Manual

08/22 MN048018EN

GALILEO User Help

Powering Business Worldwide

Service

For service and support please contact your local sales organization:

Contact details:	Eaton.com/contact
Service-page:	Eaton.com/aftersales

Original Operating Instructions

is the German-language edition of this document

Publication date

08/2022 Version 10.6.0

Authors: Electrical Sector, Business Unit Industrial Automation
Patrik Dudler, Jürgen Wagner
Editor: Eaton Industries GmbH, Hein-Moeller-Straße 7-11, D-53115 Bonn
Antje Nonnen

Table of Contents

0.1 About this manual

This manual explains how to use the Automatization Software GALILEO program
In order to be able to use this program, you must first be familiar with how to design and configure projects using software.

Please send any comments, recommendations, or suggestions regarding this document to: AfterSalesEGBonn@eaton.com

0.1.1 List of revisions

For information on new topics, deleted topics, and changes in comparison to earlier versions, please refer to the release notes.

Publication date	Keyword
09/2014	Beta version for preview 1
11/2015	New edition 10.0
11/2016	Update to software version 10.1
05/2017	Update to software version 10.2
03/2018	Update to software version 10.3
06/2019	Update to software version 10.4
10/2019	Update to software version 10.4.1
05/2020	Update to software version 10.4.5
04/2021	Software version 10.5
08/2022	Software version 10.6

0.1.2 Target group

This GALILEO User Help is intended for electricians and electrical engineers, as well as for the people who will be in charge of performing the electrical installation and people who will be using the touch panel as an operating and monitoring device or as an integrated operating and control device in their own applications.

This manual is an integral part of GALILEO and can be accessed at any time from the Automatization Software program by using the Help function, provided that the Help files have been installed.

0.1 About this manual

0.1.3 Disclaimer

All the information in this GALILEO User Help has been prepared to the best of our knowledge and in accordance with the state of the art. However, this does not exclude the possibility of there being errors or inaccuracies. We assume no liability for the correctness and completeness of this information. In particular, this information does not guarantee any particular properties.

Do not use GALILEO before reading and understanding this GALILEO User Help.

It is assumed that the user of this document is thoroughly familiar with the information found in the manuals for incorporating GALILEO into automation processes.

Hazards posed by the touch panel cannot be eliminated if the safety instructions are not observed – especially if the GALILEO is installed and commissioned by unqualified personnel and/or Automatization Software is used improperly. Eaton assumes no liability for any damages resulting from cases such as these.

0.1.4 Device designations and abbreviations

The following general terms are used throughout this manual:

Short designation	**Explanation**
GALILEO User Help	Function name
Automatization Software	Family
GALILEO	Product name
DemoApplication2.prj	Demo example based on the XV300 touch panel
demo.prj	Additional demo examples

For the exact name and version of your Automatization Software, please refer to the GALILEO section, which can be accessed by clicking on Help under the 📖▾ tab.

0.1.5 Writing conventions

Table 1: Formatting used in this manual

Award	Description
Bold text	Used for all graphical user interface elements
`Monospaced` Font format code	Used for all elements at the file level, source code command lines
Text	Used for the button labels
Menu path\submenu\...\item	Command sequences, paths, dialog boxes, and menu pages in the software
<name>	Angle brackets are used to indicate variable values that you must replace with your own values
Description	Label in GALILEO

Warning labels

Risk of personal injury warning.

⚠ **DANGER**

Warns of hazardous situations that result in serious injury or death.

⚠ **WARNING**

Warns of the possibility of hazardous situations that could result in serious injury or even death.

⚠ **DANGER!**

Dangerous Electrical Voltage!

⚠ **CAUTION**

Warns of the possibility of hazardous situations that can cause injury.

Property damage warning

CAUTION

Warns about the possibility of material damage.

Notes

→ Indicates useful tips.

▶ Indicates instructions to be followed

i+ Additional information, background information,
information worth knowing, useful additional information

0.1 About this manual

Additional information for use

Documents (such as manuals) are listed after the PDF icon together with the corresponding name and Eaton number.

PDF Publication title For identifying the Eaton publication code

External Internet addresses. They will be shown after the 🌐 icon.

🌐 Destination address

1. General

1.1 What is GALILEO

Are you ready to redefine the way you control your devices?

With the release of its GALILEO Automatization Software for visualization, Eaton is redefining what it means for software to be user-friendly and for programming to be efficient!

Create projects more quickly, efficiently, and securely: that's the slogan!
And the restructured GALILEO User Help is designed to help you along the way.

With a wide variety of new features, GALILEO is revolutionizing both controls and operation for the entire lifecycle of both machines and systems. In other words: from engineering, through commissioning and operation, all the way to maintenance! Featuring design styles, animated transitions, and gestures such as swiping, scrolling, and pinch-to-zoom, GALILEO is the ideal companion to the advantages provided by our new XV300 and XP500 series of capacitive devices.

It goes without saying that GALILEO can also be used with devices from the tried-and-true XV100 (all color devices) and XV(S)400 series (all devices with 65k colors). And thanks to the full forward compatibility of all your projects, you can rest assured knowing your investment will yield dividends for a long time.

For information on migrating existing GALILEO 8.1 projects to GALILEO, as well as on supported protocols, please refer to the Galileo Migration Guide PDF document that was stored in the selected folder on your computer when GALILEO was installed.

Galileo Migration Guide.pdf
C:\Program Files\Eaton\Galileo 10\doc (default installation path)

The Galileo Migration Guide provides help when porting projects from GALILEO 8.1 to GALILEO. GALILEO will automatically perform and configure many of the required conversions and applications during the import process. However, you may need to make certain changes by hand in certain cases. The Galileo Migration Guide is designed to provide assistance in these cases.

For more information, please consult the corresponding product page on the Internet.

Automatization Software GALILEO
Eaton.com/GALILEO

1.1.1 How GALILEO works

GALILEO is Automatization Software that makes it possible to design and create application-specific man-machine interfaces. These interfaces are provided to device users so that they can view the relevant machine's data in order to be able to make decisions. This, in turn, enables device users to enter input and control the machine accordingly.

To this end, the application from the Automatization Software communicates with the machine's PLC.

The interface is created when a project is built using the visualization software. This project is loaded onto the device or, more specifically, the touch panel. The program is then started on the touch panel and will provide the device user with the PLC's data in the form of values or characters and vice versa.
Values and characters can also be entered on the device and sent to the PLC.

The GALILEO Runtime System, or GRS for short, is the sequential program that is installed and run on your touch panel. These programs interpret the data structure (the compiled result) generated by the Automatization Software GALILEO program and display it in the Galileo Simulator.

Input from device users is obtained exclusively by means of touch gestures on the visualization objects shown on the device screen. In addition, input can be entered with a mouse when using a computer.

Data transfers to and from the PLC in the form of communication with the machine controller are implemented with integrated field bus interfaces, communication modules, serial ports, or Ethernet ports, depending on the specific device model being used.

The basis behind each project consists of a series of screens created by the GALILEO user with the intention of allowing device operators to use these screens and switch between them as necessary. One of these screens is set as the start screen, but every single screen provides information on the machine's data by displaying it in what are referred to as visualization objects. The GALILEO user can define which visualization objects should be displayed on which screens, as well as which input options will be available on which screens, when creating a project.

Both static and dynamic objects can be used in every project. Static objects are located in the screen's background and do not change at runtime. In contrast, dynamic visualization objects are located in a screen's foreground, and their appearance can change as a result of data changes.

1.2 Building blocks in GALILEO applications

Applications in GALILEO are always made up of the three following building blocks: screens, visualization objects, and tags.

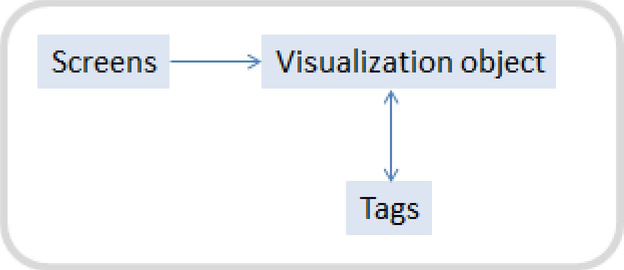


Figure 1: Overview of application building blocks

What is a screen?

Screens are displayed on and fill out the device screen at runtime. Since, however, it is not possible to fit all content on a single screen, content is distributed among multiple screens. A screen change visualization object can be used to switch between these screens – this object is accordingly placed on the various screens when an application is created. The following section provides a brief overview of the various types of screens.

What is a visualization object?

At runtime, device operators can switch between screens by tapping in a touch zone defined for a screen change visualization object on the device screen. This example already illustrates the advantage provided by visualization objects, which are a tool provided by GALILEO for conveniently displaying content. Visualization objects are placed on screens. Objects with content that does not change, such as a welcome text, are referred to as static visualization objects. Meanwhile, objects with content that can change, such as screen changes and tag values for a PLC, are referred to as dynamic visualization objects. It is also important to point out that there is a large variety of visualization objects available for displaying tags. To get started with the subject, please refer to → section "Controls for visualization", page 354.

The Tags

The tags used in the application are used to represent machine tags and the PLC's process image table. Another way to put it is by saying that tags are whatever needs to be displayed, entered, controlled, saved, or archived with the application. To understand more about how tags are used, start by reading about the subject of data communications, then refer to the explanations in → section "Working with tags", page 539.

1.2.1 What's new

Touch gesture controls

Featuring design styles, animated image transitions, and touch gestures such as flicking, dragging, panning, and pinch-to-zoom,
GALILEO is the ideal companion to the advantages provided by our new series of touch panel devices.

The aforementioned gestures make it possible to quickly display a large number of visualization objects of the same type in an easily manageable manner. In addition to this, individual screens that can be larger than the actual physical screen can be scrolled through, moved, and zoomed in and out of as necessary inside a viewing window, much like the way you can navigate websites on a browser.

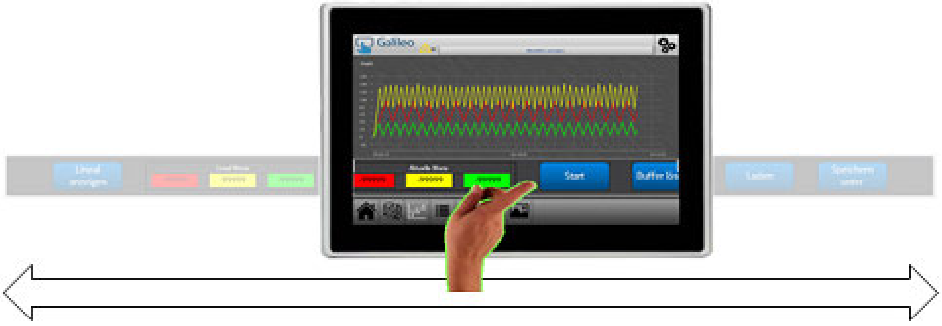


Figure 2: Viewing window

Support for vector graphics

Using vector graphics makes it possible to scale images as necessary without any loss of quality and display them on touch panels; please refer to → section " Image", page 439.

Animations

Animated touch panelGALILEO OPENscreen changes can be configured for XV300 and XP500 s, as well as for .

Just like in a Microsoft PowerPoint presentation, the transitions between screens can be animated in a number of ways. For instance, a new screen can "fly in" from above or from the side.

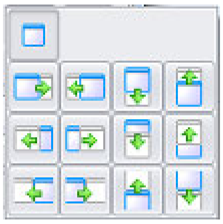


Figure 3: Available animation selections as a property for the "screen change" dynamic object

Web visualization

By enabling the GALILEO Webvisu function, you can use remote devices such as PCs, tablets, and smartphones in order to conveniently access XV300 visualizations. The function is based on the HTML5 standard and supports all common web browsers (Safari, Chrome); please refer to Web visualization, and Creating a visualization for devices.

1.2.2 Intended use

GALILEO has been developed in order to make it possible to visualize parameters ontouch panels,
and can also be used as a computer-based application together with GALILEO OPEN.

Important notes and instructions concerning safety and operation

You must be familiar with and understand how to use Eaton Automatization Software.

Hardware, software, operating systems, and drivers may only be used for the described use cases and only together with the components recommended by Eaton.

Visualization projects must be configured and created exclusively by specialists who are familiar with the system at the location of use.

> **DANGER**
> as a result of unintended actuation!
> When a GALILEO project is transferred to a controller, it will have an immediate effect on the controller's ongoing operation.
> This may result in errors in the process being controlled. The safety mechanisms being used on site must not be disabled as a result of a project being transferred.

1.2.3 Brands and trademarks

GALILEO is the name of the generation and Automatization Software used to create and manage projects.

In order to archive and restore projects in ZIP format, it uses the "DotNetZip" component, which is licensed under the MS-PL license.

In order to display SVG, it uses parts of the "SVG.NET" component, which is licensed under the MS-PL license.

For FTP communications, it uses parts of the "System.Net.FtpClient" component, which is licensed under the MIT License.

For logging, it uses the "log4net" component, which is licensed under the Apache License.

For cryptography purposes, it uses the "Bouncy Castle" library, which is licensed under the MIT License.

WebVisu uses the "libwebsockets" library, which is licensed under the LGPL2.1 license.

1.3 Installing GALILEO

Prerequisites

- An Automatization Software version
- A PC that meets all system requirements and for which the person installing the software has administrator privileges
- A License Keys

To install GALILEO successfully, you will need both a PC that meets all system requirements and a License Keys.

→ If a valid License Keys is not provided during installation, the **DEMO** key GALILEOwill make GALILEO be installed as a demo version.

You can enter the license key for the DEMO version later on by opening Application Configuration License. Functions on main tabs

Projects built with the demo version will only be executable for a limited time after they are transferred onto a device.
When the device is started, a prompt will appear in order to let the user know that demo mode will run with a limited duration of 60 min.

System requirements

Hardware:

1. PC with 1-GHz processor
2. VGA graphics adapter and color monitor with a resolution of 1024 x 768 pixels
3. Min. 1 GB of free space on the hard drive

Software:

One of the following operating systems:

- Windows 10 (32-bit/64-bit): 1 GB RAM recommended
- Windows 11 (32-bit/64-bit): 1 GB RAM recommended

→ Please note that GALILEO versions 10.3 and higher do not support Windows XP!
Support for Windows 7 and Windows 8 is no longer available as of version 10.6.

To be able to take full advantage of the GALILEO User Help, you will need a browser, e.g., Internet Explorer 10 or higher.

→ Before starting with the installation:
Disable the antivirus software on your computer and
close all open applications.

In order to be able to take full advantage of the online documentation's functionality, your computer's download manager must be configured in such a way that clicking on links will automatically open the corresponding Internet pages and PDF documents via FTP.

→ To install GALILEO, you must be logged on as the administrator for the operating system.

1.3.1 Updates

Once the GALILEO program is installed with a valid license key, you will be able to download the latest version from the Eaton Software Download Center

Eaton.com/software

the new version will remain licensed.

During the installation process, the installer will ask whether you want to check for newer versions.

If you do, the installer will check online to see if there are any newer versions available.

→ Earlier versions will be overwritten during installation by default.

Online update check

You can use the Application Options dialog box to configure the automatic update check that runs when GALILEO starts.

When working offline, or when an Internet connection cannot be established, a prompt to this effect will be shown.

Previous GALILEO versions

Galileo 8 will not be updated to GALILEO automatically.

▶ Download the full version of the Automatization SoftwareGALILEO program from the Software Download Center

Download Center - Software
Eaton.com/software

During the installation process, enter your valid License Keys for the Galileo 8 version.

See also

→ section "Messages from GALILEO", page 1145
→ section "Application Configuration", page 135

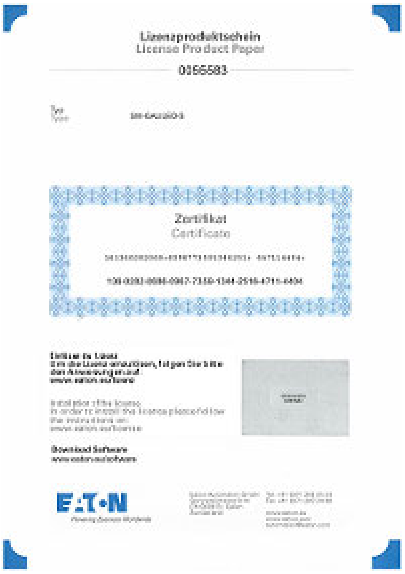
1.3.2 LicensingAutomatization Software GALILEO

During installation, you will be asked for a License Keys for your copy of Automatization Software GALILEO.

When you buy a copy of Automatization Software GALILEO, you will automatically acquire a license product certificate as well.

Your license product certificate will be printed on the certificate .

You can use this certificate to redeem your License Keys.



Lizenzproduktschein
License Product Paper
0156583

Zertifikat
Certificate

[illegible]

Download Software

Figure 4: License product certificate for a single-user license

1.3.2.1 License Keys

The License Keys will be tied to the computer and cannot be transferred to a different computer.

Updates und Hardware Changes

Once the GALILEO program is installed with a valid license key, you will be able to download the latest version from the Eaton Software Download Center

Eaton.com/software

the new version will remain licensed.

After a hardware change, make sure to use your certificate and redeem it again.

To get a certificate with your License Keys, follow the instructions at:

Eaton.com/license

Eaton.com/lizenz

After entering the certificate , you can enter the certificate owner and the e-mail address to which the License Keys should be delivered for your own security.

The License Keys will be sent to you by e-mail.

If you do not receive the code, you can simply repeat the process with the same certificate .

The e-mail will contain the following information:

License type: SW-GALILEO-...S for a single-user license or -M for a multi-user license

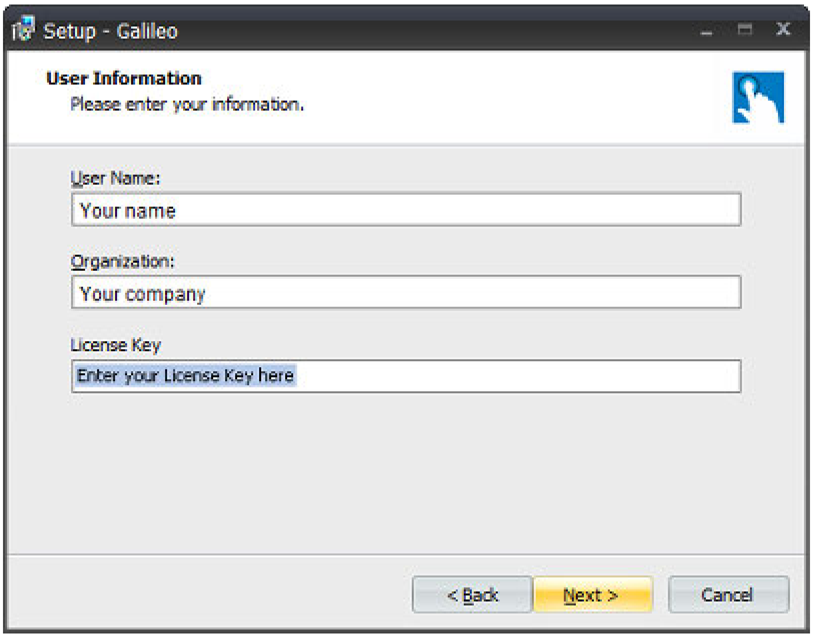
license product certificate Number: A seven-digit number of certificate , e.g., 0055583

License Keys: generated code, e.g. BELSVGK_XXY2B6MFDVXWTRQV

The owner's registration

You can enter the License Keys during installation.

setup.exe/Galileo Setup Wizard



User Information

Please enter your information.

User Name:

Your name

Organization:

Your company

License Key

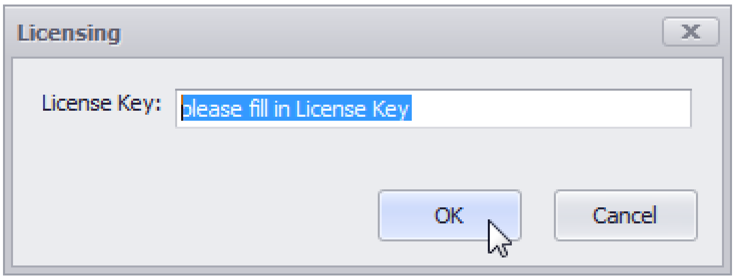
Enter your License Key here

Figure 5: Galileo Setup Wizard, user information dialog box

→ If a valid License Keys is not provided during installation, the **DEMO** key GALILEOwill make GALILEO be installed as a demo version.

If you want, you can skip entering your license key at this point and enter it later on instead by opening the *Home tab and clicking on* 🔑 *License* in the Application Configuration group.

Home\Application Configuration\ License



Licensing

License Key: please fill in License Key

Figure 6: Dialog box for entering the license key

1.3.3 Installation instructions

You will need administrator privileges in order to install the program – simple user rights will not be enough!
The installer program needs administrator privileges in order to be able to correctly install the required system components on your system.

If an error occurs during installation and the error is related to the installation routine not being able to copy or create files or folders, check to make sure that your user account has administrator privileges.

