> ^C
RedBoot> ^C
RedBoot> ^C
RedBoot> ^C
RedBoot> ^C
RedBoot> ^C
RedBoot> ^C
```

Once you see the RedBoot> prompt, you can let go of CRTL + C.
If this doesn't work, close TeraTerm/HyperTerminal and start it again.

3. Have the correct boot loader file (*Redboot_RAM.bin*) ready on your computer.
Now enter the following command:

```
load -r -b 0x100000 -m ymodem
```

The device now waits for you to send a file.
Depending on the program you're using, send the boot loader file using the **ymodem** protocol.

For Tera Term:
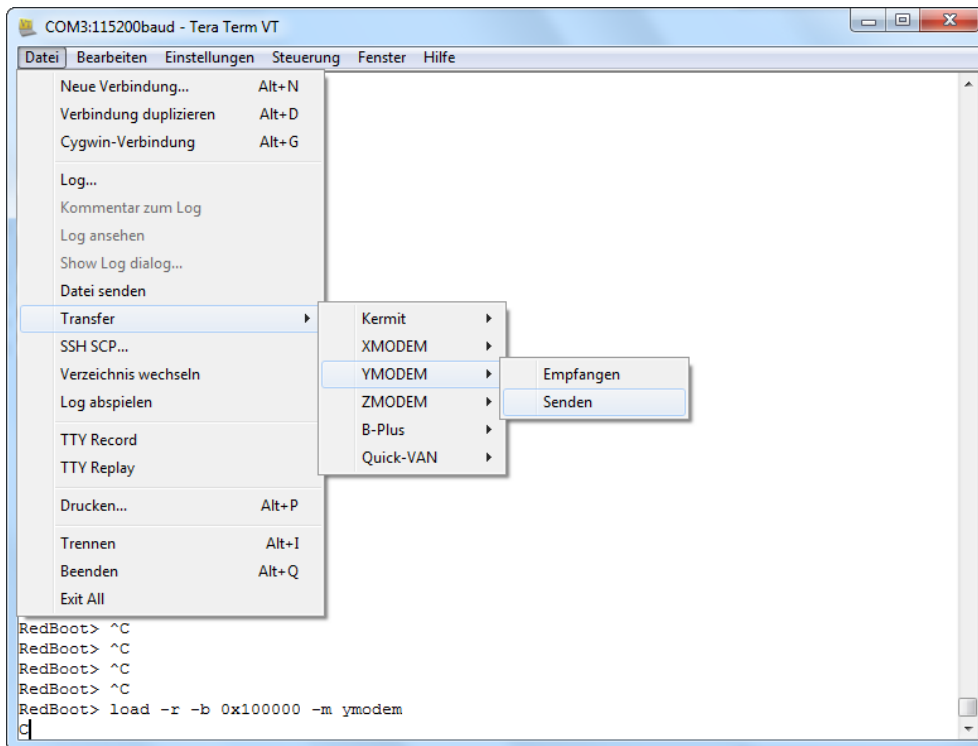
In HyperTerminal choose the menu **Transfer -> Send file** and choose the boot loader file in the dialog:
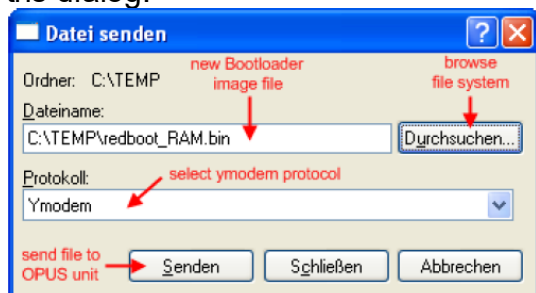
Then a dialog which shows the file transfer progress is displayed.

Tera Term: HyperTerminal:

The file transfer will take about 30 seconds. After successful file transfer the dialog is closed automatically and you will retrieve the boot prompt again.

4. Now start the boot loader by entering the command

```
run
```

but

5. stop the execution of the boot script by pressing CTRL+C instantly after pressing Enter.

6. As a last step, the boot loader has to be programmed into the flash memory by entering the command