▶ Download the full version of the Automatization Software GALILEO program from the Software Download Center

Download Center - Software
Eaton.com/software

▶ Select the GALILEO product group and the product version you want.
▶ Click on the product version you want in order to download it.

▶ Save the Setup.exe file on your computer.
▶ Run the Setup.exe file on your computer.

▶ Follow the instructions provided by the GALILEO installer on the screen.

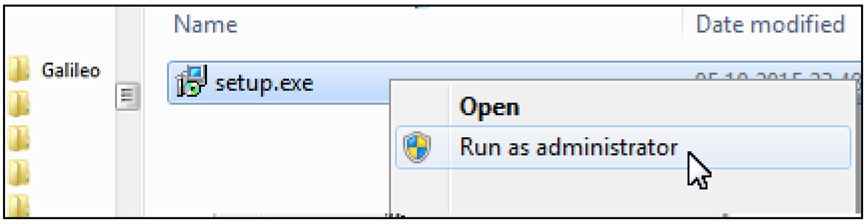
Make sure to have your License Keys ready.

→ If a valid License Keys is not provided during installation, the **DEMO** key GALILEOwill make GALILEO be installed as a demo version.

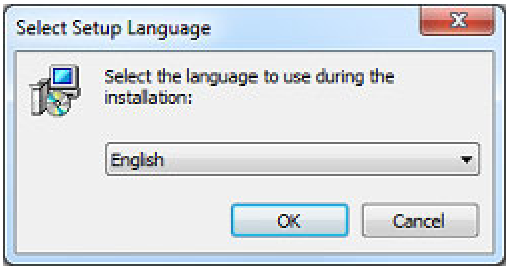
You can enter the license key for the DEMO version later on by opening Application Configuration License. Functions on main tabs
GALILEO installer

Step 1

→ To install GALILEO, you must be logged on as the administrator for the operating system.



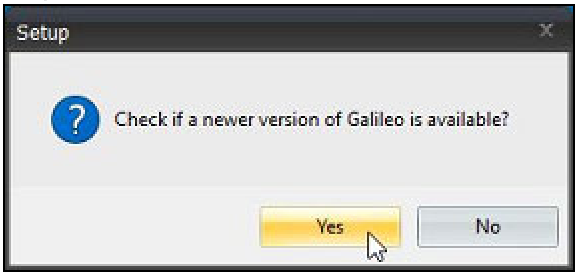
Step 2


Select Setup Language

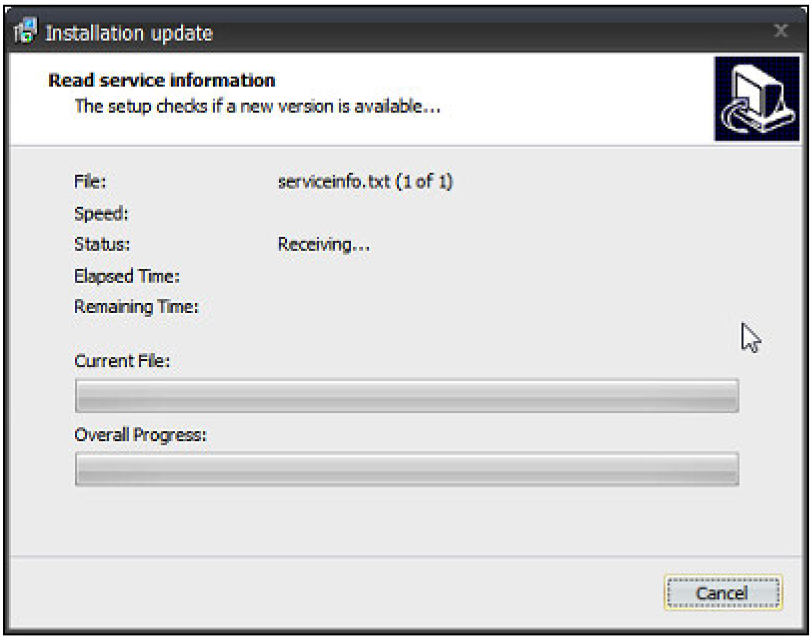
Select the language to use during the installation:

English

Step 3


Setup

Check if a newer version of Galileo is available?


Installation update

Read service information

The setup checks if a new version is available...

File: serviceinfo.txt (1 of 1)

Speed:

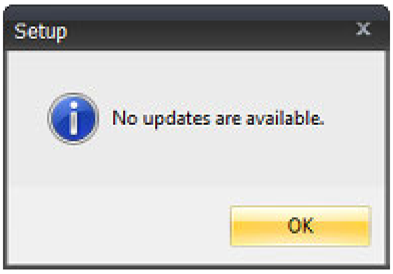
Status: Receiving...

Elapsed Time:

Remaining Time:

Current File:

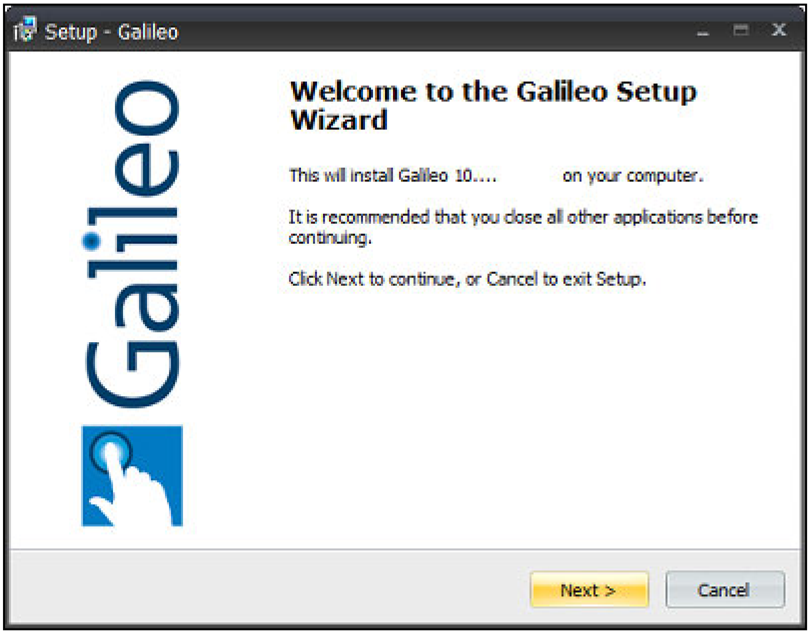
Overall Progress:



No updates are available.

If a newer version is available, it will be installed as an update.

Step 4

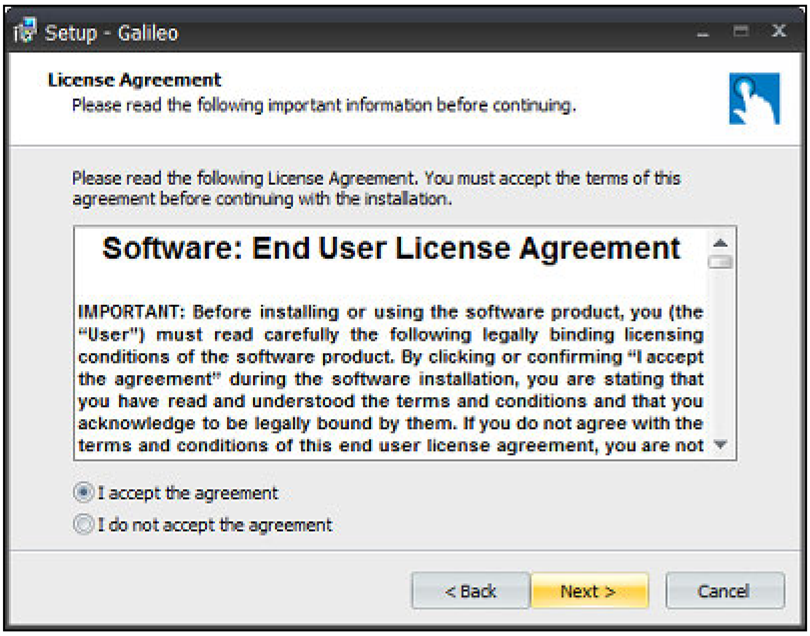


Welcome to the Galileo Setup Wizard

This will install Galileo 10.... on your computer.

It is recommended that you close all other applications before continuing.

Click Next to continue, or Cancel to exit Setup.



License Agreement

Please read the following important information before continuing.

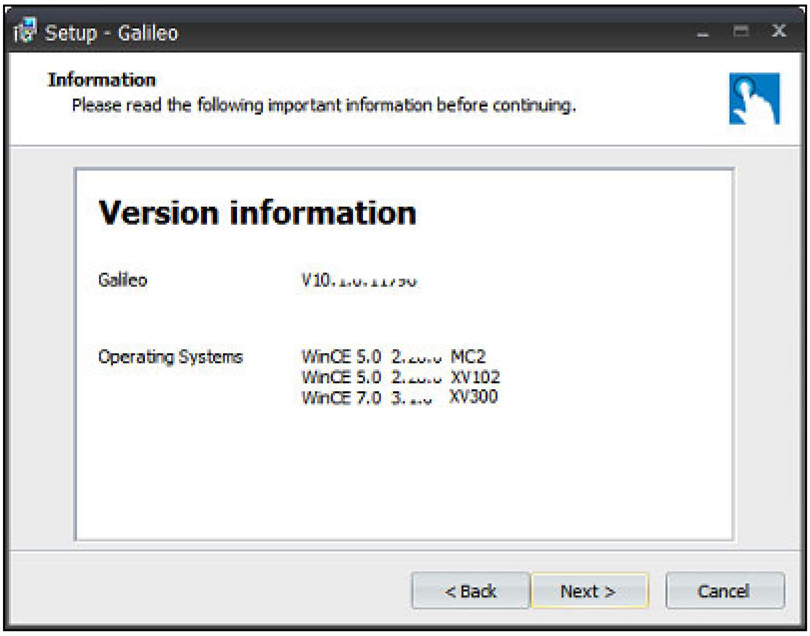
Please read the following License Agreement. You must accept the terms of this agreement before continuing with the installation.

Software: End User License Agreement

IMPORTANT: Before installing or using the software product, you (the "User") must read carefully the following legally binding licensing conditions of the software product. By clicking or confirming "I accept the agreement" during the software installation, you are stating that you have read and understood the terms and conditions and that you acknowledge to be legally bound by them. If you do not agree with the terms and conditions of this end user license agreement, you are not

● I accept the agreement

○ I do not accept the agreement


Setup - Galileo
Information
Please read the following important information before continuing.
< Back
Next >
Cancel

Version information

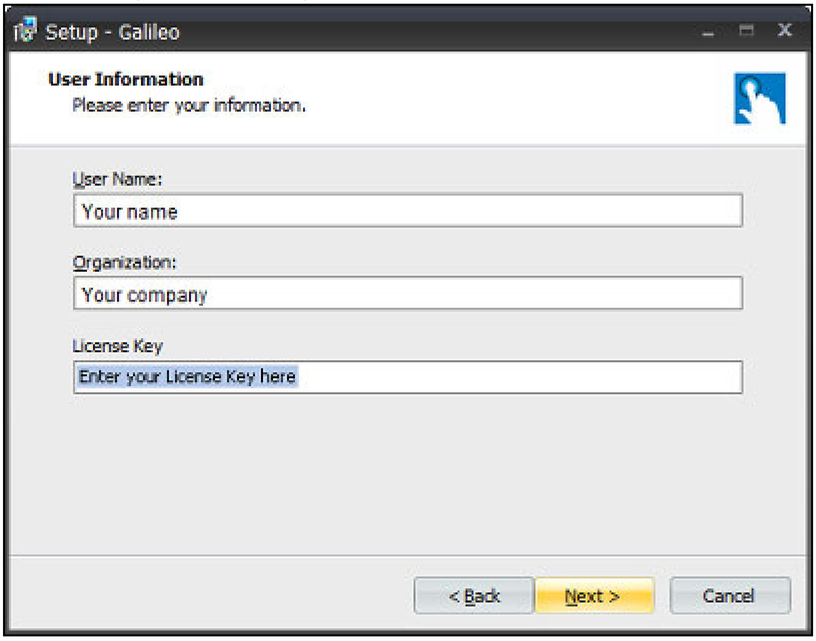
Galileo V10.1.0.11750

Operating Systems WinCE 5.0 2.20.0 MC2
WinCE 5.0 2.20.0 XV102
WinCE 7.0 3.1.0 XV300

If you are performing an update, the installer will ask you whether you want to overwrite the existing software version.

Overwriting the software version will not affect the projects you have already created.

Now enter your GALILEO personal information.

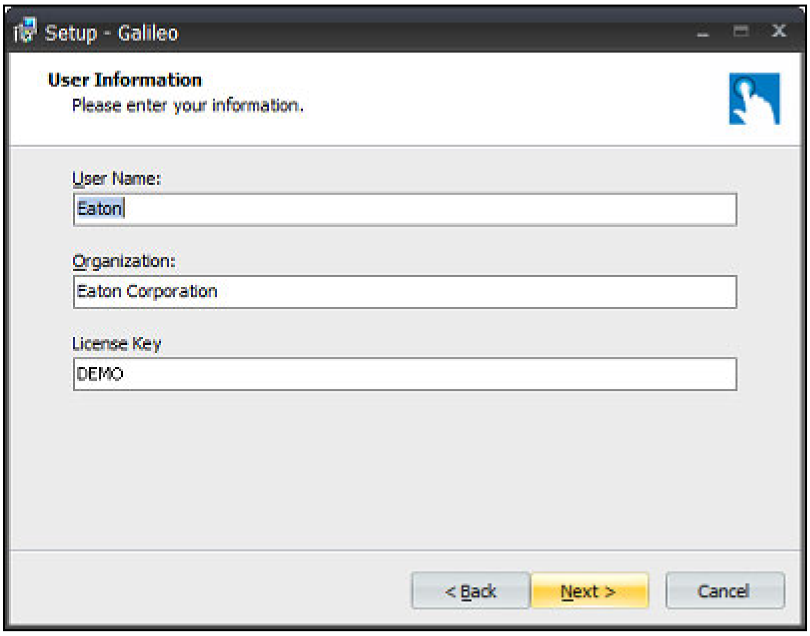


User Name: Your name

Organization: Your company

License Key Enter your License Key here

→ If a valid License Keys is not provided during installation, the **DEMO** key GALILEOwill make GALILEO be installed as a demo version.



Setup - Galileo

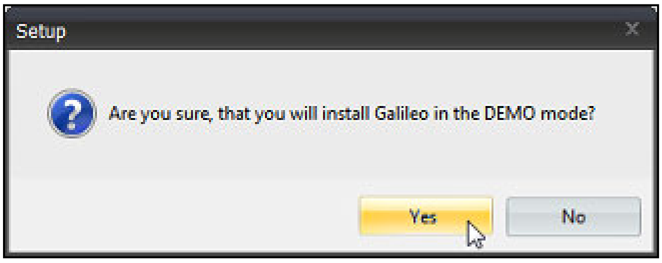
User Information
Please enter your information.

User Name:
Eaton

Organization:
Eaton Corporation

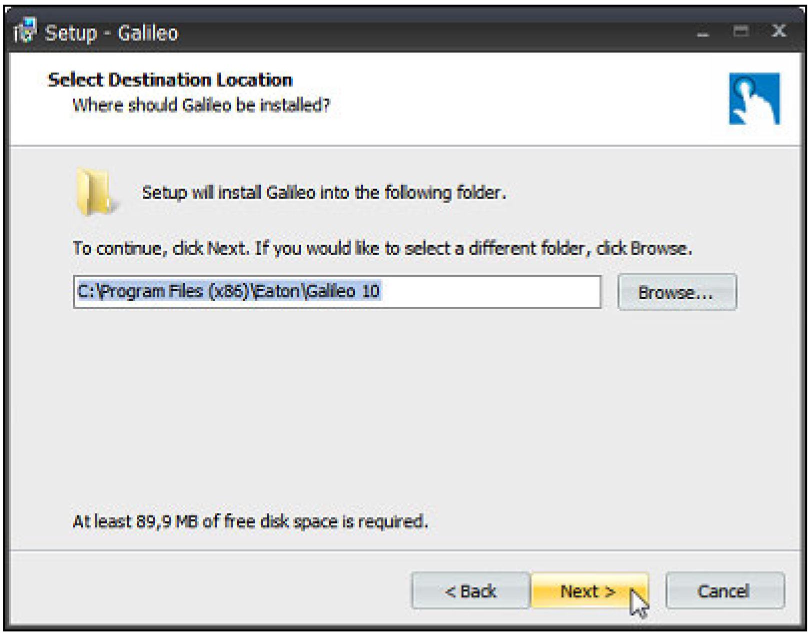
License Key
DEMO

You can enter the license key for the DEMO version later on by opening Application Configuration 🔑 License. Functions on main tabs



Setup

Are you sure, that you will install Galileo in the DEMO mode?



Setup - Galileo

Select Destination Location
Where should Galileo be installed?

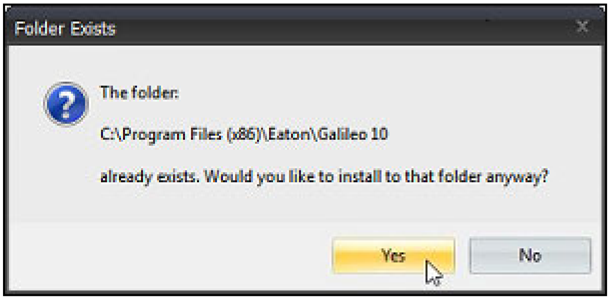
Setup will install Galileo into the following folder.

To continue, click Next. If you would like to select a different folder, click Browse.

C:\Program Files (x86)\Eaton\Galileo 10

At least 89,9 MB of free disk space is required.

When updating:

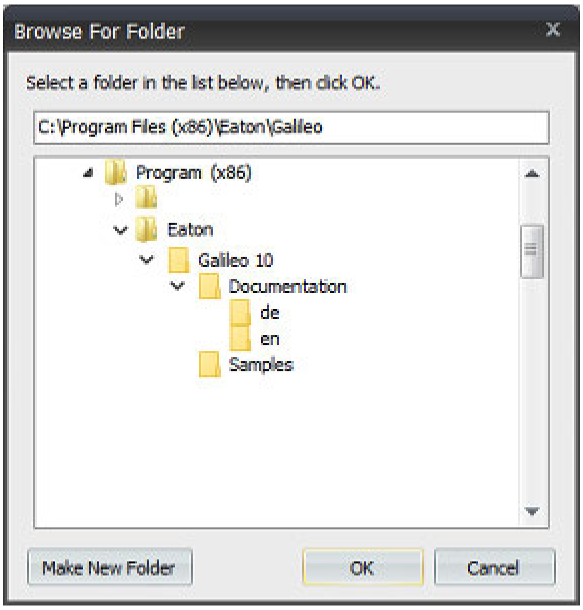

Folder Exists

The folder:

C:\Program Files (x86)\Eaton\Galileo 10

already exists. Would you like to install to that folder anyway?

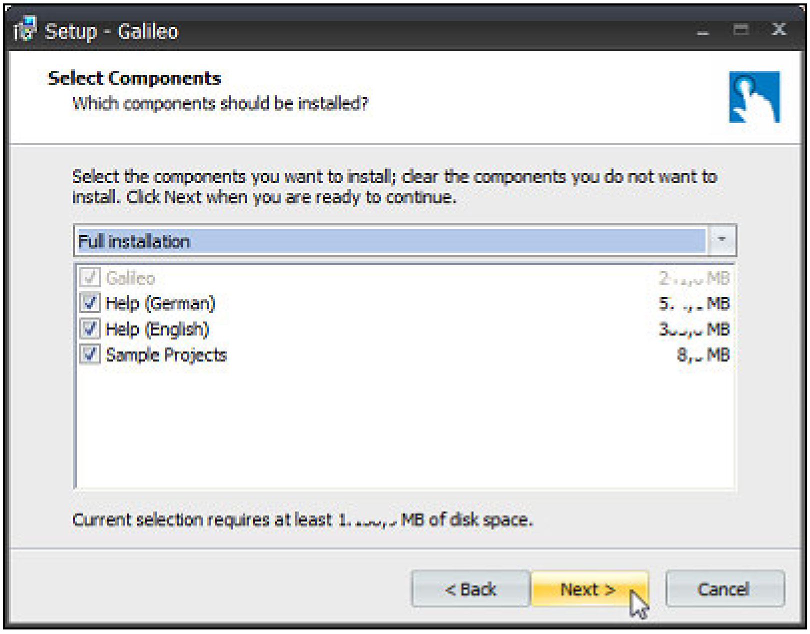
→ Earlier versions will be overwritten during installation by default.


Browse For Folder

Select a folder in the list below, then click OK.

C:\Program Files (x86)\Eaton\Galileo

Now select the component you want out of the ones that are available.


Setup - Galileo

Select Components

Which components should be installed?

Select the components you want to install; clear the components you do not want to install. Click Next when you are ready to continue.

Full installation

☑ Galileo	2[illegible] MB
☑ Help (German)	5.[illegible] MB
☑ Help (English)	3[illegible] MB
☑ Sample Projects	8,[illegible] MB

Current selection requires at least 1.[illegible] MB of disk space.

Galileo

Please note that this component cannot be disabled!
The latest program version will be installed with the selected installation option.

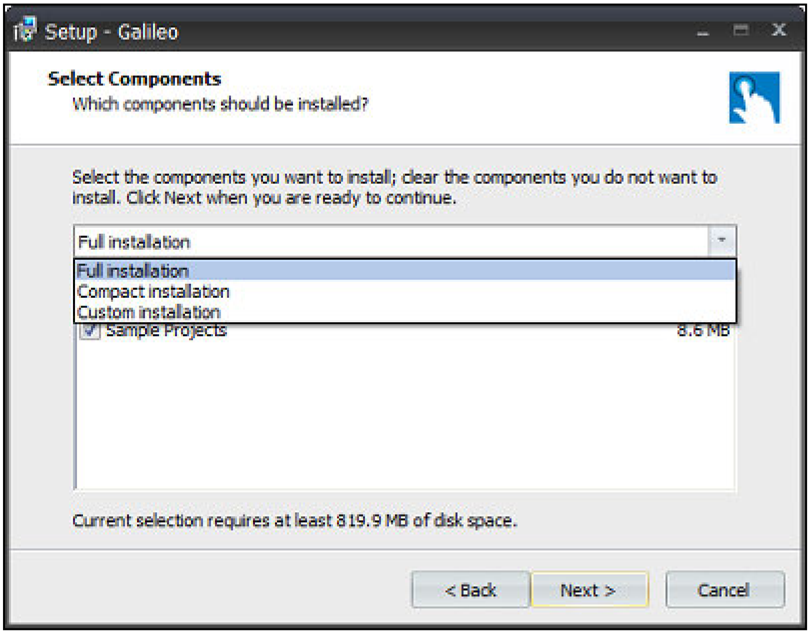
Hilfe (Deutsch)
Help (English)

If this component is enabled (☑),
the online documentation for the program will be installed in the selected language and you will be able to access it from GALILEO at any time.

Sample projects

If this component is enabled (☑),
a number of sample projects will be stored in the ...\Samples folder.
To access these sample projects, open GALILEO, click on the ▤▾ tab, and then click on Recent.

The following installation options will be available for each of the components:



Setup - Galileo

Select Components

Which components should be installed?

Select the components you want to install; clear the components you do not want to install. Click Next when you are ready to continue.

Full installation

- Full installation
- Compact installation
- Custom installation

☑ Sample Projects 8.6 MB

Current selection requires at least 819.9 MB of disk space.

Full installation

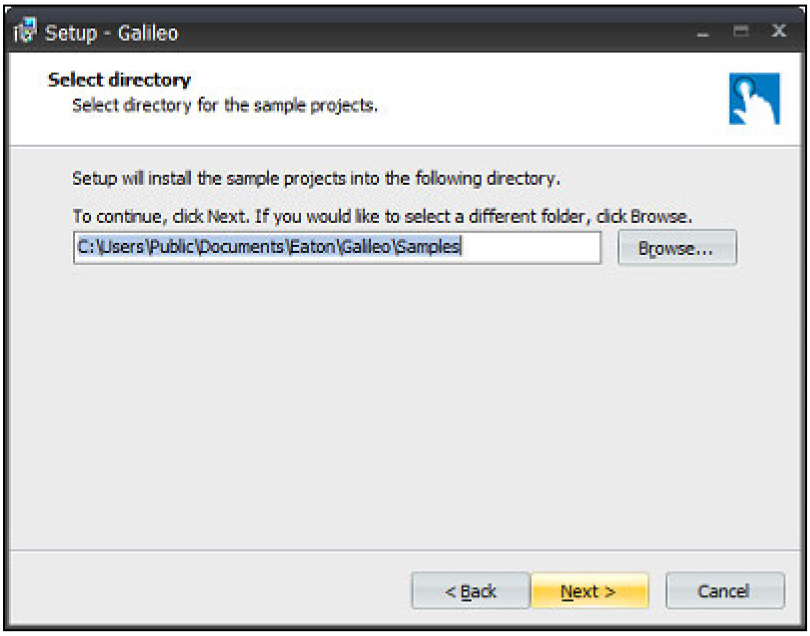
Default
All program components will be installed
Recommendation: New GALILEO users who will find the online help and sample projects useful should select this option.

Compact installation

Minimum
All essential program components will be installed
Recommendation: Advanced users who don't need any additional help should select this option.

Custom installation

Settings
Only the program components you select will be installed



Setup - Galileo

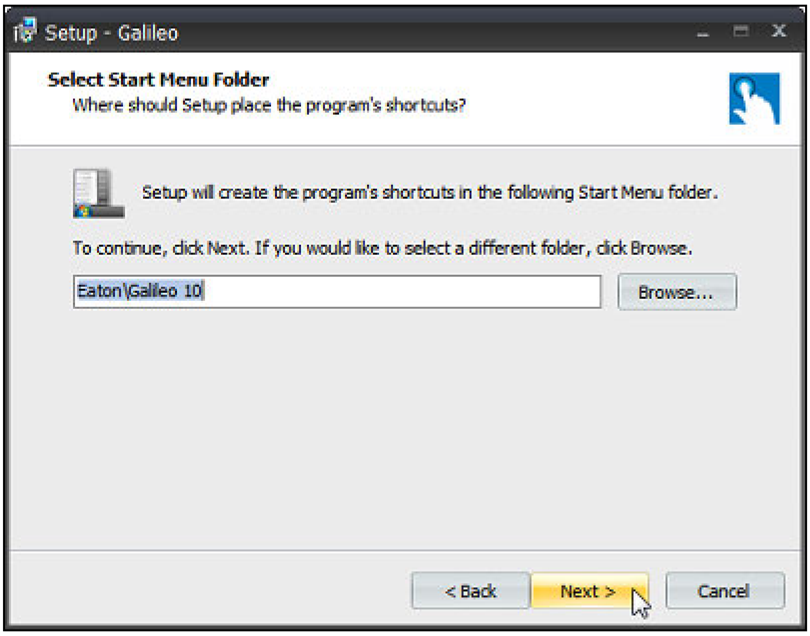
Select directory

Select directory for the sample projects.

Setup will install the sample projects into the following directory.

To continue, click Next. If you would like to select a different folder, click Browse.

C:\Users\Public\Documents\Eaton\Galileo\Samples



Setup - Galileo

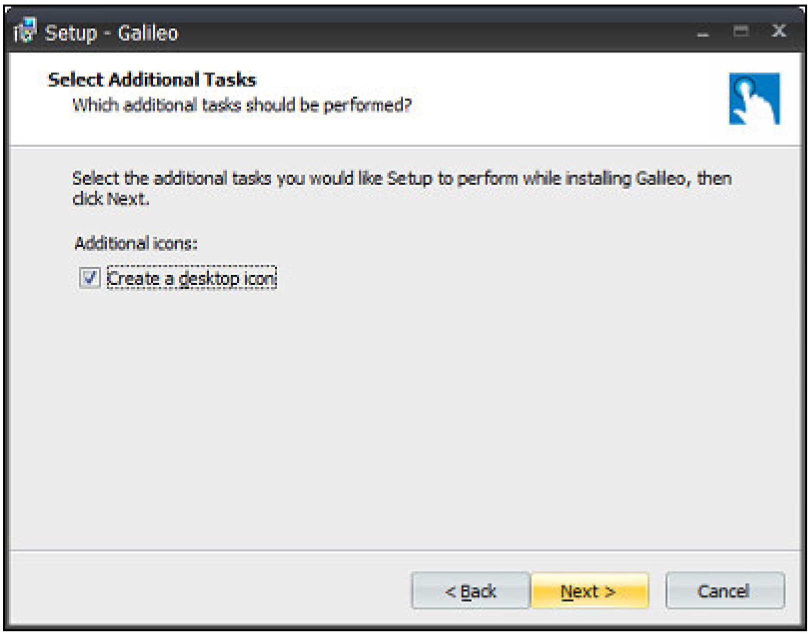
Select Start Menu Folder

Where should Setup place the program's shortcuts?

Setup will create the program's shortcuts in the following Start Menu folder.

To continue, click Next. If you would like to select a different folder, click Browse.

Eaton\Galileo 10



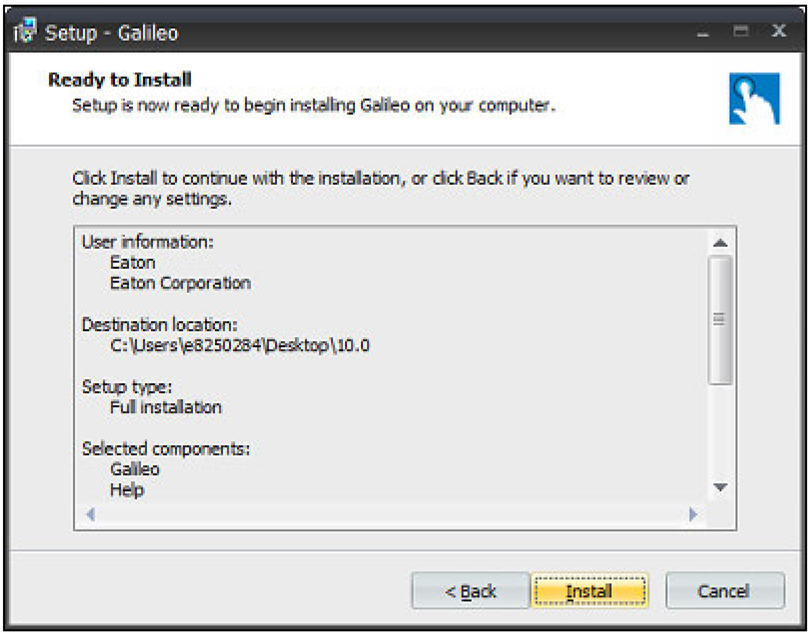
Select Additional Tasks

Which additional tasks should be performed?

Select the additional tasks you would like Setup to perform while installing Galileo, then click Next.

Additional icons:

☑ Create a desktop icon

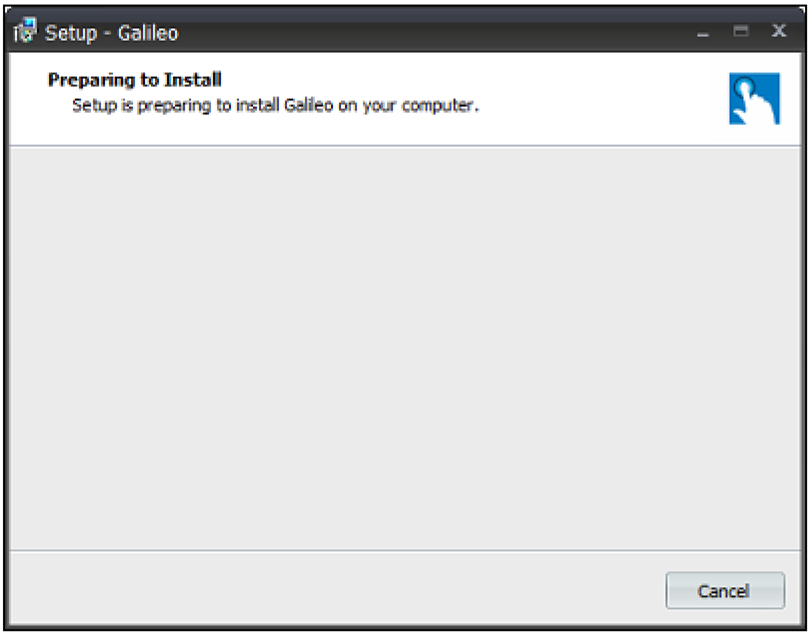


Ready to Install

Setup is now ready to begin installing Galileo on your computer.

Click Install to continue with the installation, or click Back if you want to review or change any settings.

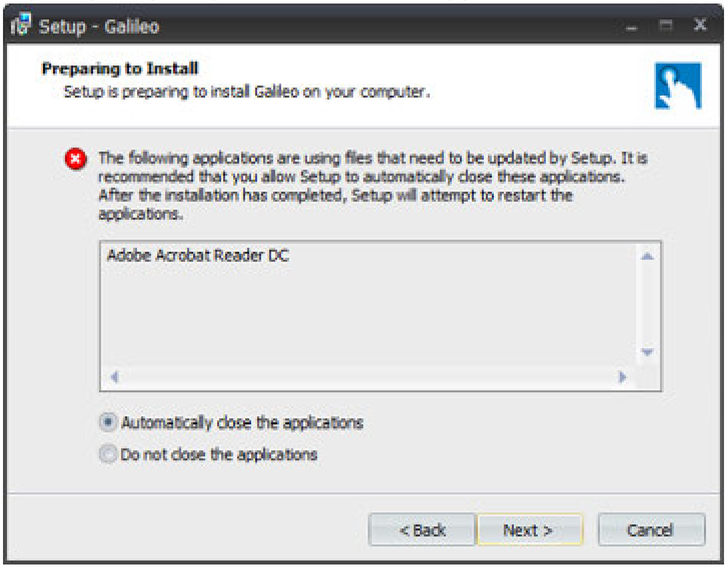
```
User information:
      Eaton
      Eaton Corporation

Destination location:
      C:\Users\e8250284\Desktop\10.0

Setup type:
      Full installation

Selected components:
      Galileo
      Help
```



Preparing to Install

Setup is preparing to install Galileo on your computer.

→ Running applications on your computer.
A prompt requiring confirmation will appear.



Preparing to Install

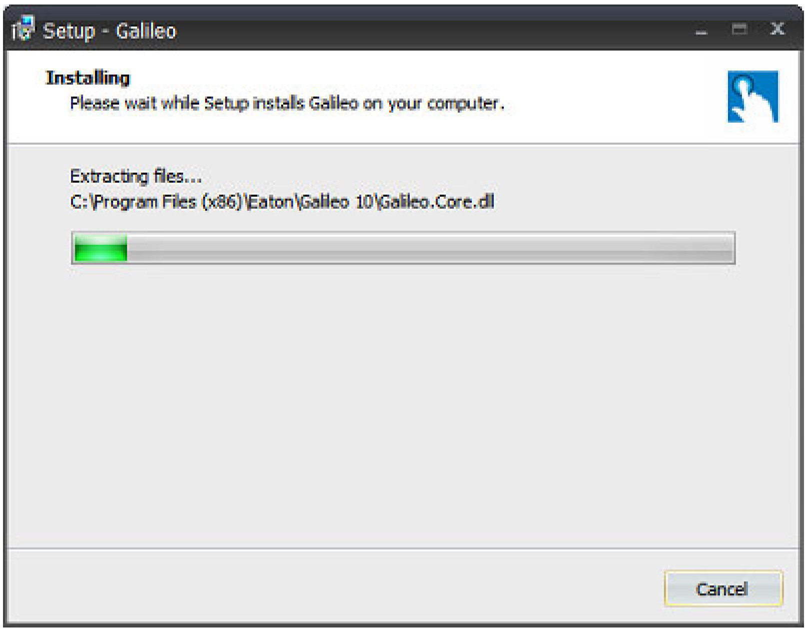
Setup is preparing to install Galileo on your computer.

The following applications are using files that need to be updated by Setup. It is recommended that you allow Setup to automatically close these applications. After the installation has completed, Setup will attempt to restart the applications.

Adobe Acrobat Reader DC

● Automatically close the applications

○ Do not close the applications

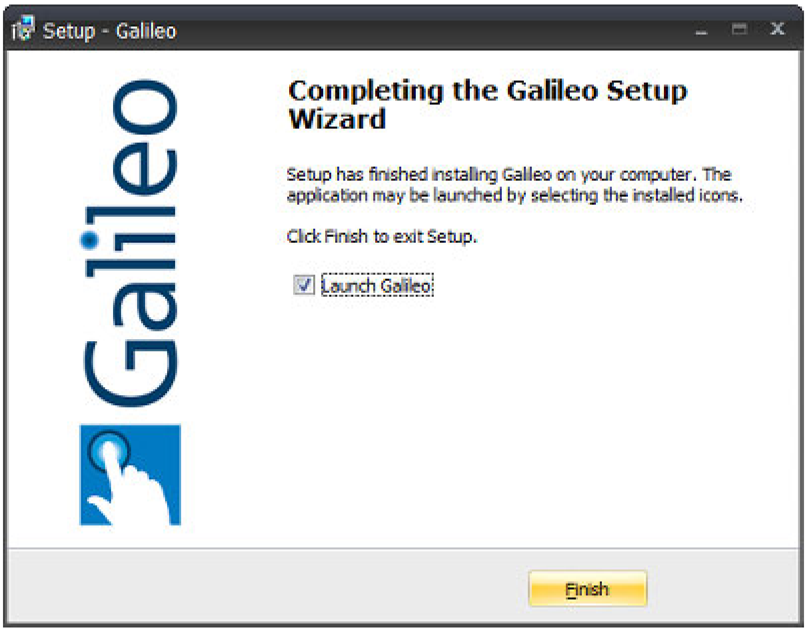


Installing

Please wait while Setup installs Galileo on your computer.

Extracting files...

C:\Program Files (x86)\Eaton\Galileo 10\Galileo.Core.dll

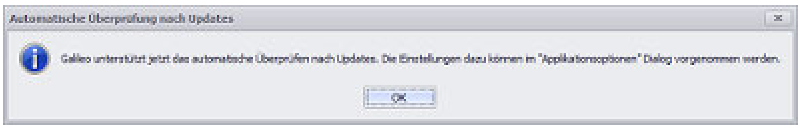


Completing the Galileo Setup Wizard

Setup has finished installing Galileo on your computer. The application may be launched by selecting the installed icons.

Click Finish to exit Setup.

☑ Launch Galileo



Automatische Überprüfung nach Updates

Galileo unterstützt jetzt das automatische Überprüfen nach Updates. Die Einstellungen dazu können im "Applikationsoptionen" Dialog vorgenommen werden.

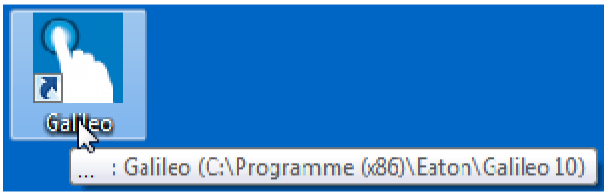
This prompt will be shown only once per PC/user.

See also

→ section "Messages from GALILEO", page 1145

1.4 GALILEO start

- ▶ To run the program, double-click on the icon that was created on your desktop during installation



Or

- ▶ Run the application by double-clicking on the "Galileo.exe" file in the program folder

Or

- ▶ Run the application from the Quick Launch bar

When you start GALILEO, it will open on its start screen.

→ You can run multiple instances of the program simultaneously. This can come in handy if you need to compare projects, for example.

→ In order to be able to use GALILEO, you must be familiar with how to use your computer and the usual applications for it. The way in which the GALILEO program is used is based on the Windows user interface developed by Microsoft Corporation.

Requirements for using the program

In order to be able to use the program, you must be familiar with the use and features of your touch panel, computer, and the usual Microsoft applications.

GALILEO uses the modular approach used in OOPM.

1.4.1 Running GALILEO for the first time

The first time you run it, the Automatization Software GALILEO program will start with English as the user interface language.

1.4.1.1 Used to change the user interface language

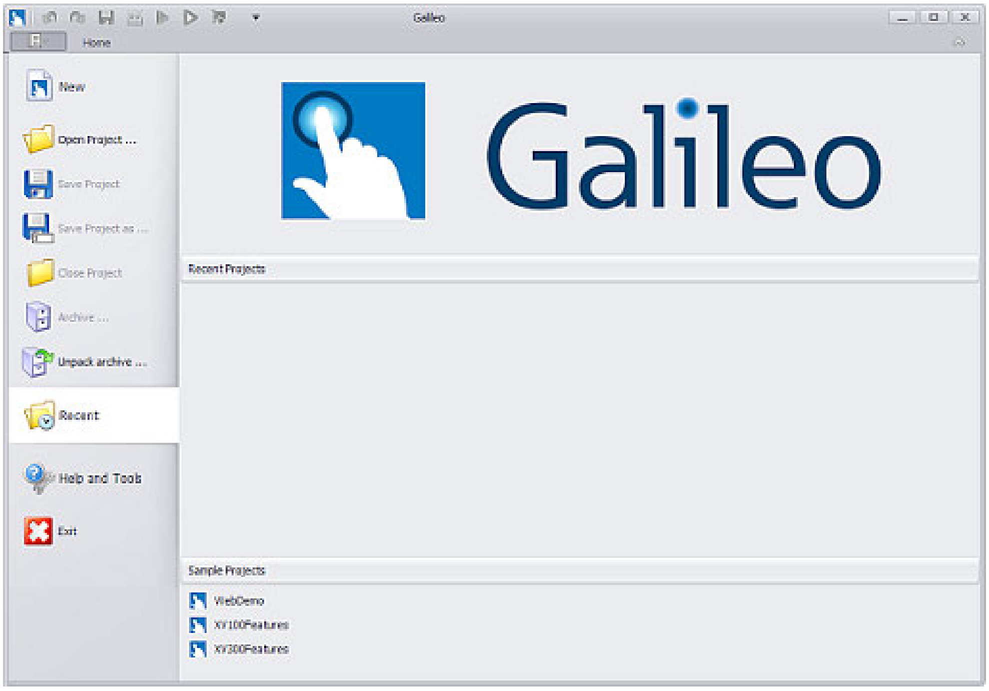


Figure 7: Start screen in English the first time GALILEO is run

Close the tab to access the Home tab, which will show the Application Configuration options used to change languages.