```
romupdate
```

Now power off the device and try installing the whole OS using the method described in chapter 3.
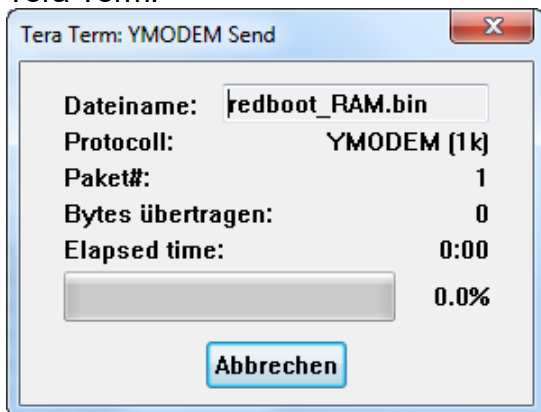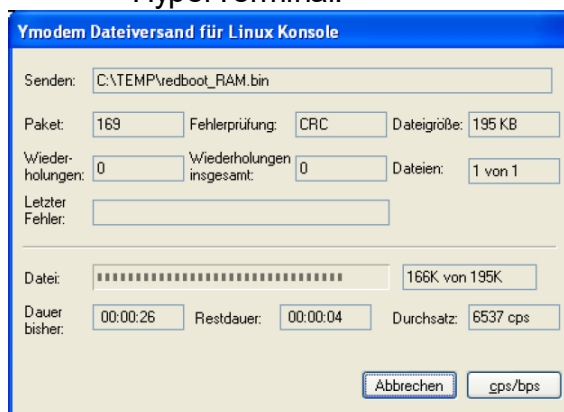
Here is what the complete process looks like on the serial console:

```
++Searching for BBT table in the flash ...

Found version 1 Bbt0 at block 8191 (0x3ffe0000)
Total bad blocks: 0
hardware reset by POR

Clock input is 24 MHz
Hardware: OPUS A3 standard full
Total RAM size: 268435456 byte (256MB)
Chip is working in consumer mode
Using display configuration "Seiko WQVGA"

RedBoot(tm) bootstrap and debug environment [ROMRAM]
Non-certified release, version OPUS A3 1.3.2-3 - built
15:12:00, Mar 25 2015

Platform: OPUS A3 (i.MX35) PASS 2.0 [x32 DDR2]
Copyright (C) 2000, 2001, 2002, 2003, 2004 Red Hat, Inc.
Copyright (C) 2003, 2004, 2005, 2006 eCosCentric Limited

RAM: 0x00000000-0x07f00000, [0x00096618-0x07ed1000] available
FLASH: 0x00000000 - 0x40000000, 8192 blocks of 0x00020000 bytes
each.
== Executing boot script in 0.100 seconds - enter ^C to abort
^C
RedBoot> ^C          STEP 2
RedBoot> ^C
RedBoot>
RedBoot> load -r -b 0x100000 -m ymodem          STEP 3
CCRaw file loaded 0x00100000-0x0013218b, assumed entry at
0x00100000
xyzModem - CRC mode, 0(SOH)/202(STX)/0(CAN) packets, 6 retries
RedBoot> run          STEP 4
load entry_address=0x100000
virt_addr=0x100000
```

```
phys_addr=0x80100000
++Searching for BBT table in the flash ...

Found version 1 Bbt0 at block 8191 (0x3ffe0000)
Total bad blocks: 0
hardware reset by POR

Clock input is 24 MHz
Hardware: OPUS A3 standard full
Total RAM size: 268435456 byte (256MB)
Chip is working in consumer mode
Using display configuration "Seiko WQVGA"

RedBoot(tm) bootstrap and debug environment [ROMRAM]
Non-certified release, version OPUS A3 1.3.2-3 - built
15:12:00, Mar 25 2015

Platform: OPUS A3 (i.MX35) PASS 2.0 [x32 DDR2]
Copyright (C) 2000, 2001, 2002, 2003, 2004 Red Hat, Inc.
Copyright (C) 2003, 2004, 2005, 2006 eCosCentric Limited

RAM: 0x00000000-0x07f00000, [0x00096618-0x07ed1000] available
FLASH: 0x00000000 - 0x40000000, 8192 blocks of 0x00020000 bytes
each.
== Executing boot script in 0.100 seconds - enter ^C to abort
^C
```

RedBoot> ^C   **STEP 5**
RedBoot> ^C
RedBoot>
RedBoot> **romupdate**   **STEP 6**
Updating ROM in NAND flash
RedBoot>

14.4.3.2   Manual update of the service ramdisk

If the update still won't start, you also have to update the service ramdisk.
For this, have the file *zImage.service* ready on your computer and follow the same steps as for the boot loader file up to step 3, where you send the file zimage.service.

The loading of this file takes about 15 to 25 minutes; please be patient and wait until it is finished!

After you've returned to the boot prompt, install the service ramdisk by entering the command