Home\Application Configuration

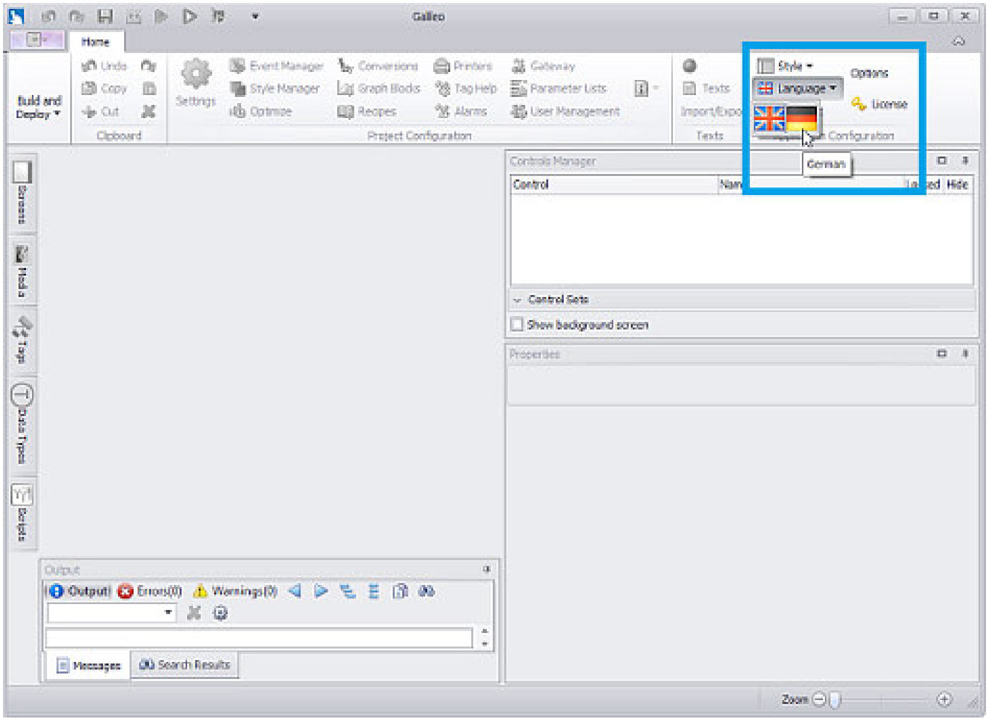


Figure 8: GALILEO start screen without an open project

- ▶ Click on 🇬🇧 Language and select the language 🇩🇪 you want (German, for example) for the user interface.
- ▶ Close the application and start it again.
  The user interface language will have changed.

> If you have not opened or created a project using the project management options, all other ribbon commands will be grayed out.

See also

→ section "The user interface", page 58

1.5 Commissioning touch panel

In order to ensure that the touch panel runs smoothly, it is especially important to make sure that the wiring is set up properly. Among other things, it must be routed separately from low-voltage wires and cables or be insulated with double or reinforced insulation.

→ For more information on how to connect and use your touch panel, as well as on its startup behavior, please refer to the corresponding operator manual touch panel.

1.5.1 Starting and shutting down the touch panel

When the touch panels are used as intended, they will often be installed in a machine or system to function as controls and will be wired accordingly.
In this case, the touch panels will need to be switched on and off using the corresponding machine controller.

⚠ **DANGER!**
Electric shock hazard!
When setting up and testing the visualization interface, make sure to use a test environment that is fused as per the state of the art and in accordance with all applicable standards in order to power the devices.

The touch panel will boot up as soon as it is energized.

If the touch panel will not start, or if an error message appears, consult the "Troubleshooting" section in the operating manual for your touch panel.

The touch panel will be shut down as soon as it is not being powered.

→ For more information on how to connect and use your touch panel, as well as on its startup behavior, please refer to the corresponding operator manual touch panel.

1.6 Getting started with GALILEO

1.6.1 Using the Automatization Software program

1.6.1.1 The user interface

GALILEO's user interface is not only intuitive, but also features an ergonomic design that keeps projects manageable at all times.
Following is a description of the various individual components that make up this user interface.

When you start GALILEO, it will open on its start screen. The configuration for this start screen will depend on your computer settings and on the program's default settings.

When you restart the program, your custom language, menu view, and window view settings will be applied.

Start screens in GALILEO

The application will start with a screen in which the tab on the ribbon is expanded. This means that both the project management functions and the Help function will be immediately available for selection.

Likewise, the → section "Example projects", page 139 will also be immediately available for selection if they have been installed.

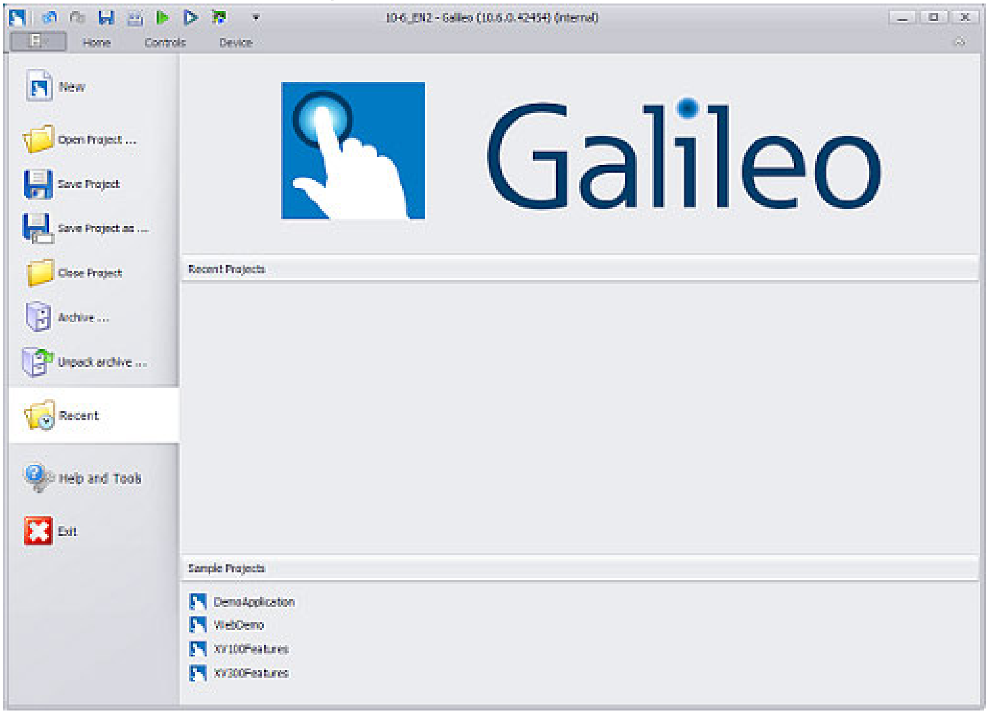


Figure 9: Start screen the first time GALILEO is run

If you close the ▾ tab, the Home tab will show the various commands available for customizing the application.

Home tab

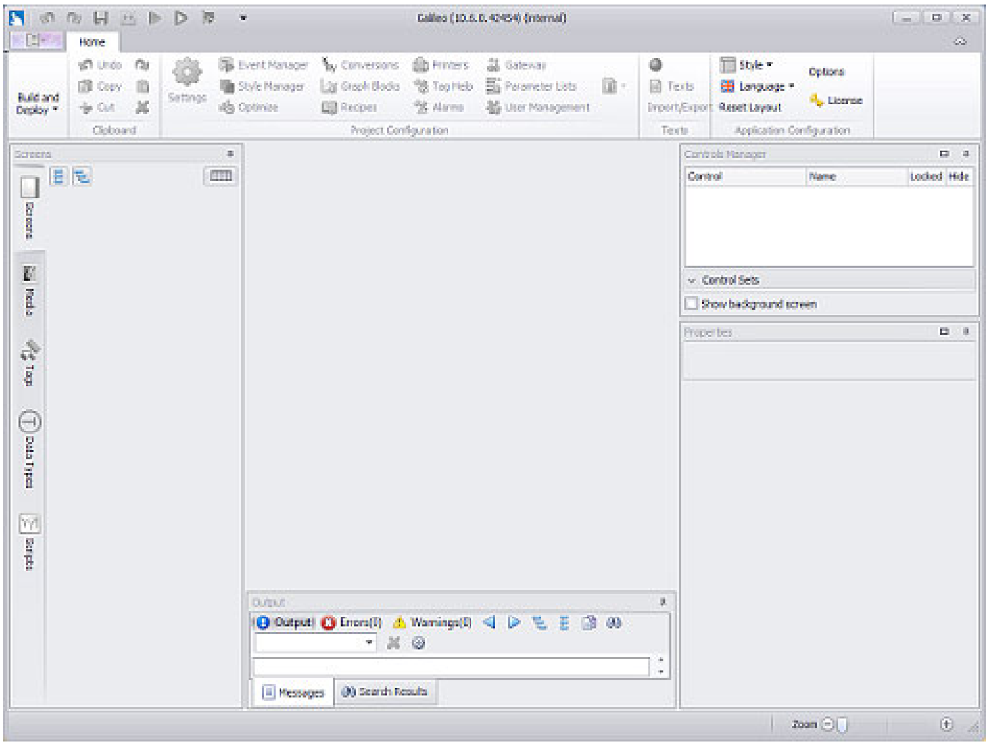


Figure 10: GALILEO start screen without an open project

Application configuration

Groups the functions used to configure the application in a user-specific mannerGALILEO

Style

Used to select the background for the application

There are several options available.

▶ Select one of the options from the ▾ drop-down menu.

Language

Used to change the user interface language

The languages into which the application has been localized will be available for selection.

▶ Select one of the options from the ▾ drop-down menu.

If the language is changed, the change will take effect only after the application is restarted.

Reset Layout

Resets the layout to the default settings, next time the application is opened.

All user-specific settings will be lost.

Options
Basic settings for the application on the computer,
e.g., autosave
Opens the → section "Window Application Options", page 135

License
Shows the License Keys
Opens the Licensing dialog box so that a license key can be entered in order to unlock the program.

If a project has not been opened or created using the options available under the tab, no other tabs or commands will be available for use.

User interface layout

The user interface is made up of the following sections:

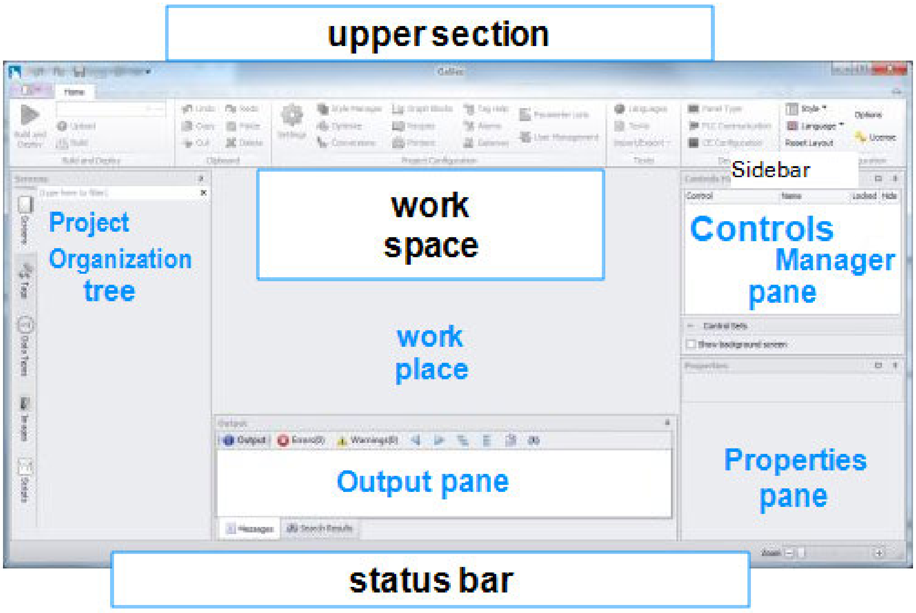


Figure 11: User interface layout

Upper section

The upper section contains the various control commands for GALILEO itself.

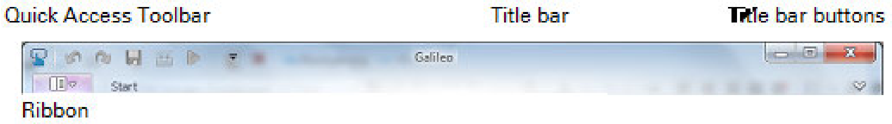


Figure 12: Upper section

1.6.1.2 Quick Access Toolbar

Quick Access Toolbar - The icons available in this toolbar can be used to quickly access frequently used commands.

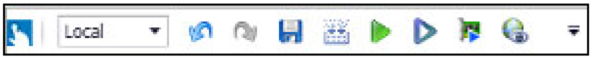
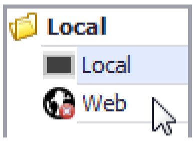


Figure 13: Toolbar above the ribbon (this option will save more space)



Visualization mode

Option

If the project is not only available locally - in other words, if visualizations with a variety of resolutions or web visualizations have been created for the project - you can use this drop-down menu to switch directly between the local application with its various visualizations and the application available through the web with all of the corresponding visualizations as well.
Described in section:
Web visualization

Undo

Undo function
When the cursor is moved over the icon, the action that will be undone will be shown. This function can be used to undo a maximum of 20 actions since the last time the project was saved

Redo

Redo function
When the cursor is moved over the icon, the undo action that will be redone will be shown. This function can be used to redo functions until the project is saved

Save Project

Saves the entire project in the specified storage location

Build

Generates an executable program for the selected Panel Type
(triggers the conversion of the GALILEO project into an executable program that the panel can run)
The Output pane will show the corresponding progress in real time

Building and deploying

Compiles the project and transfers the data to the configured destination

Deploy last successful compiled project state

Deploys the last successful compiled project state to the target or runs the simulator with it.

Run Communication Test

Runs the communication test for the addressed tags.

Show in browser

Shows the web visualization in the browser.
The browser can be set in the → section "Window Application Options", page 135.

▾	Drop-down menu for customizing the toolbar Context menu for positioning the toolbar User-specific settings can be used to show and hide individual functions and place the toolbar above or below the ribbon.

1.6.1.3 Title bar

The title bar shows which version of GALILEO is currently installed.

When working on a project, the following will be shown before the version information in order to make things easier to follow:

- The name of the screen that is currently active for editing
- The name of the open project

1.6.1.4 Title bar buttons

The title bar buttons for the program include buttons for minimizing and maximizing the window, as well as for closing the applicationGALILEO.

Figure 14: Title bar buttons

1.6.1.5 Ribbon

The ribbon features tabs that group thematically related commands, objects, and tools in GALILEO and provide access to them.
The Controls tab will not be shown until a screen has been created in the project.

The various available commands are combined into clearly laid out groups on these main tabs and made available in the ribbon.

The → "Functions on main tabs", page 76 are described in a separate section that explains how to use the program.

▤▾ Nameless tab
Provides access to the various project management functions:

- New Project... Wizard
- Opening, closing, saving, and archiving projects
- Help

Start Provides access to the following groups:

- Building and deploying
- Clipboard
- Project configuration
- Texts
- Application Configuration

Controls Provides access to the following groups:

	• Display • Controls • Arrange
Device	Provides access to the following groups • Device
⌃ ,	Minimize Collapses the ribbon
⌄	Maximize Expands the ribbon

Workplace

The workspace is used to show the project's screens so that they can be edited as necessary using the corresponding panes on the sides.

The view can be configured as a custom workspace.
→ "Customizing your personal workspace", page 131

1.6.1.6 Project tree

When a project is open, the project tree will show all the structures created in the project. These structures will be subdivided into individual folders and subfolders as appropriate in order to make managing the project easier.

The context menu for each tab makes it possible to create, structure, and manage the screens, tags, data types, images, and scripts in the project as applicable. Please note that certain commands are only available in this context menu.

The properties for the project data sets created in the relevant tab are defined in the Properties pane.

For every new project, a basic set of elements will be created as a data set in the various tabs. This basic set will include a "Start" screen, simple data types, and system tags.

[type here to filter X]

String search function
Within the corresponding tab, entries will be filtered directly based on whether they contain the string entered in this box, with the results shown in real time as the string is being entered.

Example: Finding all screens containing "Start"

If "Start" is entered into the search box for the Screens tag, all screens will be filtered and only the Start screen and any other screens with a name that contains "start" will be shown.

▦

Opens the table view with the overview of the screens and tables created in the project tree.

The settings for the order in which the tabs are displayed on the user interface, as well as for the tabs' type and location, can be customized as necessary for the PC user that is logged in.

Hidden project tree tag panes can be docked by clicking on ⊟ in the active tag.

□ Screens

This project tree tab contains the folders for the various screen types: Standard screens, UserControls (former Subscreens), Help screens, Keyboards, and Background screens as well as the Print report folder.

The corresponding context menu can be used not only to open the screens, but also to access a series of functions, including a preview of selected screens. These functions are listed below:

- Adding new folders and screens
- Renaming and deleting existing folders and screens
- Finding all references in the project
- Finding all exits in the project
- Showing a graph with all the screens in the project
- Setting one of the standard screens as the start screen for the project

Project Tree Screens

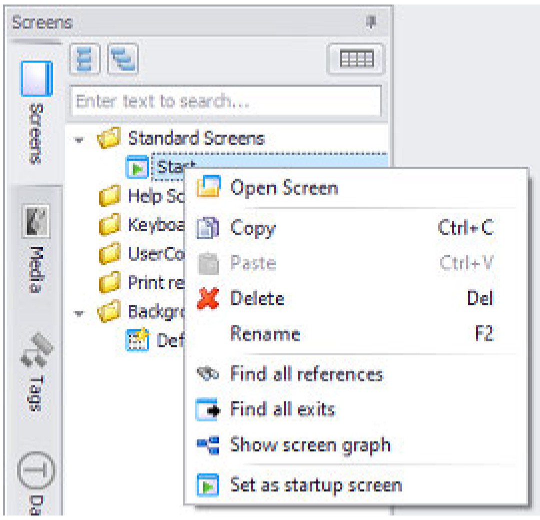


Figure 15: Context menu Screens

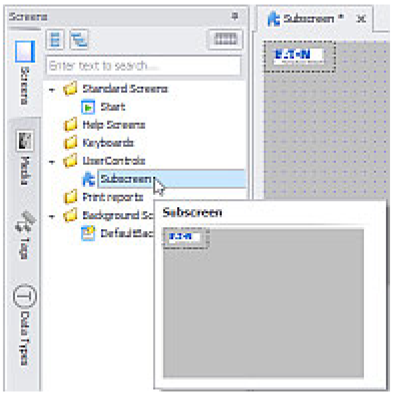


Figure 16: Preview in tab pane

Setting a screen as the start screen

The option for setting a screen as the start screen can be accessed by using the context menu in the Project Tree/Screens tab.

Please note that only standard screens can be set as a start screen.

▶ Open the Standard screens folder

▶ Select the screen you want

▶ Right-click on it

The context menu will appear.

▶ Select the "Set as startup screen" option

The screen will be set as the start screen and will accordingly be marked with the ▶ icon.

→ If you create a new project using the wizard, the empty "Start" standard screen created during the process will automatically be set as the start screen.

See also

→ section "Project organization and overviews", page 125
→ section "Displaying, editing, and saving screens on the workspace", page 133
→ section "Screens and User Controls", page 260
→ "Screen types", page 263
→ "Notes on creating a Project", page 192
Working with the screen table

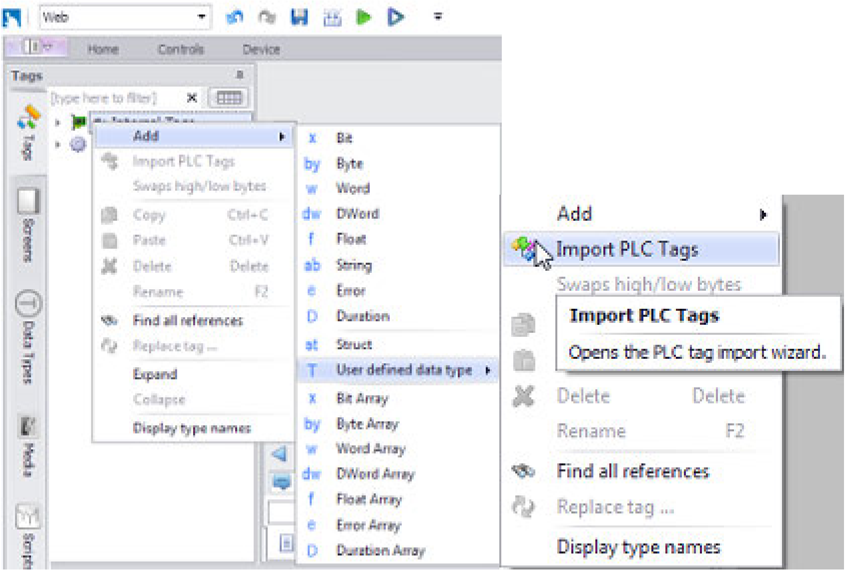
Tags

This project tree tab is used to manage both internal tags and system tags, as well as the tags provided for communication in data sets by the selected interfaces.

In addition to options for working with the tags, the context menu provides access to a series of functions and tooltips for the individual tags, as well as a table that can be used to edit tags in a clearly arranged overview. This table is described separately in → section "Working with the tags table", page 560

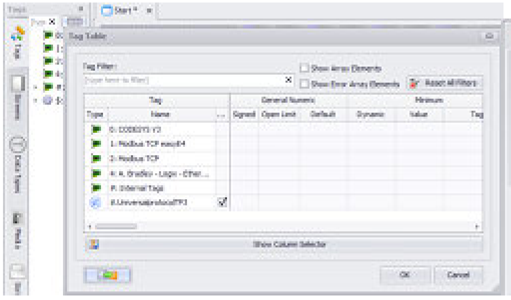
- Creating new tags
- Starting the wizard for importing PLC tags
- Renaming and deleting existing tags
- Finding all references in the project
- Showing properties

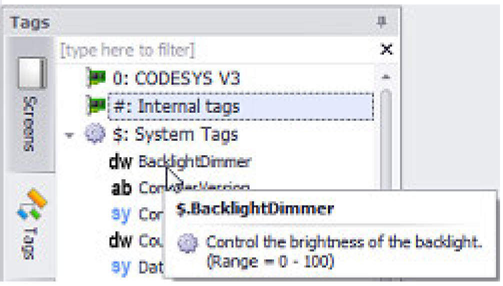
Tags tab in the project tree



Import PLC Tags

Opens the PLC tag import wizard.





$.BacklightDimmer

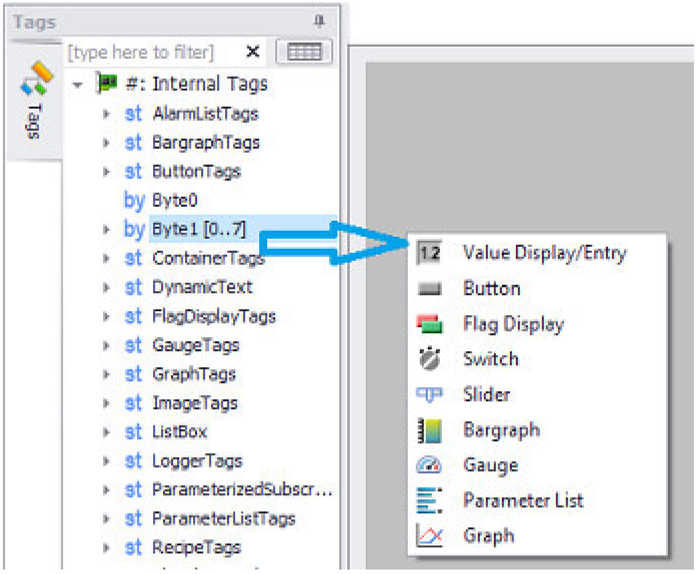
Control the brightness of the backlight. (Range = 0 - 100)

Figure 17: Context menu Tags

Figure 18: Tooltip in Folder

The Drag&Drop function is not supported.

Left-clicking will offer the most likely function, such as Add, while right-clicking will show all available functions.



See also

→ section "Variables /Tags", page 530
→ section "Tags and Data Types", page 532
→ section "Working with the tags table", page 560

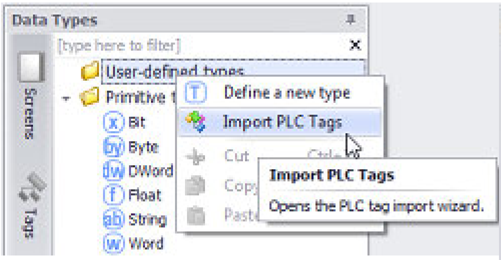
Data Types

This project tree tab is used to manage user-defined types and primitive types

In addition to options for working with the data types, the context menu provides access to a series of functions and tooltips for the individual data types. These functions are listed below:

- Creating new data types and importing PLC tags
- Changing data types

Project tree Data Types tab pane



Import PLC Tags
Opens the PLC tag import wizard.

Figure 19: Context menu Data Types

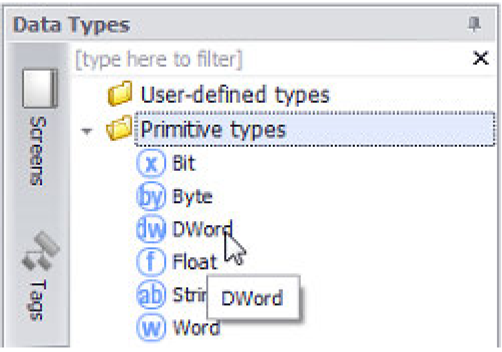


Figure 20: Tooltip in Folder

See also

→ "Data communication principles", page 580
→ section "Tags and Data Types", page 532
→ section "Importing tags", page 546

Media

This project tree tab is used to manage the Project images and Project videos folders containing the (image) media stored for the project.
The pane will show a preview of all imported project images. In addition, it will also show an image's size when the cursor is moved over the image.

In addition to options for importing media, the context menu provides access to a series of functions and a preview for the selected image. These functions are listed below:

- Adding additional folders, as well as filtering and sorting images by dragging &dropping them
  The project image must be in one of the following formats: *.png, *.jpg/*.jpeg , *.tiff/*.tif, *.gif , *.svg or *.bmp, image sequences with the *.gif format. The project video must be in one of the following formats: *.mp4, *.mov or *.avi.Icon library images are in .svg format.
- Editing images with the configured graphics editor
  (set in*Home\Application Configuration\Options\External tools*)
- Displaying the available media in the computer's directory structure
- Finding all references in the project
- Import icons from the GALILEO library

Project Tree\Media

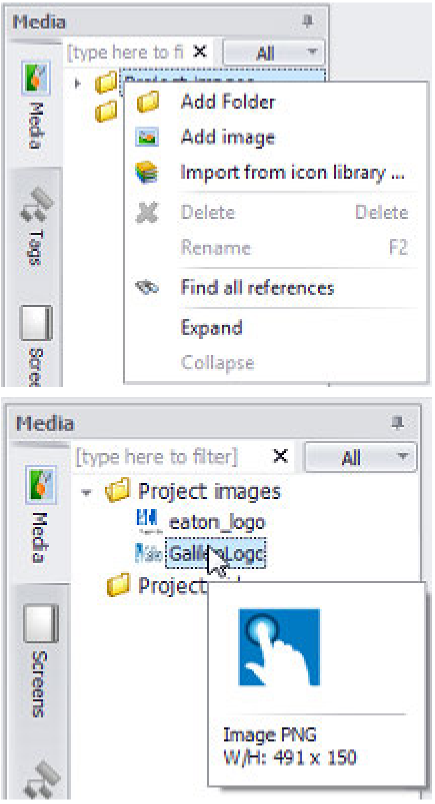


Figure 21: Context menu Images and Preview

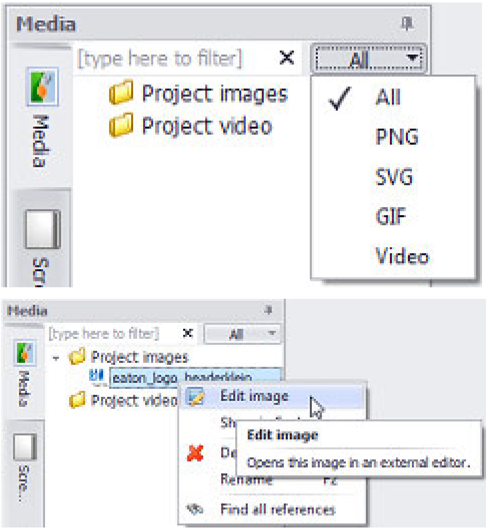


Figure 22: Format filter and tooltip in folder

See also

→ section "Substep 1: Adding an image (medium) to the project tree", page 174
→ section "Media - Using images and videos in a project", page 100
→ " Image", page 439
→ section "Icon library", page 101

Scripts

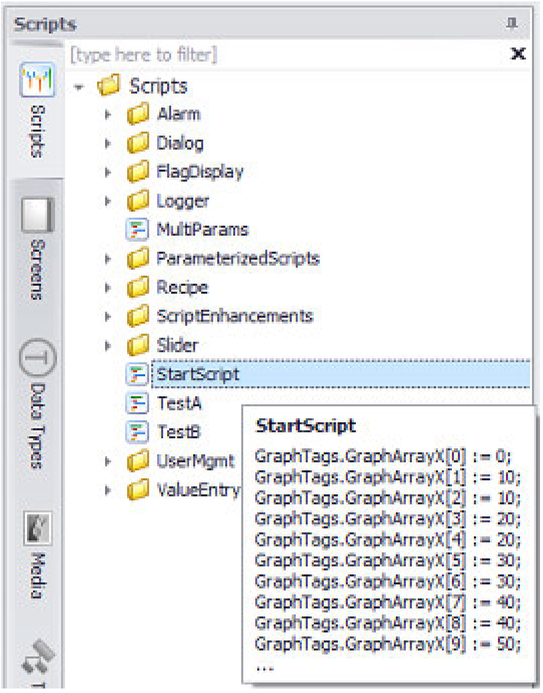
This project tree tab is used to manage scripts.

As soon as a script is selected, the Script tab will appear. This tab can be used to access and use the Script Editor's functions in an open script.

In addition to options for working with the scripts, the context menu provides access to a series of functions and tooltips for the individual scripts. These functions are listed below:

- Adding additional folders and sorting the scripts by dragging &dropping them
- Finding all references in the project

Scripts tab in the project tree



```
StartScript
GraphTags.GraphArrayX[0] := 0;
GraphTags.GraphArrayX[1] := 10;
GraphTags.GraphArrayX[2] := 10;
GraphTags.GraphArrayX[3] := 20;
GraphTags.GraphArrayX[4] := 20;
GraphTags.GraphArrayX[5] := 30;
GraphTags.GraphArrayX[6] := 30;
GraphTags.GraphArrayX[7] := 40;
GraphTags.GraphArrayX[8] := 40;
GraphTags.GraphArrayX[9] := 50;
...
```

Figure 23: Context menu Scripts

See also

→ section "Using scripts to perform actions", page 956

1.6.1.7 Workspace

As soon as a screen is activated, it will be opened in the workspace.

There is no limit to the number of screens that can be open at the same time. Clicking on the ▾ button will show a list of all open screens.

Open screens will be shown as individual tabs, and each one can be positioned on the workspace as necessary.
The context menu accessed by double-clicking on the <screen name title> can be used to customize the view.

See also

→ section "Context menu for<screen name> tab", page 134 tab

1.6.1.8 Controls Manager window

As soon as a screen is opened in the project, the various available tools and elements for designing the individual screens will be made available on the Controls tab. These controls can be managed using theControls Manager .

The Controls Manager pane will list every single object that has been placed on the corresponding screen.

The Controls Manager pane can be used to quickly, in an easily manageable manner, configure basic settings for the object being used.

As soon as you move the cursor over an object in the Controls Manager pane, the object will be highlighted with a blue border on the screen.

Every single object can be locked in place (so that it cannot be moved by accident) and/or hidden.
When one of these options is selected for an object, the setting will be applied immediately on the screen.
Hidden objects will:

- Stop being visible on the screen
- Be identified in the Controls Manager pane with ●

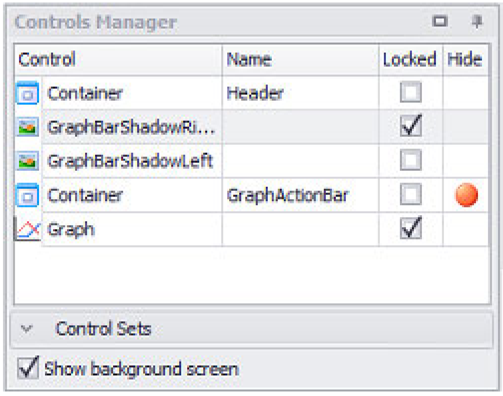
Locked objects will:

- Make the cursor change to when it is moved over them on the screen
- Be identified in the Controls Manager pane with ☑

The order of the objects in the Controls Manager pane defines the virtual arrangement of the objects on the screen, i.e., the object on the bottommost layer will be at the bottom of the list and the object on the uppermost layer will be at the top.
This order can be changed by dragging & dropping the objects as

necessary.
The resulting virtual arrangement will be important in various cases, such as when aligning objects relative to each other.

Example showing the Controls Manager pane



Controls Manager

Control	Name	Locked	Hide
Container	Header	☐	
GraphBarShadowRi...		☑	
GraphBarShadowLeft		☐	
Container	GraphActionBar	☐	●
Graph		☑	

Control Sets

☑ Show background screen

Figure 24: Controls Manager pane for the screen being shown

→ "Working with the Controls Manager", page 523
→ section "Controls for visualization", page 354

1.6.1.9 Properties window

The Properties pane is used to manage the specific properties of each individual object used in a project, from screens, to tags, to the text on a screen.

The pane shown here is the most basic version and only shows the most important properties. If specific, advanced properties are available and required, the pane will show appropriate icons as buttons.

Clicking on any of the icons at the top of the Properties pane, e.g., , will open the Extended properties dialog box.

A variety of context-sensitive settings can be configured for the object that is selected.

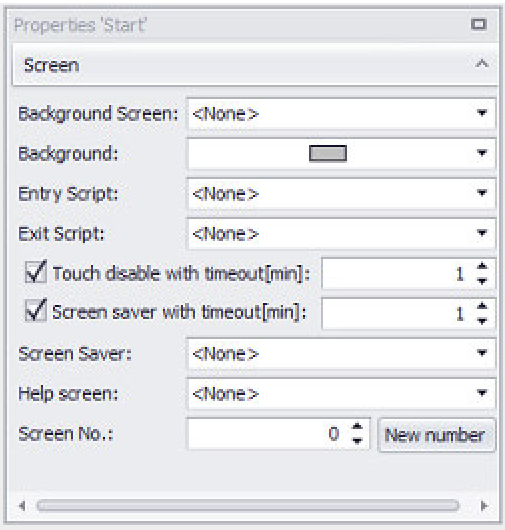
For more information on the various properties and their settings, please refer to the sections for the relevant object in the individual topics.

e.g.→ " Bargraph", page 356

Properties can be saved as a style so that they can be easily reused.
Please refer to the following section: → section "Style management", page 458

Two Properties panes shown as examples for comparison

Properties pane for the "Start" screen	*Properties pane for the parameter list object*



Properties 'Start'

Screen

Background Screen: <None>

Background:

Entry Script: <None>

Exit Script: <None>

☑ Touch disable with timeout[min]: 1

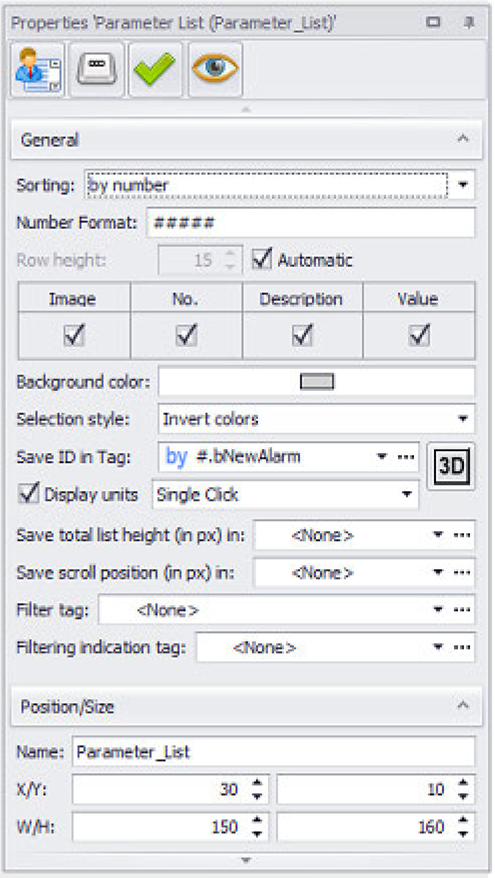
☑ Screen saver with timeout[min]: 1

Screen Saver: <None>

Help screen: <None>

Screen No.: 0 New number

There are no extended properties for screens



Properties 'Parameter List (Parameter_List)'

General

Sorting: by number

Number Format: #####

Row height: 15 ☑ Automatic

Image	No.	Description	Value
☑	☑	☑	☑

Background color:

Selection style: Invert colors

Save ID in Tag: by #.bNewAlarm

☑ Display units Single Click

Save total list height (in px) in: <None>

Save scroll position (in px) in: <None>

Filter tag: <None>

Filtering indication tag: <None>

Position/Size

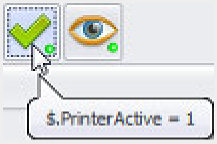
Name: Parameter_List

X/Y: 30 10

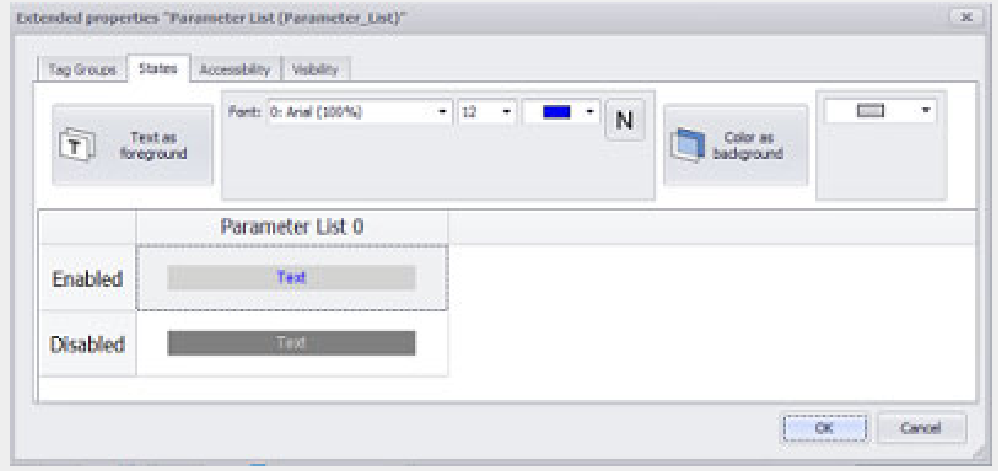
W/H: 150 160

If a parameter list object is selected, the expandable "General" and "Position/Size" menus will be shown together with the corresponding options.

In addition, the "
Visibility," "
Accessibility," "
States," and
"Tag Groups"
icons will be shown as well. When clicked on, these icons will open the Extended properties "Parameter List" dialog box, which will provide additional configuration options that will then be shown as tooltips.



$.PrinterActive = 1



Font: 0: Arial (100%) 12

Figure 25: *Extended properties 'Parameter List' dialog box, States tab*

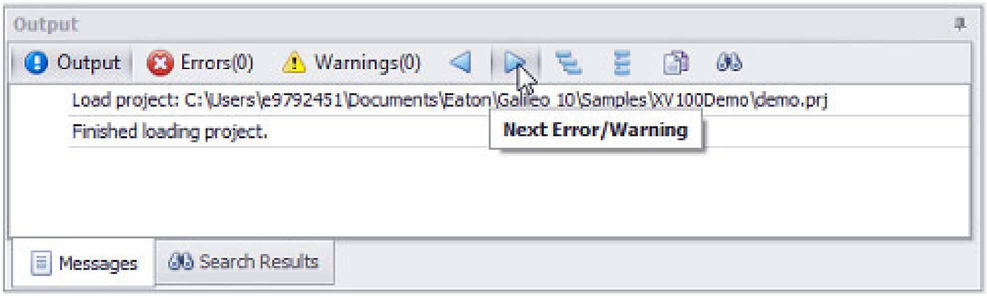
1.6.1.10 Output window

Register ☰ Messages

All messages generated by the GALILEO while you are working with Automatization Software will be shown above the Messages tab in this pane.

When you compile the project, all the corresponding results will be shown in the Output pane.