```
fis create service.
```

The line

```
"An image named 'service' exists – continue (y/n)?"
```

is displayed, confirm by entering "y". After that power off the device and install the correct OS again using the method described in chapter 3.

14.4.4　Touch Screen Calibration

Devices with a resistive touch need a calibration after an update of the user application (i.e. when the /opt folder is deleted during the update), or if the touch doesn't work as expected, anymore.

Calibration after Update

After an update that deletes the /opt folder, i.e. after the installation of a OS / user tar file through the service mode, the calibration file is deleted. Because of that the calibration is executed automatically at the first start up of the device.

Calibration because of bad touch performance

If the touch doesn't work well, it might help to re-calibrate the touch screen.
To do that, there are several ways:

1. From the application:

With Projektor/PClient, the calibration can be started using the JavaScript command `startTouchscreenCalibration();`

With CODESYS, the calibration can be started using the Driver module Driver -> System -> TSCalibrate and then in code the command `TSCalibrate.Trigger();`

2. Through deleting the calibration file

When the device starts up it is checked if the calibration file exists. If it doesn't, the calibration program will be executed.
So to force the calibration to be executed at start up, the calibration file can be deleted.
This can be done through the [serial console](#)☐775.
The command to delete the file is: `rm /opt/etc/pointercal`
Afterwards execute the following command to restart the device: `reboot`
After the restart the calibration will be started.

3. Through the serial console

The calibration program can also be executed directly from the serial console, using the command `ts_calibrate`
However, this can be slightly problematic if another GUI application is running, because then both applications, the user application and the calibration program, will paint on the screen and it can be difficult to do the calibration properly.
To work around that, the running application should be stopped.

In case of **PClient**, use the command: `/opt/etc/init.d/02_pclient stop`
Then execute the calibration program: `ts_calibrate`
And once the calibration has finished, restart the
PClient: `/opt/etc/init.d/02_pclient start`

In case of **CODESYS**, use the command: `/opt/CoDeSysControl/Files/run stop`
Then execute the calibration program: `ts_calibrate`
And once the calibration has finished, restart the
PClient: `/opt/CoDeSysControl/Files/run start`

In case of a customer programmed **C++ application**, consult the vendor of the
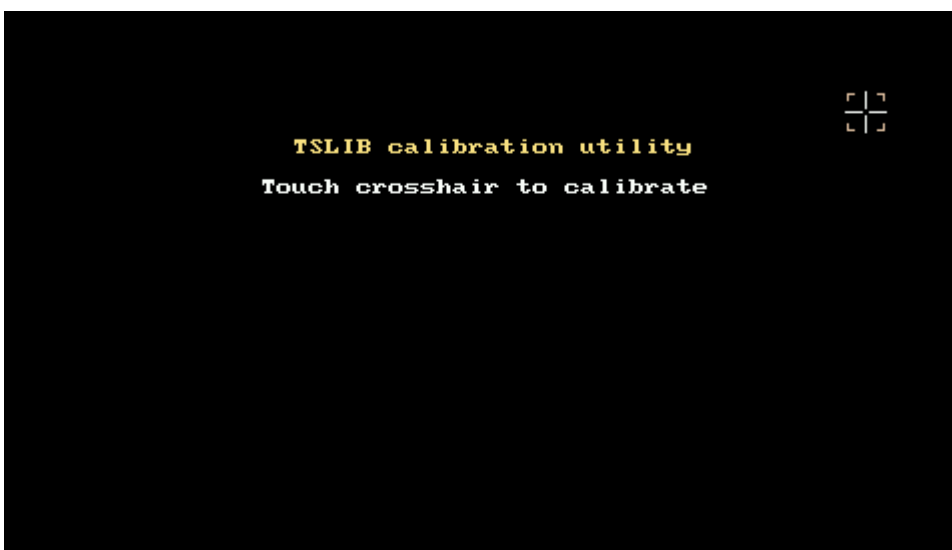vehicle/system for the exact command.

Calibrating the Device

When the calibration is started, this will be the first screen:
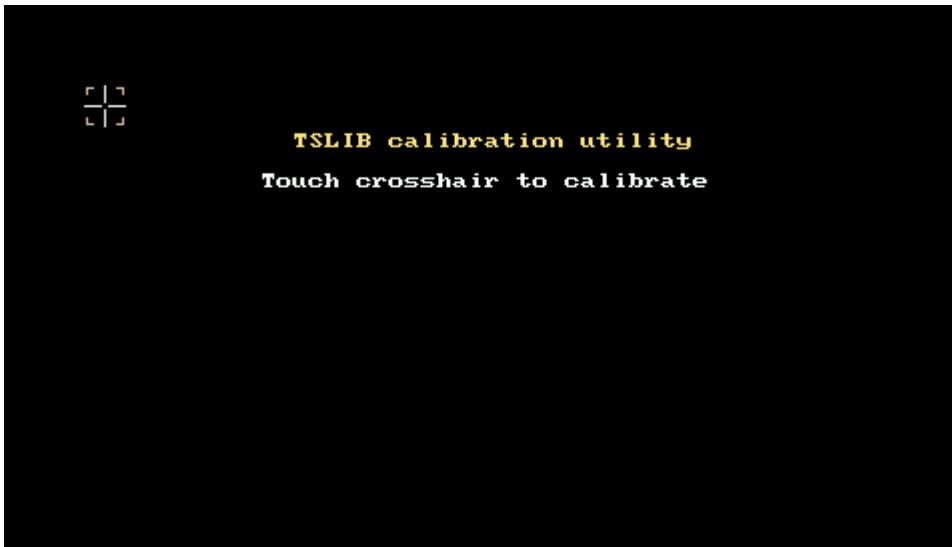
Press the cross hair in the upper left corner once, firmly, with your thumb (NOT with a touch
pen or anything else).
When this is done, the cross hair will be shown in the upper right corner:

Press the cross hair again, one time, with a firm thumb press, and continue like that with the
cross hairs in the lower right, lower left and center.

If the calibration went wrong, i.e. there was a double press and the cross hair "jumped" two positions at once, repeat the calibration in one of the ways described above.

On the serial console, the output while running the calibration, looks like this:

```
root@rosi /$ ts_calibrate
xres = 480, yres = 272
Took 24 samples...
Top left : X = 3593 Y = 3072
Took 34 samples...
Top right : X =  476 Y = 3211
Took 36 samples...
Bot right : X =  542 Y =  657
Took 43 samples...
Bot left : X = 3633 Y =  806
Took 46 samples...
Center : X = 2075 Y = 2098
497.903076 -0.122414 -0.002674
275.258942 0.000120 -0.070859
Calibration constants: 32630576 -8022 -175 18039370 7 -4643
65536
```

Note that xres and yres will differ depending on the device that is calibrated (the example above is from an A3, for A6 it will be 800 and 480, respectively).

Also note that the values for X and Y and the calibration constants will differ for each device depending on each individual touch screen.

Testing the touch screen

There is also a touch screen test program installed with the Linux operating system. This can be started through the serial console with the command ts_test
Note that for this to work best the running application should be stopped as described above[786].
The touch screen test program can be stopped by pressing CTRL+C on the serial console.

Tools

15   Tools

ISO 17215 Camera Configuration[790]

15.1   ISO 17215 Camera Configuration

The ISO 17215 standard provides a standardized way to configure ethernet cameras in an automotive environment.

The ISO 17215 ethernet camera configuration tool is a small desktop application that allows configuration of ethernet cameras that support ISO 17215. It is intended for end of line configuration.
To open the tool, select *Tools* => *Camera* => *ISO 17215 Camera Configuration*.
The tool allows retrieving and setting of some of the standardized mandatory settings and also camera manufacturer specific settings if available.
To be able to change camera settings, the tool has to be licensed. See Licensing chapter below for more details.
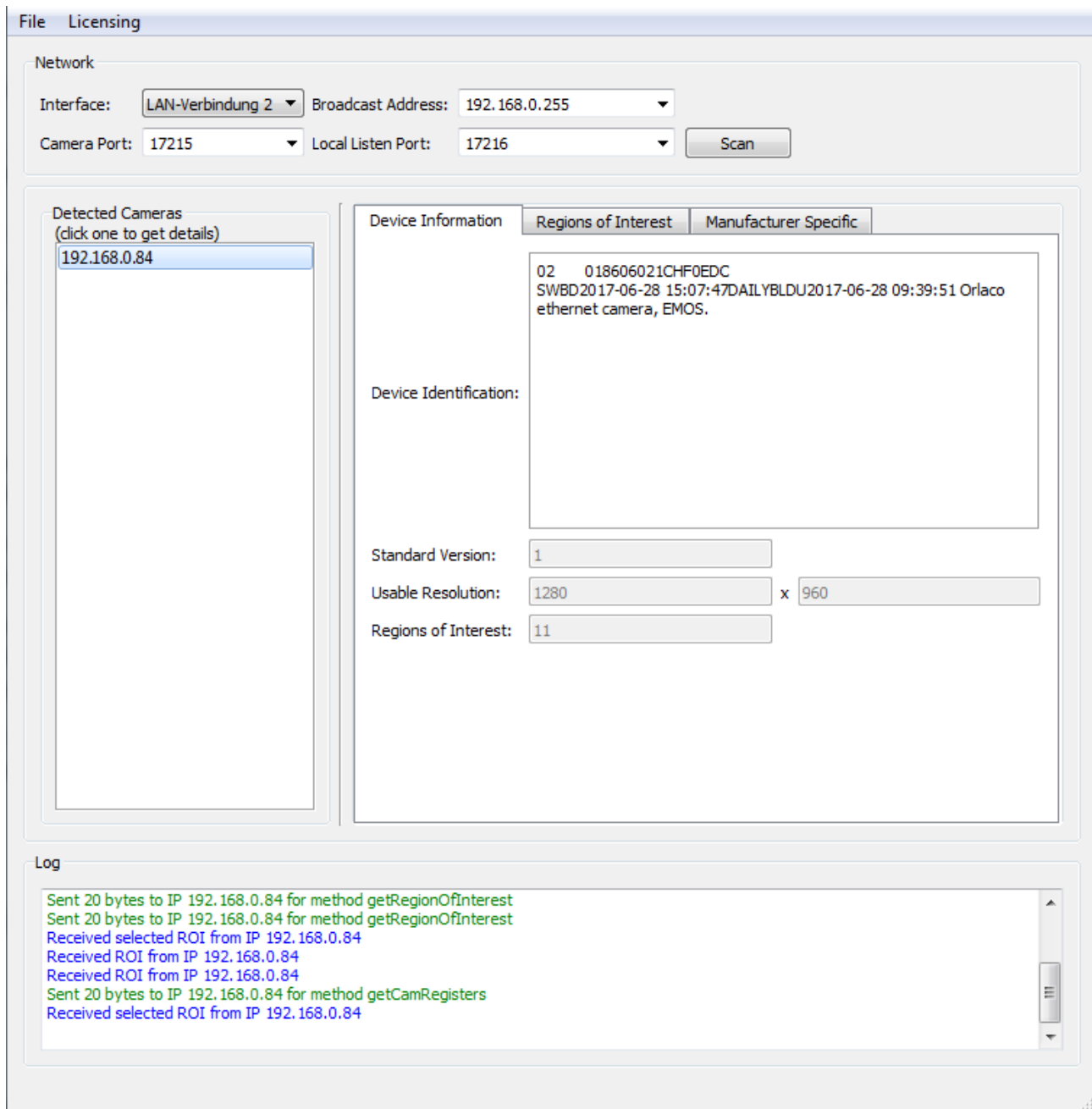
Network settings and scanning

On start up, the tool automatically scans the available ethernet interfaces of the PC and lists them in the *Interfaces* drop down box. It only lists interfaces that are up and running (e.g. are configured and a cable is connected).
Choose the interface to which the camera is connected.

The *Broadcast Address* field will automatically be filled with the matching IPv4 broadcast address of the chosen interface but can be changed manually if needed. Note that the network address of the interface has to be configured so it matches the network of the camera (e.g. if the camera has the address 192.168.135.16, the interface on the PC should also have an address in the 192.168.135.X network (where X needs to be an unused number, typically 1)).

If the IP address of your network interface can't be changed, consider using a second network card / USB-Ethernet interface.

The **Camera Port** field needs to contain the UDP communication port number on which the camera listens. This is not the port for the actual video stream but the one for the ISO 17215 communication. Typical values are 17215 (ISO 17215), 30490 (SOME/IP-SD) or 30391 (SOME/IP).