In order to make the messages easier to read and follow, they will be grouped into the ⓘ Output, ✖ Errors, and ⚠ Warnings categories, and the number of messages will be shown between parentheses if applicable.



Load project: C:\Users\e9792451\Documents\Eaton\Galileo 10\Samples\XV100Demo\demo.prj
Finished loading project.

Figure 26: Output pane

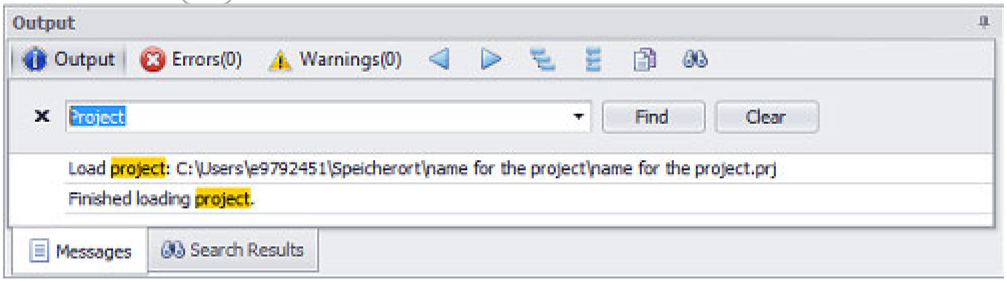
The ◁ ▷ icons can be used to navigate.

Clicking on the button will copy the entries being shown in the pane to the clipboard so that they can be used as necessary.

Clicking on the icon will display a search box that can be used to search for specific messages in a targeted manner.

Tooltips will be shown in order to help you select the function you want.

Please note that this search will only search in the Output pane.
Clicking on the Find button will search for the text in the Output panes.



Load project: C:\Users\e9792451\Speicherort\name for the project\name for the project.prj
Finished loading project.

Figure 27: Search in Output pane

If a list is not shown, this means either that the Output pane is "blank" or that the search term does not appear in it.

Clicking on x will close the search box.

⚙ will open the Migration Guide.

Register Search Results

Shows the relevant search results after using the Find all references context menu option on the project tree folders.

See also

→ section "What happens If ...", page 1145
→ section "System Messages", page 1147
→ section "Compile and transfer", page 1035

Status bar

When a project is being worked on, the status bar will show the position and size of the object selected on the screen.

The zoom function can be used to zoom in to and out of all the open screens on the workspace. Double-clicking on Zoom will restore all screens to a zoom level of 100%.

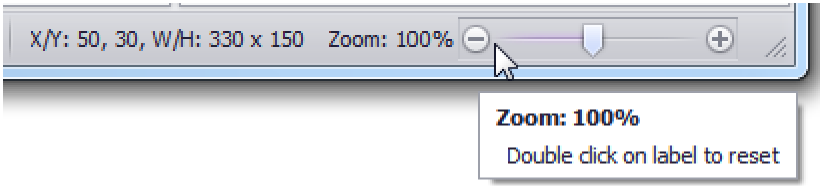


Figure 28: Active zoom function

See also

→ section "Control elements & User guidance", page 92
→ section "Customizing the application's user interface", page 131
→ section "Functions on main tabs", page 76

1.6.1.11 Functions on main tabs

The various available options are combined into clearly laid out groups on the main tabs.

→ "Home tab ", page 76 ▤	→ "Tab Home", page 78	→ "Controls tab", page 82	→ "Tab Device", page 91
Project management	Project organization	Project design	Project implementation

Referred to in the following as register.

Home tab ▤

The ▤ tab groups the basic functions used for project management purposes, as well as the application's basic commands.

Basic commands

New Project

Starts the wizard used to create a new project
The New Project... Wizard will guide you through the process.
When you create a new project, the standard "Start" screen will be created automatically.
Described in section:
→ section "Step 1: Creating a new project", page 158

Open Project

Opens the directory structure on the computer's hard drive so that you can look for and open the GALILEO project file (.prj) you want

Save Project

Saves the open GALILEO project files (.prj)

Save Project as-

Saves the entire open GALILEO project data(.prj) under a different project name
Described in section:→ section " Project as...", page 115

Close Project

Closes the open GALILEO project file (.prj) after asking whether you want to save the changes you have made.

Archive...

Saves the open GALILEO project (.prj)as a .zip archive
Described in section:→ section "Archive...", page 117

Unzip archive...

Opens the directory structure on the computer's hard drive so that you can select the Galileo archive file (.zip) you want.
After you select the archive file, you will need to select an empty folder where you want the contents of the file to be extracted. The unzipped project will then be

shown directly for opening.

Recent

Shows a chronological list of the most recent projects,
as well as a list of sample projects if the latter have been installed.

If the list is longer than the space available on the window, it will be a scrollable list.
Please note that this list will be blank right after installing GALILEO!

Help and Tools

Shows the available documents and can be used to open the remote client tool for support
After an update, this page will be shown and new online contents will be indicated with a star. For example:

Tools

Can be used to access available communication tools, e.g., Remote Client

Help

Accessing the online documentation
To open the online documentation at any time, simply press F1 while on the application.

Direct access to the corresponding PDF document, which will be opened outside GALILEO
You will need to have a PDF viewer installed in order to be able to view the document.

Online contents
These contents are opened outside GALILEO
You will need to have a corresponding viewer installed and/or Internet access in order to be able to view these contents.

Direct access to the corresponding PDF document

Direct access to the corresponding tutorial,
videos that explain how individual functions work
You will need to have access to Eaton's websites or YouTube

You can find the tutorials on the Internet
at [youtube.com]

available updates

Exit

Closes the program;
standard Windows function

To access the application's settings, go to→ "Tab Home", page 78
A section on how to use the program goes over → "Customizing the application's user interface", page 131.

Tab Home

The Home tab can be used to directly access commands that affect the entire project.

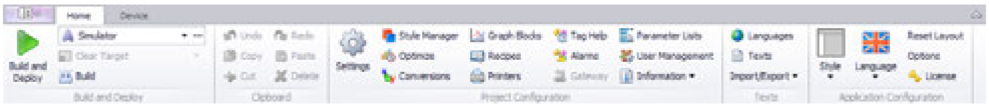


Figure 29: Home tab

These commands are grouped as follows:

- → "Build and Deploy", page 78
- → "Clipboard", page 79
- → "Project configuration", page 80
- → "Texts", page 81
- → "Application configuration", page 82

Build and Deploy

Contains the commands for building and compiling projects

▶ Building and deploying		Compiles the project and transfers the data to the configured destination
Simulator		Shows the currently selected data connection for the project:
	▾	Opens the list of all configured computer connections for selection
		FTP connections - to devices with an Ethernet port
		File System Folders - Data storage on a local path
		Simulator - Galileo Simulator on the computer
...		Opens the Configuration for Build and Deploy windowDescribed in section:→ section "Configuration for Build and Deploy", page 1056
Empty Target System		Shows the currently configured Build and Deploy settings Deletes the project directories found on the target system
	▾	Opens the list of all configured computer connections for selection
		Empty Target System Deletes the project directories found on the target system

Download OS
Downloads the operating system and reboots the target

Download components
Downloads the selected components

Restart GRS
Restarts the GALILEO Runtime System on the target system

Reboot
Reboots the target

Upload
Upload data from target

Create

Generates the data for the GRS,
triggers the function that generates the required data and converts the open project to an executable program for the selected panel model,
the corresponding progress will be shown in the Output pane

See also

→ section "Getting started with ", page 144
→ section "Compile and transfer", page 1035

Clipboard

Refers to the controls grouped on the Controls tab

Undo

Undo function
When the cursor is moved over the icon, the action that will be undone will be shown. This function can be used to undo a maximum of 20 actions since the last time the project was saved

Redo

Redo function
When the cursor is moved over the icon, the undo action that will be redone will be shown. This function can be used to redo functions until the project is saved

Copy

Copies the parameters for the selected tool/object, after which they can be duplicated as many times as necessary with the Paste command (the new pasted object will be offset towards the bottom right)

Paste

Pastes the (internally) copied tool/object into the active screen.

Cut

Internally copies the parameters for the selected tool/object and deletes the tool/object from its position, after which the Paste command can be used to paste it into a different screen, for example.

Delete

Deletes the selected tool/object

See also

→ section "Control elements & User guidance", page 92

Project configuration

Groups the various input and selection dialog boxes used to manage the objects being used in the project

Settings	Basic project settings for all the screens in the project Opens the Project Settings dialog box Described in section: → "Project Configuration", page 196
Event Manager	Used to manage the script events created for the project Opens the Event Manager dialog box Described in section: → section "Event Manager", page 1001
Style Manager	Used to manage the object styles created for the project Opens the Style Manager dialog box Described in section: → section "Create Styles with the Style Manager", page 461
Optimize	Used to manage the object styles created for the project Opens theOptimize dialog box Described in section: → section " Optimizing project", page 219
Conversions	Groups tags and can be used to apply simple arithmetic operations Opens the Conversions Described in section: → "Conversions", page 921 dialog box
Graph Blocks Definitions	Groups the lists used to plot tag values and curves Opens the Graph Blocks Described in section: → "Graph Blocks", page 661
Recipes	Used to group values and sequences together Opens the Recipe Editor Described in section: → "Recipes", page 736 dialog box
Printers	Used to add printers Opens the Printers Described in section: → " Printer", page 1026 dialog box
Tag Help	Used to create and customize the help management elements for tags Opens the Tag Help Described in section: → " Tag Help", page 935 dialog box
	Used to create and customize the messages generated by the project Opens the Alarms

Alarms

Described in section:
→ " Alarm list", page 867 dialog box

Gateway

Used to group the relationships between tags and communication methods
Opens the Gateway dialog box
Described in section:
→ section "Using the gateway functionality to communicate with multiple PLCs", page 596

Parameter Lists

Used to group existing tags into a parameter list
Opens the Parameter Lists
Described in section:
→ section " Parameter Lists", page 820 dialog box

User Management

Used to create user profiles
Opens the User Management
Described in section:
→ "Properties User management", page 894 dialog box

Information

Shows information about the project
With selection drop-down menu

Project information
Opens the Project Information dialog box with statistical information on the project and the option of exporting all created screens as PNG images to a folder.

Show screen graph
Opens the Screen Graph dialog box with the View Finder
Described in section:
→ section "Navigating with the Screen Graph", page 352

Texts

Groups the various commands for managing and customizing texts together. The following section goes over how to work with texts in a project:

Described in section:

→ "Languages and texts", page 630

Languages

Used to add additional languages to the open project
Opens the Project Languages dialog box
Described in section:
→ "Languages", page 633

Texts

Used to manage the standard, help, and error texts that have been created
Opens the Text Editor
Described in section:
→ section " Texts - Text Editor", page 649 dialog box

Import/Export ▾

Used to open the wizard for importing and exporting texts

▶ Select one of the options from the ▾ drop-down menu.

Import Texts

Can be used to import external defined text files into the project in the form of standard texts, help texts, or error texts
Opens the→ " Import texts", page 653

Export Texts

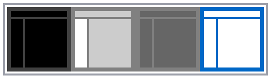
Used to export standard texts, help texts, or error texts from the project in CSV format
Opens the Text Export
Described in section:
→ "Importing/Exporting texts", page 653 dialog box

Application configuration

Groups the functions used to configure the application in a user-specific mannerGALILEO

Style

Used to select the background for the application

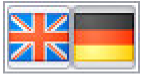


There are several options available.

▶ Select one of the options from the ▾ drop-down menu.

Language

Used to change the user interface language



The languages into which the application has been localized will be available for selection.

▶ Select one of the options from the ▾ drop-down menu.

If the language is changed, the change will take effect only after the application is restarted.

Reset Layout

Resets the layout to the default settings,
next time the application is opened.
All user-specific settings will be lost.

Options

Basic settings for the application on the computer,
e.g., autosave
Opens the → section "Window Application Options", page 135

License

Shows the License Keys
Opens the Licensing dialog box so that a license key can be entered in order to unlock the program.

Controls tab

As soon as a screen is opened in the project, the Controls tab will provide all the commands required in order to design the contents for the various individual screens.

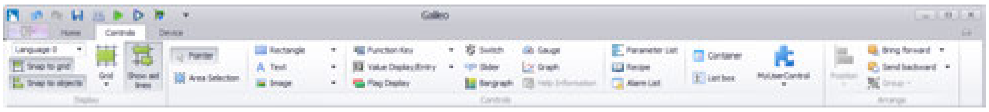


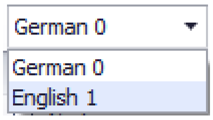
Figure 30: Controls tab

These commands are grouped as follows:

- → "Display", page 83
- → "Controls", page 84
- → "Arrange", page 90

Display

Used to switch between the project languages that have been created for the project.
All available texts on the various screens will be shown in the selected project language.
If a project language has not been set up, only the default setting, <Language 0>, will be available.
To add additional languages to the open project, go to *Home\Texts\Languages*.
In the following example, German has been set as project language 0 and English has been set as the 1st alternative language



Snap to grid

Positioning aid
When the left mouse button is being held down, the selected object will automatically snap to the grid.
This function can also be used after selecting multiple objects.
Grid settings can be configured by going to → section "Grid Settings", page 212.
Described in section:
→ section "Project Configuration", page 196

Snap to controls

Placement Help

Grid

Used to select how the grid will be shown

Show in background

Show in front

Don't show

Show auxiliary lines

Can be used to align and position the objects on the screen

Controls

Pointer

Used to select objects (one or more objects) within the active screen
in order to run commands in a targeted manner.

To select multiple objects, draw a rectangular marquee around them by holding down the left mouse button while moving the mouse or
select them in the Controls Manager while holding down the Shift or Ctrl key as applicable.

Area Selection

Used to select multiple objects on the screen by drawing a range with the mouse.

This tool is especially helpful for selecting multiple objects when, for example, there is a bigger object in the background for these objects.
Using the conventional "selection" tool would automatically select the background object instead on the first click.
The Area Selection tool avoids this problem.

are grouped by function. For a description of the object concepts, please refer to → "Controls for visualization", page 354

Objects used to design screens

Static visualization objectsare tools that can be used to graphically design the view for the individual screens. Static objects will be placed in the screen's background and will not change at runtime, i.e., they will not change later on while the generated visualization interface is running on the panel.

- Select the design tool you want on the Controls tab before placing it on the screen.

When you select a control, the cursor icon will change from ↖ to +.

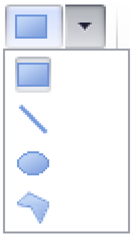
- After selecting the object and clicking on the screen position you want, keep the left mouse button held down and drag the cursor in order to set the size you want for the object.

The static controls in the program consist of ready-to-use geometric shapes for drawing and tools for inserting texts and images into screens.

- Controls for drawing (Line, Rectangle, Ellipse, Polygon)
- Controls for inserting text boxes and animations (text, image, HTML Viewer, video player, webcam)

Described in section:

→ section "Controls for visualization", page 354



Drop-down menu used to select ready-to-use geometric shapes.
The currently active shape will be shown as a button.

Rectangle

Used to insert a rectangular shape into a screen.
This shape can then be configured as necessary using the Properties 'Rectangle' pane.

Line

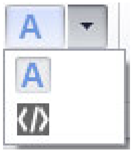
Inserts a line that can then be configured as necessary using the Properties 'Line' pane.

Ellipse

Inserts an ellipse that can then
be configured as necessary using theProperties 'Ellipse' pane.

Polygon

Inserts a "polygon" drawing element that can then
be configured as necessary using the Properties 'Polygon' pane.



Drop-down menu used to select objects.
The active medium will be shown as a button.

Text

filled with text and visually configured as necessary using the
Inserts a box as a placeholder that is then
Properties 'Text' pane.
Described in section:
→ section "Languages and texts", page 630

HTML Viewer

filled with a URL and configured as necessary using the
Inserts a box as a placeholder that is then
Properties 'HTML Viewer' pane.
Described in section:
→ section " HTML viewer", page 435

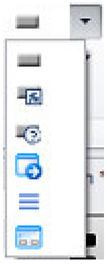
	Drop-down menu used to select objects from the Media project tree tab. The active medium will be shown as a button.
Image	filled with an image or animation and positioned using the Inserts a box as a placeholder that is then Properties 'Image' pane. Described in section: → section " Image", page 439
Webcam	filled with a web address (URL) and positioned using the Inserts a box as a placeholder that is then Properties 'Webcam' pane. Described in section: → " Webcam", page 515
Video player	filled with a video and positioned using the Inserts a box as a placeholder that is then Properties 'Video player' pane. Described in section: → section "Integrating videos", page 498

Controls for entering and displaying values

Dynamic controls are tools that can be used to add functional elements to the various individual screens.

The dynamic controls in the program consist of ready-to-use objects. In contrast, dynamic visualization objects are located in a screen's foreground, and their appearance can change as a result of data changes. These data changes, in turn, can be the result of time passing (time of day, etc.) or, depending on the device, machine, and system controllers (PLC) being used, of automatic processes into which the project has been integrated.

- Objects for displaying a state in the PLC
  (Flag Display, Value Display/Entry, Bargraph, Alarm List, PLC Text)
- Objects for manipulating data in the panel's (HMI) PLC
  (button, switch, value display/entry, bargraph, graph, recipe, slider)
- Objects that trigger or display special functions
  (screen change, alarm list, date/time display, container, webcam, video player)
- Objects for inserting lists (Parameter List)



Drop-down menu used to select ready-to-use buttons.
The currently active button type will be shown as a button.

Pre-defined button functions

Button

This object can be used to set individual bits in the PLC to a value of 0 or 1.
Described in section:
→ section " Button", page 365

Function Key

This object adds a key to the screen that can be used to start defined actions from the panel.
Inserts a box as a placeholder that is then
Assigned to specific functions using the Properties 'Function Key' pane.
Described in section:
→ section "Creating a function key visualization object and visually configuring the corresponding button", page 390

Help Button

This object can be used to display information in an additional screen on the screen.
Described in section:
→ section " Help Button", page 428

Screen Change

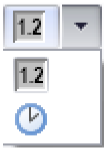
This object can be used to enable users to change the screen being shown on the panel.
Inserts a box as a placeholder that is then
Defined as necessary using the Properties 'Screen Change' pane.
Described in section:
→ section "Screen types", page 263

Menu Button

Special user control
This object can be used to take individual commands, group them together as a single object, and then show this object as an option that can be selected in the application
Described in section:
→ section "Using containers", page 294

Dialog Button

Special user control
This object can be used to show dialog boxes with the user on the screens.
Described in section:
→ section " Dialog button", page 375



Drop-down menu used to select objects as ready-to-use display/entry objects.
The currently active display/entry object type will be shown as a button.

Pre-defined display/entry functions

1.2
Value Display/Entry

This object can be used to display a value on the screen.
Inserts a box as a placeholder that is then
Filled with a tag and configured as necessary using the Properties 'Value Display/Entry' pane. The keyboard that will appear so that a value can be entered is one of the properties that can be defined in the pane.
Value display/entry
Described in section:
→ section " Value Display/Entry", page 506

Date/Time Display

This object can be used to add a clock element to the screen.
Configured as necessary using the
Inserts a box as a placeholder that is then
Properties 'Date/Time Display' pane.
Described in section:
→ section "Date Properties/Time Display", page 372

Pre-defined objects used for visual feedback and displaying states

Flag Display

This object can be used to display a switching state on the screen.
Inserts a box as a placeholder that is then
Filled with a tag and configured as necessary using the Properties 'Flag Display' pane.
Described in section:
→ section " Flag Display", page 379

Switch

This object can be used to display a switch on the screen.
Inserts a box as a placeholder that is then
Filled with a tag and configured as necessary using the Properties 'Switch' pane.
Described in section:
→ section " Switch", page 485

Slider

This object can be used to adjust tag values, e.g., to dim a light. The object can be defined with a horizontal or vertical format.
Described in section:
→ section " Slider", page 453

Bargraph

This object can be used to display the magnitude and weight of a signal in the form of the length of a bar on the screen.
Inserts a box as a placeholder that is then
Filled with a tag and configured as necessary using the Properties 'Bargraph' pane.
Described in section:
→ section " Bargraph", page 356

Gauge

This object can be used to display a gauge used to show readings with a scale on the screen. Filled with a tag and configured as necessary using the
Inserts a box as a placeholder that is then
Properties 'Gauge' pane.
Described in section:
→ section " Gauge", page 408

Pre-defined graph and table elements

Graph Block

This object can be used to display series of values as a function of time on the screen.
Inserts a box as a placeholder that is then
filled with tags and configured as necessary using the Properties 'Graph' pane.
Described in section:
→ section "Graph Blocks", page 661

Parameter list

This object can be used to display a table on the screen. Filled with tag groups and configured as necessary using the
Inserts a box as a placeholder that is then
Properties 'Parameter List' pane.
Described in section:
→ section "Parameter list", page 785

Recipe

This object can be used to display a table on the screen.
Inserts a box as a placeholder that is then
Filled with a recipe and configured as necessary using the Properties 'Recipe' pane.
Described in section:
→ section "Recipes", page 736

Alarm List

This object can be used to display a list of messages on the screen in table format.
Inserts a box as a placeholder that is then
filled and configured as necessary using the Properties 'Alarm List' pane.
Described in section:
→ section " Alarm list", page 867

List box

This object can be used to display a list box with values on the screen.
The object's properties can be configured as necessary using the Properties 'List box' pane.
Described in section:
→ section " List box - used to enable users to select options", page 397

Pre-defined field functions

Container

This object can be used to insert an element for grouping individual objects together into the screen.
Inserts a box as a placeholder that is then
filled with user controls and configured as necessary using the Properties 'Container' pane.
Described in section:
→ section "Using containers", page 294

Help Information

A Help information object enables the display of Help texts and Help pictures for a Help button or tag. The Help information for tags is defined in the Help Manager.
Described in section:
→ section " Help information", page 433

UserControls

UserControls

This drop-down menu can be used to select your own user controls, provided they have already been created in the project tree.
The name of the object used most recently will be shown as the text for the button.
The selected user control can be added directly by dragging&dropping it directly onto the screen. User controls can be nested and reused.

If the project does not contain any user controls, this object will be disabled (gray) and the text underneath it will say "Not available."
Described in section:
→ section "UserControls", page 267→ section "Nesting user controls", page 288

Arrange

Positioning options, layers, and groups are all available as ways to arrange the objects on the various screens.
Please note that these commands will only be available if you have previously selected multiple objects on the screen.

Location

This button can be used to align the selected objects within the screen relative to the bottommost selected object.
This function will only be available if multiple objects on the screen have been selected

▾ The following alignment options will be available:

- Left
- Middle
- Right
- Up
- Middle
- Bottom

The Controls Manager can be used to define the layer to which each object should belong.

More specifically, the order of the objects in the Controls Manager pane defines the virtual arrangement of the objects on the screen, i.e., the object on the bottommost layer will be at the bottom of the list and the object on the uppermost layer will be at the top. When a new object is added, it will be added at the very top together with a new uppermost layer.
This order can be changed by dragging&dropping the objects as necessary.

Bring forward

Can be used to move selected objects on the screen and in the Controls Manager

▾ The following options will be available for selection:

Bring forward
Moves the object one position up

Bring to front
Moves the selected object all the way to the top

Send backward

Can be used to move selected objects on the screen and in the Controls Manager

▾ The following options will be available for selection:

Send backward
Moves the object one position down

Send to back
Moves the selected object all the way to the bottom

Group

This function will only be available if multiple objects on the screen have been selected

▾ The following options will be available for selection:

Group

Ungroup

Tab Device

As soon as a project is created, the Device tab will show buttons that can be used to access the various input and selection dialog boxes used to configure connections via and to external components.

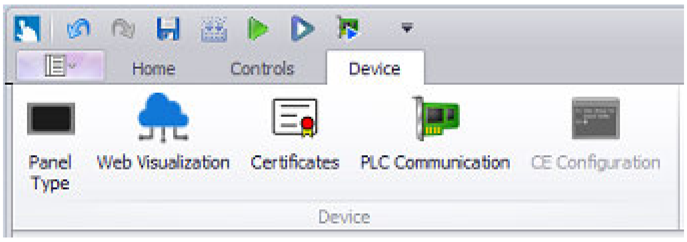


Figure 31: Device tab

Panel Type	Can be used to change devices Opens the Panel dialog box Described in section: → section "Settings for communicating with devices", page 220
Web Visualization	Option for the touch panel in the XV300 series, the XP500 industrial PCs and GALILEO OPEN applications Can be used to access a web service that makes it possible to make an existing project available not only on a local device, but on other devices on the Web as well Described in section: → section " Web visualization", page 1070
Certificates	Certificates for authenticating the applications on the device GALILEO Version 10.5 and higher Opens theCertificates dialog box Described in section: → section " Certificate manager", page 1091
PLC Communication	Used to configure an existing project for an additional PLC Opens the PLC Selection & Configuration dialog box Described in section: → section "Settings PLC Configuration", page 585
CE Configuration	Enabled only for panel models with Windows CE Opens the → " CE Configuration", page 228 Described in section: → section "Settings for communicating with devices", page 220

→ "Compile and transfer", page 1035

→ "Settings for communicating with devices", page 220

1.6.1.12 Control elements & User guidance

GALILEO uses an entirely new user interface that no longer features the old menu structures.
The usual menus and functions have now been replaced by a ribbon with tabs.

GALILEO is intuitive, easy to learn, and supports users with powerful user help and guidance functionalities. In addition, GALILEO features convenient functions that can be used to customize the user interface. Following is a brief description of the various features and elements that are part of this new design:

User interface designed to assist users

→ section "Quick access to controls and tag properties", page 93

→ section "Property changes are shown immediately", page 95

→ section "Paste special function for copying

→ section " Quick access menu - help you to work faster", page 110

→ section "Selecting multiple visualization objects at once", page 112

→ section "Grouping objects", page 113

User interface designed to assist users

conditions", page 95

→ "Advanced Text handling", page 98

→ "Text management compiler", page 100

→ "Media - Using images and videos in a project", page 100

→ "Icon library", page 101

→ "Importing tags ", page 103

→ "Dragging and dropping& from the project tree ", page 104

→ "Touch gestures", page 106

→ "Reusing user controls", page 107

→ "Positioning assistance", page 114

→ "Project organization and overviews", page 125

→ " Table overview ", page 126

→ "Controls Manager", page 127

→ "Real-time full-text search", page 127

→ "Find all references", page 128

→ "Working with styles ", page 123

→ "Keyboards", page 123

→ "Value Conversion Manager", page 124

→ "Window, Assistants (Wizards) and dialog boxes", page 130

→ " Undo and Redo", page 130

→ "Tooltips", page 131

Working efficiently

→ " Project as...", page 115

→ "Archive...", page 117, → "Unzip archive...", page 119

→ "What to do when a project crashes - restoring project files ", page 119

→ "Copying between projects - Ctrl+C/Ctrl-V ", page 119

→ "Convert project ", page 122

Other features

→ "Calling dynamic link library (DLL) functions", page 131

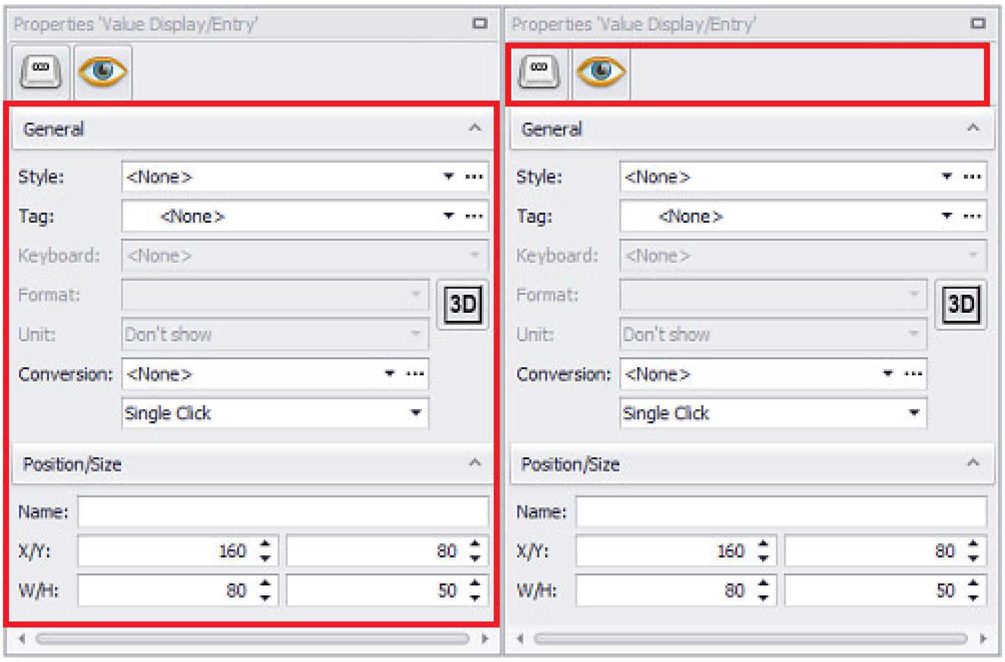
Quick access to controls and tag properties

GALILEO features two different menus for tags and visualization objects.

The quick access pane will appear in the sidebar as soon as you click on a tag or visualization object. This pane will contain the properties that are most frequently required for the selected object.

If you want to access all the properties for the object instead, you can access the Extended properties dialog box by double-clicking on the object or clicking on the icons at the top left of the quick access pane.

Properties 'Value Display/Entry' pane



Properties 'Value Display/Entry'

General

Style: <None>

Tag: <None>

Keyboard: <None>

Format:

Unit: Don't show

Conversion: <None>

Single Click

Position/Size

Name:

X/Y: 160 80

W/H: 80 50

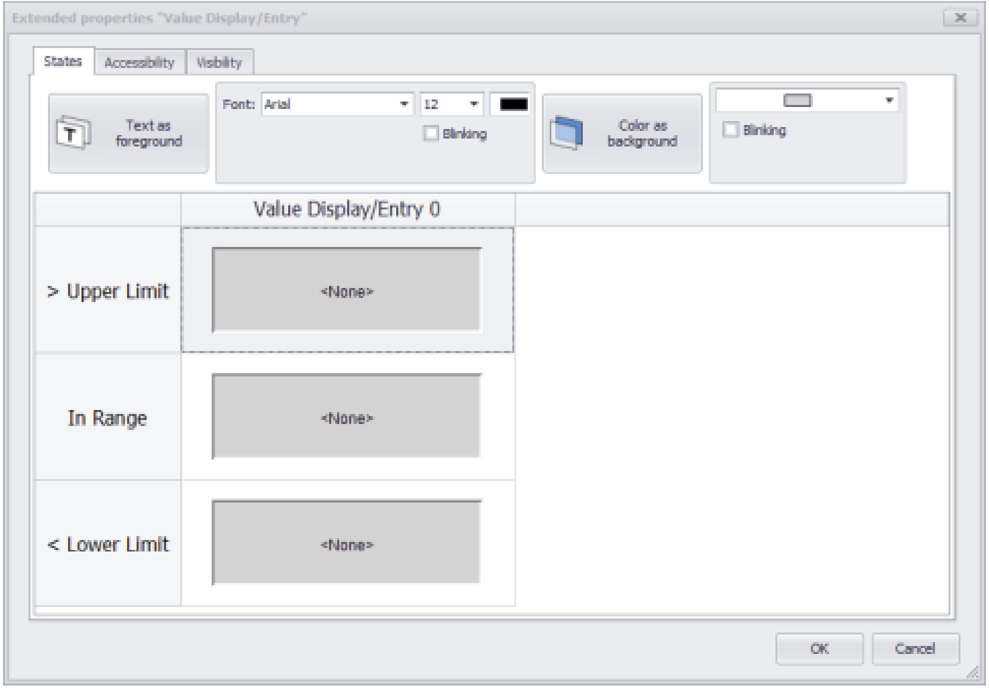
Quick access provided by the Properties pane

The Extended properties dialog box can be accessed by clicking on the icons

The Extended properties dialog box will contain all the properties for the selected object.

Every object will have the same basic extended properties, e.g., visibility, as well as specific customization options that only apply to the specific object in question and that will be available on the various tabs in the Extended properties dialog box for the object.

Double-clicking on the object / detailed menu



Extended properties "Value Display/Entry"

States | Accessibility | Visibility

Text as foreground

Font: Arial 12

□ Blinking

Color as background

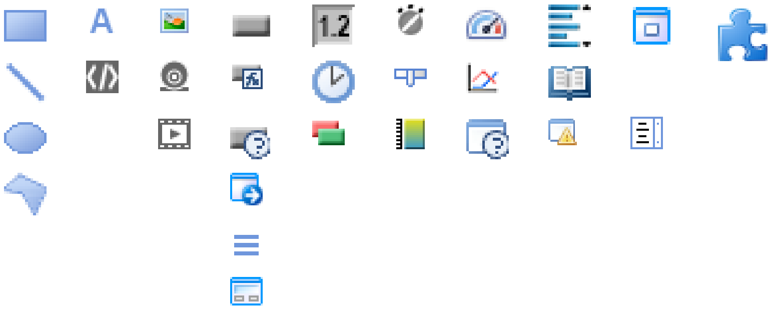
□ Blinking

	Value Display/Entry 0
> Upper Limit	<None>
In Range	<None>
< Lower Limit	<None>

OK Cancel

Figure 32: Extended properties dialog box for Value Display/Entry

For a description of the various available settings and options, please refer to the sections for the specific objects you want.



Property changes are shown immediately

If the visual properties for a visualization object are modified, the workspace will change immediately in order to reflect this.

If any properties are changed in the Properties pane for an object, e.g., the size of a button, the changes will be applied and shown immediately without requiring any additional confirmation.

Paste special function for copying conditions

The paste special function can be used to copy a visualization object's accessibility and visibility conditions to another object.

Paste special example

Prerequisites

You must have already created three internal tags: **Access** and **Visible**, of data type bit, and **Text**, of data type word.

Task definition

Say that a hydraulic press has a Value Display/Entry control for pressure changes. Due to safety reasons, this control is only visible, and can only be operated, if certain conditions are met.

Now say that a new hydraulic press is added to the manufacturing plant and you want an identical Value Display/Entry control for this new press.

Realization

- ▶ Create a Value Display/Entry object and assign a word tag called **Value** to it.
- ▶ Select the Numeric keyboard for the Value Display/Entry object.
- ▶ Add a bit tag called Access for the **Value Display/Entry** object's accessibility. The Value Display/Entry object should only be accessible if **Access** is ON.
- ▶ Add a bit tag called Visible for the **Value Display/Entry** object's visibility. The Value Display/Entry object should only be visible if **Visible** is ON.

An overview of the functions for Value Display/Entry objects is provided in a different section – please refer to → " Value Display/Entry", page 506.

- ▶ Now create an additional Value Display/Entry object and assign the **Value** tag to it.
- ▶ Select the Numeric keyboard for the Value Display/Entry object.
- ▶ Right-click on the first Value Display/Entry object you created.
- ▶ In the context menu, click on the Copy option.

Context menu/Copy

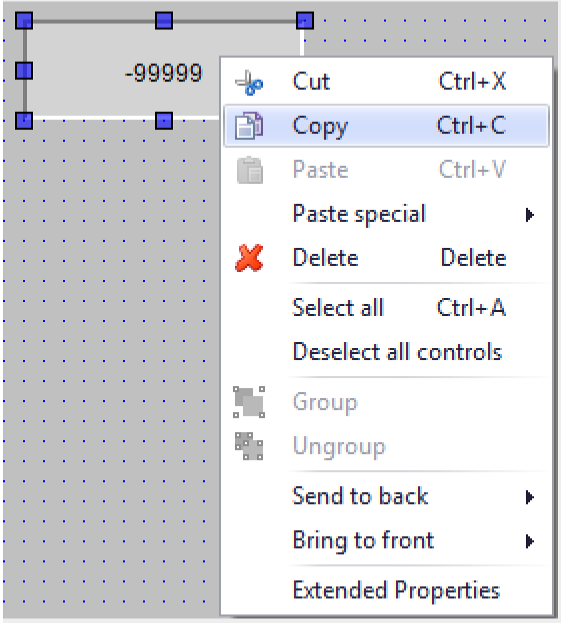


Figure 33: Copying before using the paste special function

- Right-click on the second Value Display/Entry object you created.
- Move the cursor to Paste special and select the <Paste accessibility> option.

Context menu/Paste special/Paste accessibility

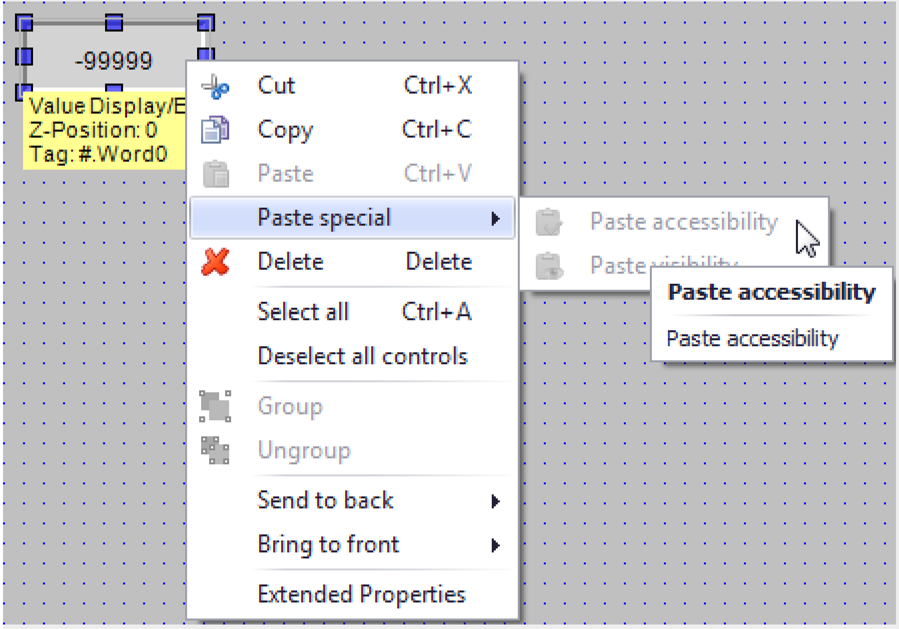


Figure 34: Paste special, accessibility

- Right-click on the second Value Display/Entry object you created.
- Move the cursor to Paste special and select the <Paste visibility> option.

Context menu/Paste special/Paste visibility

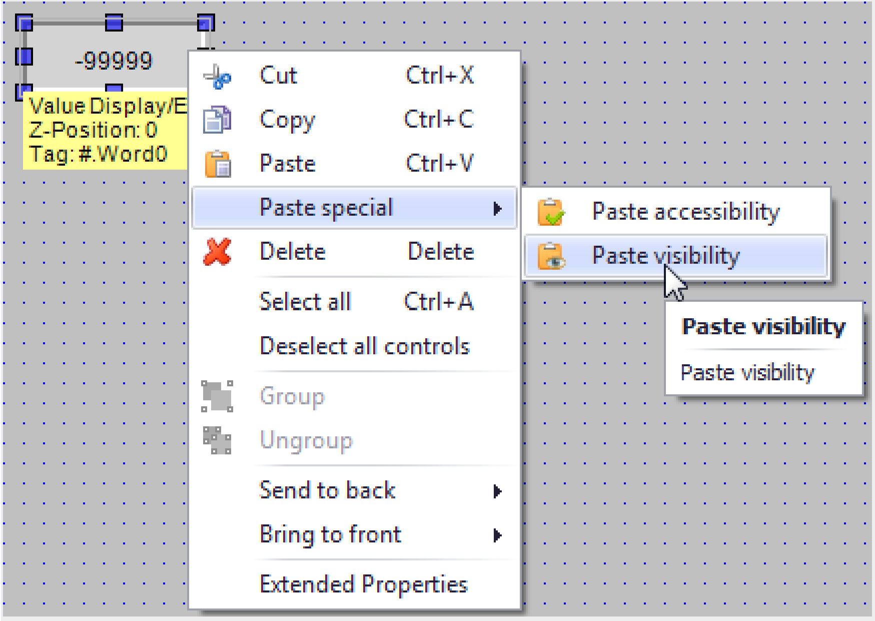


Figure 35: Paste special, visibility

Advanced Text handling

In order to make text input more efficient, GALILEO features an autocomplete function that helps avoid unnecessary text duplicates.
This autocomplete function will take its texts from the texts in the Text Editor.

If, for example, you are entering text for a button's label, a drop-down menu will open and offer existing texts for selection in real time while you are typing (the texts shown will match the letters you have already entered).

▶ Click on one of the texts in the list.

The text will be copied to the field.

If none of the texts are what you want:

▶ Enter the full text you want into the field.

▶ Click on OK.

A text entry will be created with the text you enter and will immediately be available for selection in the list of texts.

Double-click on the Text object

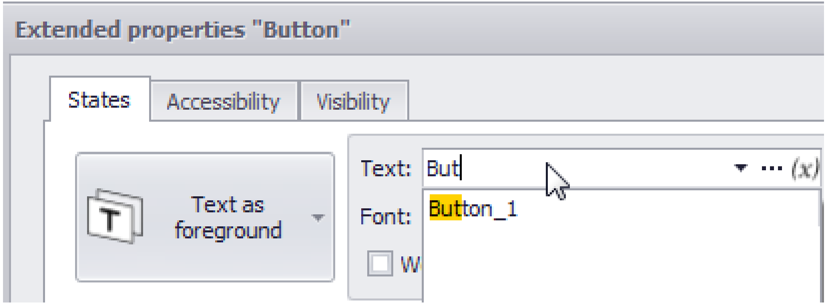


Figure 36: Drop-down menu when entering text

You can also select a text using its text ID.
To do so, enter the text ID into the text box. If the text ID exists, the corresponding text will be listed and can be entered by pressing the RETURN key on the keyboard or by clicking on it.