The **Local Listen Port** field contains the UDP port to which the camera will send its answers (the port on which the PC tool will listen). Typically this is **Camera Port** + 1.

Refer to the manufacturer documentation of the camera to get the matching addresses and port numbers.

When hitting the Scan button, a service discovery broadcast message is sent on the selected network. If one or more camera answer the request, they will be listed in the **Detected Cameras** list.

Camera settings and configuration

When a camera is detected, it will be listed in the **Detected Cameras** list. Select one of the detected IP addresses to select the camera.

The data set of the camera will automatically be requested and the information is presented in the **Device Information** tab.

Device Information

The **Device Identification** typically shows the name and/or type of camera along with manufacturer and hardware software revisions. The content of this field is manufacturer specific.

The **Standard Version** field contains the supported ISO 17215 standard version.

The **Usable Resolution** Fields represent the horizontal and vertical resolution of the camera's sensor.

The **Regions of Interest** field contains the number of available regions of interest that can be configured (see below).

Regions of Interest

A Region of Interest (ROI) defines a setting that consists of
- the region of the camera's sensor that shall be used (Sensor Geometry)
- the output resolution
- video compression
- maximum bit rate that shall be used for the video stream

The **Select Region of Interest** drop down box lets you select one of the available ROIs of the camera. The current setting of each ROI will be downloaded from the camera when changing the selection.

Note: ROI 0 is typically read-only! Select another number to make changes.

To change the camera sensor geometry, two points on the sensor surface can be changed (P1 and P2). The points form a rectangle on the sensor surface where P1 is the top left point of the rectangle and P2 is the lower right point of the rectangle. The orientation of the points can also be changed to create a horizontal and/or vertical mirroring of the camera image (e.g. if P2 is moved to the left and above P1, the image will be horizontally and vertically mirrored). The rectangle defined by P1 and P2 define the resolution of the captured image.

The **Output Resolution** fields define the resolution of the actual video stream. If the sensor resolution is different to the output resolution, the camera will scale the image accordingly. Choose the lowest possible output resolution to save bandwith (e.g. if your

display has a resolution of 800x480, it makes no sense to choose an output resolution of 1280x960).

WARNING: A certain combination of sensor geometry and output resolution may not be supported by the camera. This is a manufacturer specific limit and can not be checked. Please check camera documentation and error output!

In the *Video Compression* drop down box the type of video compression can be selected. Select the type that is supported by the display.

The *Max. Bitrate* field defines the maximum bit rate used by the camera. The lower the value, the higher the compression and the lower the used bandwidth will be:
- Low Bitrate: High compression, worse picture quality, better display performance
- High Bitrate: Low compression, better picture quality, worse display performance
Select a bitrate limit that gives the best compromise in picture quality and performance.

To save the settings for the selected ROI, hit the *Apply the above settings of the selected ROI to Camera* button. If the check box *Subscribe to this ROI* is enabled, the camera will directly use this ROI for video streaming. If the checkmark is not enabled, the ROI settings will be saved but the camera will still stream with the settings of the previously selected ROI (only if another ROI than the currently selected was active).

To make the settings of sensor geometry and output resolution more intuitive, a ROI Configuration Wizard is available. Hit the *ROI Configuration Wizard* button to open it.
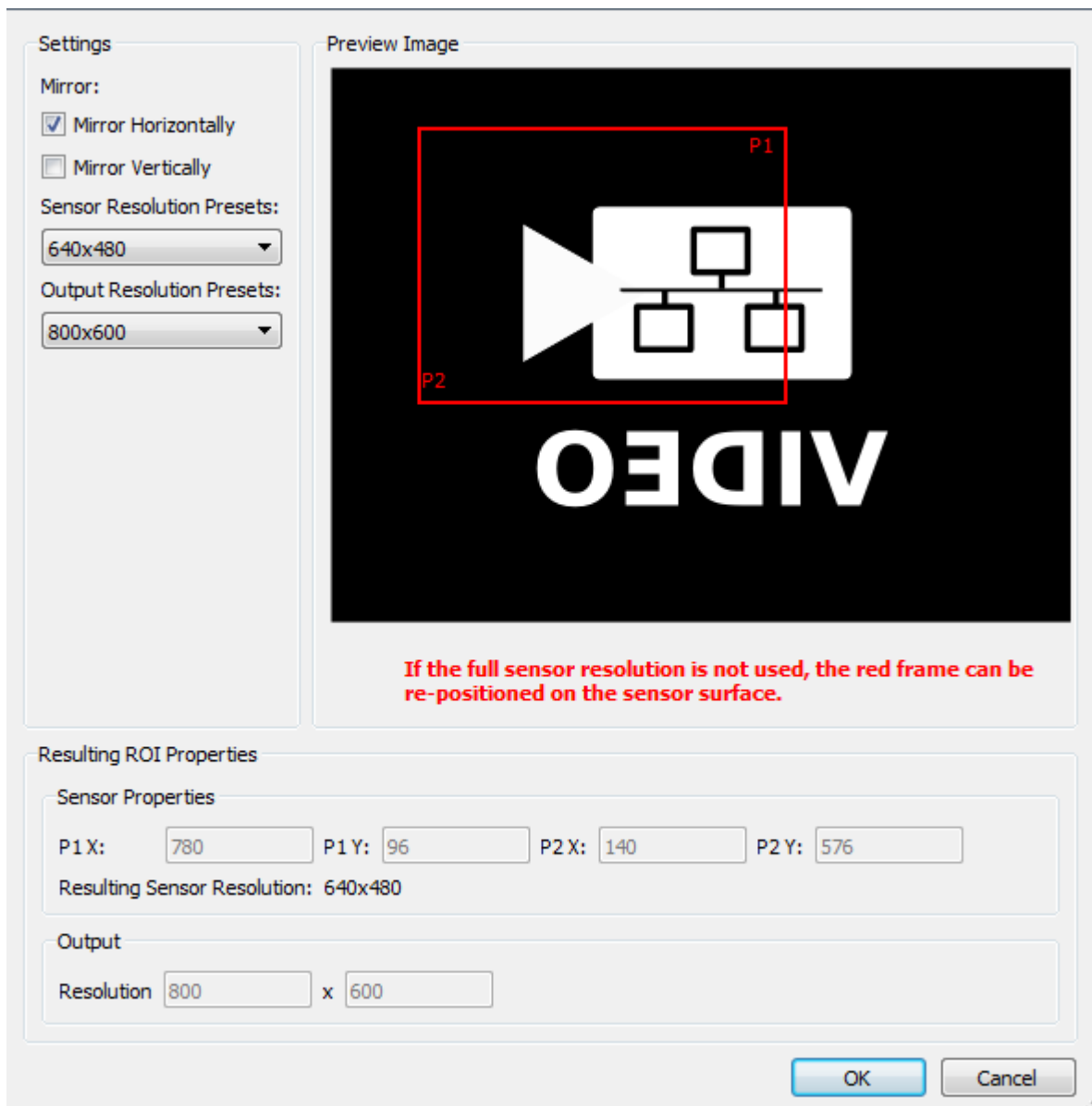
Use the **Mirror Horizontally** (left will be right, right will be left) and **Mirror Vertically** (up will be down, down will be up) check boxes to configure mirroring of the camera. The preview image will change according to the mirror settings.
Use the **Sensor Resolution Presets** drop down box to select a region of the camera sensor.
Use the **Output Resolution Presets** drop down box to select the resolution of the video stream.

If the resolution selected in **Sensor Resolution Presets** does not cover the full sensor area, the red rectangle in the **Image Preview** can be positioned with the mouse to select the desired region of the camera sensor that shall be used.

The resulting ROI properties (P1/P2 and output resolution) are displayed at the bottom.

Note: The sensor and output resolution presets are currently designed to work with an Orlaco Emos camera. Not all presets can be combined with this camera. A warning will be displayed if a resolution combination will not work for that camera. Other cameras might support that combination so it is still possible to apply this to the camera. Note the error output in the *Log* window for error messages.

When pressing the **OK** button, the ROI settings from the wizard will update the settings in the Regions of Interest tab of the tool itself. Press the *Apply...* button to save it to the camera.

Manufacturer Specific

The *Manufacturer Specific* tab contains settings for a certain camera manufacturer. The settings that can be made here are not part of the ISO 17215 standard. Therefore special implementations for every camera manufacturer and even model need to be implemented.
Currently only settings specific to an Orlaco EMOS camera are supported. If manufacturer settings for other brands and models are needed please contact support.

Warning: These settings should only be changed if the meaning of the settings are very clear and match the target / configuration platform. Setting a wrong value here might make the camera unresponsive and not being able to work or be configured after that! Changing settings here is on your own risk. The provider of this tool is not responsible for any harm that a misconfiguration may cause!

Orlaco EMOS

Only change the settings in this tab if an Orlaco EMOS camera is connected and selected. When enabling the *I am sure that this is an Orlaco EMOS camera* check box, the current settings are downloaded from the camera and can be changed. Every setting needs to be changed separately with the matching *Set* button. After the *Set* button was clicked, the field will be cleared and the new setting will be downloaded from the camera. Watch the **Log** section for error messages.
Note that some of the settings will only have an effect after switching the camera off and on again.

Static IP Address

The static IPv4 address of the camera. Change this to an address that matches the target system network. DHCP can also be enabled.

Static Network Mask

The static IPv4 network mask of the camera. Change this to a mask that matches the target system network. DHCP can also be enabled.

MAC Address

The MAC address of the camera. This setting is read-only and can not be changed.

DHCP

Select if DHCP shall be used or not. If DHCP is enabled, the camera will only work if a DHCP server is available in the network. If DHCP is disabled, the static IP address and mask will be used.

DHCP Host Name

It is not 100% clear what this field means. Therefore we don't allow to change anything here. Maybe this will contain the DHCP host name if the camera was configured by a DHCP server?

RTP Stream Destination Port

To this UDP port the actual video stream will be sent.

RTP Stream Destination IP Address

In this field either a specific destination IP address or a broadcast address (e.g. 255.255.255.255) can be entered. If a specific address was set, only the display with that address can receive the video stream.

RTP Stream Destination MAC Address

In this field either a specific destination MAC address or a broadcast address (e.g. ff:ff:ff:ff:ff:ff) can be entered.
If a specific MAC address was set, only the display with that address can receive the video stream. This can be used if an intelligent switch is used in the network. This switch then routes the video stream packets only to the port where the display with the specified MAC is connected.
If a broadcast MAC address is entered, the ethernet packages will be sent to every participant in the network.

RTP Stream Source Port

The UDP source port of the video stream. Can be same or different as/to **RTP Stream Destination Port**.

HDR

Enable or disable the HDR (High Dynamic Range) setting of the camera. If enabled the camera can produce pictures with different light conditions better.

Overlay

If enabled, two rotating arcs are overlaid to the video stream. This helps the user to detect video freezes and is a safety feature.

LED Mode

LED Mode can be either **enabled**, **disabled** or set to **auto**. The effect of this setting is currently unknown.

Stream Protocol

Select **RTP** or **AVB** for setting the stream protocol. Many displays only support RTP.

Flash Mode

This property can be set to **normal**, **flash** or **flashed**. The purpose of this is unknown.

VLAN ID

Here the ID (tag) of a VLAN can be set. Set to 0 to disable VLAN tagging. Any other value will tag the ethernet stream packages with the configured VALN ID.

Stream ID

Here the name or ID of the stream can be set. This is for identification purposes only (e.g. set the name of the camera to be able to identify it later in the vehicle: frontCAM, backR, backL, etc.). Max. 8 characters.

UDP Communication Port

The UDP port for doing ISO 17215 communication with.

Sensor Temperature (°C)

The current temperature of the camera sensor. Read only.

Board Temperature (°C)

The current temperature of the camera board. Read only.

LED Temperature (°C)

The current temperature of the camera LED. Read only.

LED Percentage

The current camera LED brightness in %. Read only.

Selected ROI

The currently selected ROI. This can also be changed in the Regions of Interest tab.

No Stream at Boot

Select *stream* or *no stream*. If stream is selected, the camera will begin streaming the video directly after connecting it to power. If no stream is selected, the camera will only stream after someone subscribed to a ROI.

Wait for MAC:

do not wait or *wait* can be selected. It is currently unknown what this setting means. This may make the camera unresponsive for any configuration messages. This setting should **NOT** be set to **wait** currently.

Wait for PTP sync:

do not wait or *wait* can be selected. It is currently unknown what this setting means. If wait is selected, the camera will wait for a PTP (precision time protocol) message on startup before it will do something else. If this PTP mechanism is not supported on the system, this makes the camera unresponsive for any configuration messages. This setting should **NOT** be set to **wait** currently.

Log

Check the Log window for message with the tool and the camera. Error messages are shown in red color. Reception messages are shown in blue. Sent messages in green.

Licensing

If the program is not licensed, settings from cameras can be read but no changes can be sent to the camera.
Please contact your vendor to obtain a license for the tool.
If a license was already obtained and the tool is started from within the program, the existing license file will automatically be provided to the tool.
If this tool shall be used as a stand-alone version, existing customers may also send a license request to the vendor. This request needs to contain the key file which can be generated via **Licensing** => **Generate Key File**. The vendor will return a license file which can be loaded via **Licensing** => **Load License File**. The location of the license file will be remembered and only needs to be done once (or whenever the license file is moved on the file system).