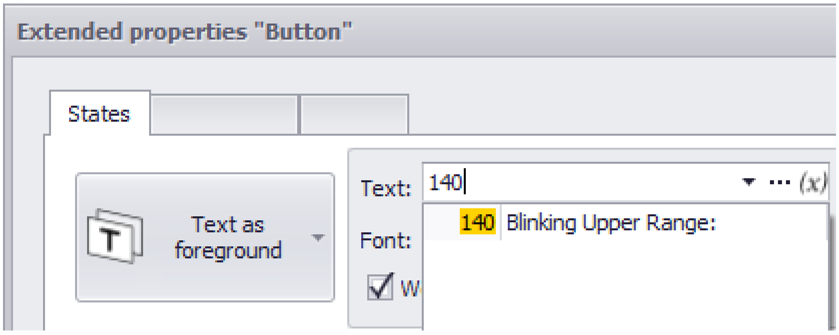
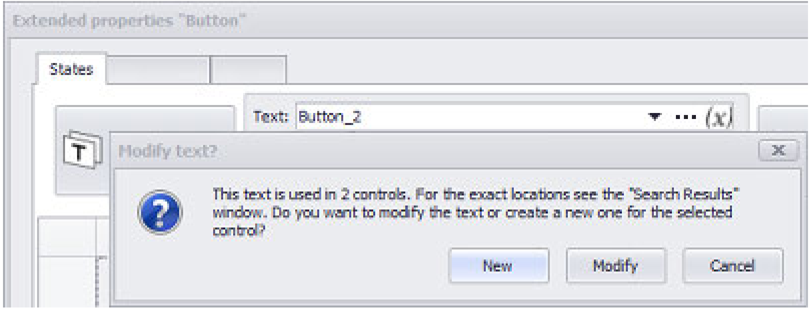


Figure 37: Text entry via ID

If you have already assigned a text to a visualization object, you can change it if necessary.

If an existing text is changed, the program will check whether the text is being used more than once. The Modify text prompt will appear and ask whether you want to use the changed text as a New text or whether the change should be applied to all the objects using the original text.

Double-click on object/text change

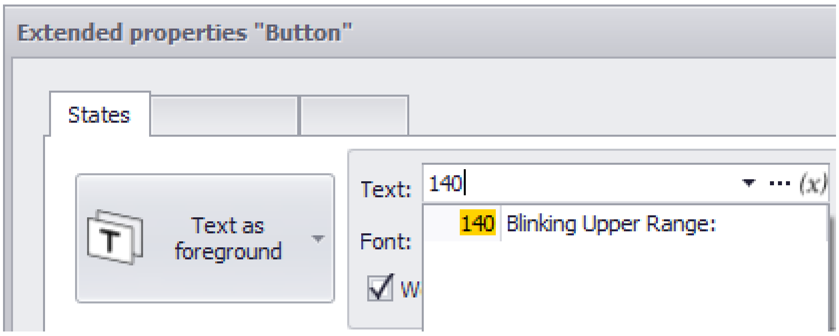


This text is used in 2 controls. For the exact locations see the "Search Results" window. Do you want to modify the text or create a new one for the selected control?

Figure 38: Modify text? Prompt shown after changing a text that is being used multiple times

You can also select a text using its text ID.
To do so, enter the text ID into the text box. If the text ID exists, the corresponding

text will be listed and can be entered by pressing the RETURN key on the keyboard or by clicking on it.



Extended properties "Button"

States

Text: 140

140 Blinking Upper Range:

Figure 39: Text entry via ID

Text management compiler

The GALILEO compiler will output warnings if the translation for a text is missing. It will also output warnings if a text object is too small for the text assigned to it It will also output warnings if a text object is too small for the text assigned to it (this applies to text translations as well).

Additional compiler functions are described in other sections.

See also

→ "Languages and texts", page 630
→ section "Configuration for Build and Deploy", page 1056

The appendix lists the various → "System Messages", page 1147.

Media - Using images and videos in a project

The Media tab in the project tree must be used to import all the images and videos being used in the project.

The project image must be in one of the following formats: *.png, *.jpg/*.jpeg , *.tiff/*.tif, *.gif , *.svg or *.bmp,

image sequences with the *.gif format.

The project video must be in one of the following formats: *.mp4, *.mov or *.avi.

Icon library images are in .svg format.

There are two ways to add media to the project tree:

Copying media files to the project by dragging and dropping them from Windows Explorer

- In the project/Media tree, open the Project images folder.
- Open Windows Explorer.

- Drag the media files you want from Windows Explorer and drop them onto theProject images folder.

Importing media files

- In the Project/Media tree, open the Project images folder.
  If necessary for organization purposes, create a new subfolder.
- Select this (sub)folder (e.g., *Button*) and open the context menu.
- Click on *Import images*.
- In the dialog box that appears, select the image file you want.
- Once you have selected the image, click on Open.

When you click on "Open," the image will be converted to PNG format. GALILEO uses only png-images in the project

Project\Media

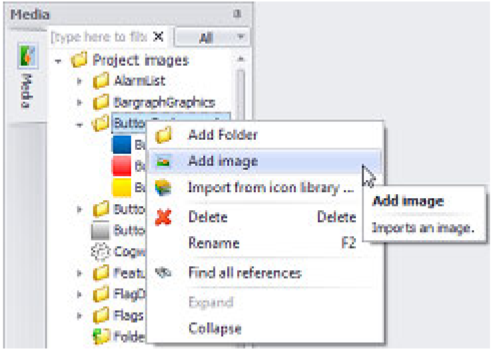


Figure 40: Importing image files

You can drag
media from the project tree in its original size and drop it onto the screen in the position you want.

See also

→ "Project tree", page 64, → " Media", page 68
→ section "Substep 1: Adding an image (medium) to the project tree", page 174

Icon library

GALILEO features an icon library that can be accessed from the project tree's Media tab.

As soon as an icon is imported, it will become an image that is independent from the library, meaning that it can be used just like any other image (including editing, for example).

Importing from the icon library

- In the Project/Media tree, open the Project images folder.
  If necessary for organization purposes, create a new subfolder.

- Select this (sub)folder and open the context menu.
- Select *Import from icon library ...*.
- In the dialog box that appears, select the icon you want.
- Confirm with Import.

Project\Media

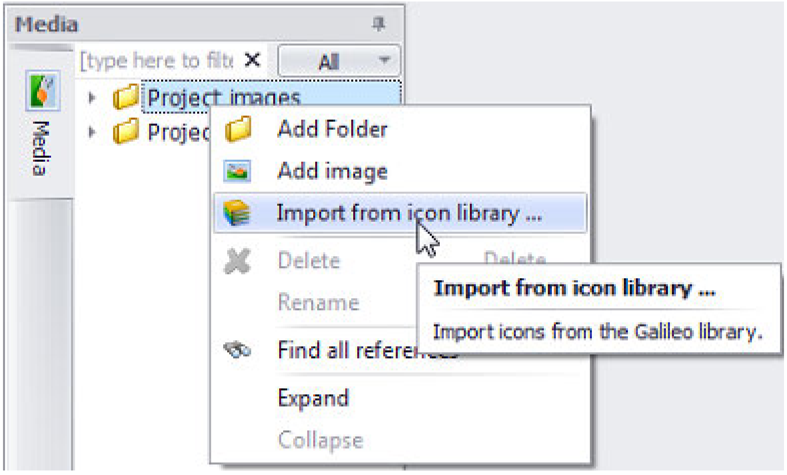


Figure 41: Importing from the icon library

Icon Library dialog box

Project Tree\Media\Project images\Import from icon library...

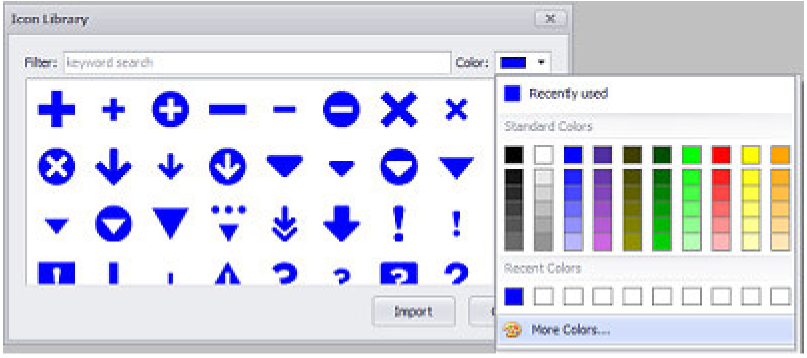


Figure 42: Icon Library dialog box

Filtering:	keyword search As soon as your start typing, the icons being displayed will be filtered accordingly. Deleting the text in the Filter field will make all the icons be displayed again.
Color:	You can choose one → "Color choice - Object's Color Palette", page 234 per import.

You can use the scrollbar to go through all the available icons.

To select multiple icons:

- You can select multiple individual icons by holding down the Ctrl key.
- You can select a contiguous range of icons by holding down the Shift key.

Project Tree\Media\Project images\Import from icon library...

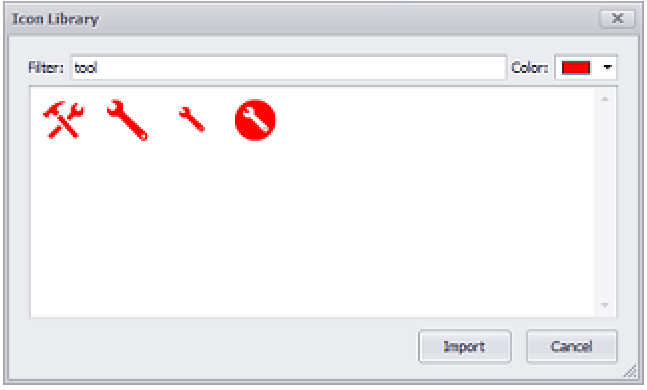


Figure 43: Filter in Icon Library dialog box

Importing tags

Tags can be imported into the project tree by using the context menu for the Tags tab pane.

Tags tab in the project tree

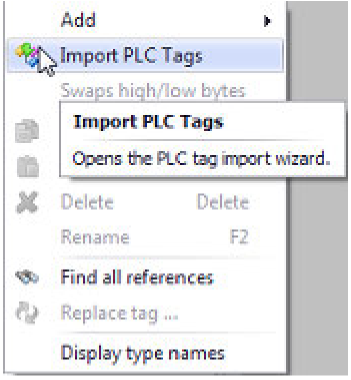
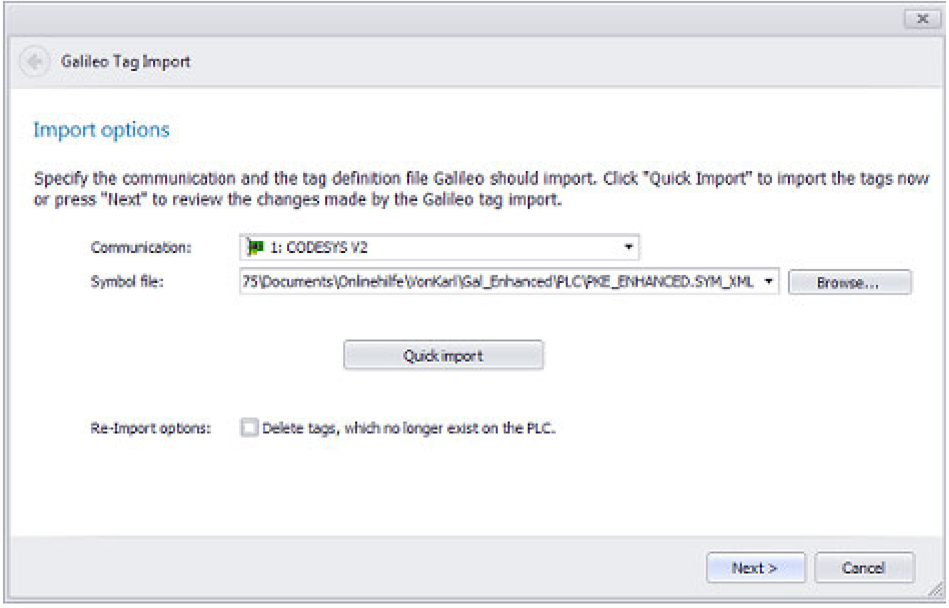


Figure 44: Tags context menu

The wizard will open in a new dialog box on your monitor.

Tags tab in the project tree



Galileo Tag Import

Import options

Specify the communication and the tag definition file Galileo should import. Click "Quick Import" to import the tags now or press "Next" to review the changes made by the Galileo tag import.

Communication: 1: CODESYS V2

Symbol file: 75\Documents\Onlinehilfe\VonKarl\Gal_Enhanced\PLC\PKE_ENHANCED.SYM_XML

Re-Import options: □ Delete tags, which no longer exist on the PLC.

Figure 45: Wizard Tag import

Clicking on the Quick import button will automatically import all tags.

Otherwise, you can select a symbol file and then click on Next> to run a custom import and import tags and properties in a targeted manner.

You can use the wizard's import preview to make sure that the tags will be imported exactly the way you want.

Tags tab in the project tree

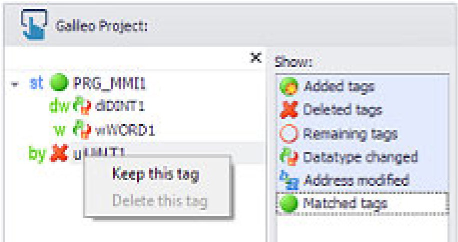


Figure 46: Wizard Tag import, Preview

For detailed information on importing tags, please refer to:→ section "Working with tags", page 539

Dragging and dropping& from the project tree

Various elements in the project tree's Tags, Media, Scripts, and Screens tabs can be dragged&dropped while holding the right or left mouse button.

Left-clicking will offer the most likely function, such as Add, while right-clicking will show all available functions.

Drag&Drop menu left

One way to assign tags to visualization objects in the program is to drag&drop them onto the objects you want. To do so, simply click on the tag, hold the mouse button down, and then drop the tag on top of the visualization object you want.

You can also add media and UserControls to a screen by simply dragging & dropping them from the project tree.

The following visualization objects will be automatically generated when you drag & drop certain elements with the left mouse button:

Element from project tree	Visualization object generated on the screen		
Standard Masks	Screen Change		Already linked to the screen as a target screen
User Controls	→ section "Dragging and dropping user controls", page 272		
Help Screens	Help Button		Already linked to the help screen
Printreport	Function key with the Print Report function		Already linked to the print report
Script	Function key with the Call Script Function function		Already linked to the script
Tag data type			
BIT	Button		Already linked to the tags
BYTE, WORD, DWORD, FLOAT	Value Display/Entry		
STRING	Value Display/Entry		

Drag & drop menu right

If you drag elements onto the screen while holding down the right mouse button, a context menu with options will appear when you drop the element.

Element from project tree	Context menu shown		
Standard Masks	Same as using the left mouse button		Already linked to the screen as a target screen
UserControls	→ "Dragging and dropping user controls", page 272		
Help Screens	Set as help screen Help Button		Already linked to the help screen
Printreport	Same as using the left mouse button		Already linked to the print report
Background screens	Set as background screen		Already linked to the standard screen
Script	Function key with the Call Script Function function Set as entry script Set as exit script		Already linked to the script
Tag data type			
BIT	Value Display/Entry Button Flag Display Switch		Already linked to the tags When using tag arrays, a matrix will be shown after selecting the object so that you can create

			objects.
BYTE, WORD, DWORD,	Value Display/EntryButton Flag Display Switch Slider Bargraph Gauge		
FLOAT	Value Display/Entry Button Slider Bargraph Gauge		
STRING	Same as using the left mouse button for Value Display/Entry		

For tags, scripts and arrays

Tag drag&drop example

Say you want a speed to be displayed both as a bargraph and in a Value Display/Entry object.

▶ Create a tag of data type word called Speed. For information on how to work with tags, please refer to Working with tags.

▶ Create a bargraph on the workspace. For information on how to work with bargraphs, please refer to Using bargraphs to display values.

▶ In the project tree, click on the Speed tag and hold the mouse button down.

▶ Move the tag over the bargraph you just created and drop it there.

▶ In the project tree, click on the Speed tag and hold the mouse button down.

▶ Move the tag to a location on the screen below the bargraph and drop it there.

Touch gestures

GALILEO supports touch gesture controls:

- Drag
  Large lists, such as parameter lists, alarm lists, and recipes, can be completely scrolled through with one finger.
- Panning
  If an image is too big for the corresponding display, it can be moved around by pressing on it with a finger and dragging it around. This can be done with graphs, for example, in which case it will only be possible to move the graph within the defined display area.
- Pinch close/Pinch open
  You can zoom into images and graphs or into curves (e.g., the X-axis of graphs) in order to view contents in greater detail.
  This gesture requires the use of more than one finger at the same time – for instance, touching the screen with your thumb and index finger and spreading

them apart will zoom in.
Please note that the object element (e.g., image) cannot be made smaller than the defined display area.

- Flick
  If there are multiple user controls next to or on top of each other (e.g., in a container) and the display object is too small, they can be moved around with a horizontal or vertical flick gesture on the screen, within the defined display area.
- Double tap
  This will restore the object's (e.g., image) original appearance or show the image in twice the resolution.

→ These touch gestures can only be used with PTC devices such as the XV300 and XP500.
This function is also supported by GALILEO Open applications.

Scrolling trough the user help does not count as a touch gesture within this context.

Reusing user controls

UserControls are ideal for grouping visualization objects that are constantly reused throughout a project at several locations or on several screens, e.g., headers, footers, and recurring object groups.
If necessary, UserControls can be used multiple times in all types of screens.

A standard screen can contain various user controls.

These user controls can be used to put together your own custom object groups in a modular manner and reuse these groups as many times as necessary.

Reusing your own custom user controls as a sort of module in other user controls ensures that you will be able to make changes to the project quickly and easily later on.

UserControls can also be used to enable scrolling on a panel screen.
The corresponding swipe gesture can be supplemented with scroll bars if necessary.

These touch gestures can only be used with PTC devices such as the XV300 and XP500.
This function is also supported by GALILEO Open applications.

→ These touch gestures can only be used with PTC devices such as the XV300 and XP500.
This function is also supported by GALILEO Open applications.
These touch gestures can only be used with PTC devices such as the XV300 and XP500.
This function is also supported by GALILEO Open applications.

UserControls can be shown and hidden at runtime.

To integrate scrolling, follow the steps below:

- ▶ Create both a standard screen and two or more UserControls with contents that you want the user to be able to see by scrolling.

For more information on how to create and insert standard objects and UserControls, please refer to: → section "Your own first project", page 141

- ▶ Now insert all the UserControls you want on top of each other on the standard screen.
- ▶ Click on the topmost object.

The Properties 'Container' pane will open.

- ▶ Use the ▾ drop-down menu under General to individually select each user control you created
- ▶ Use the Add button to insert the Contained UserControls table.

Properties 'Container' pane

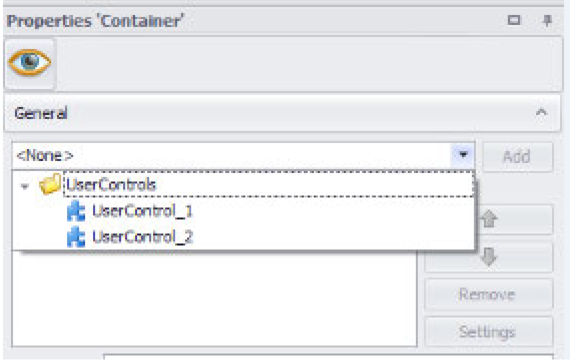
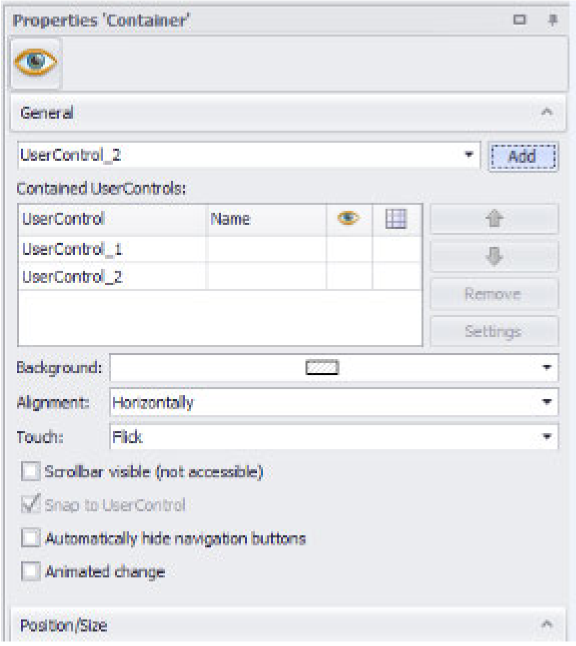


Figure 47: Selecting a user control

Properties 'Container' pane



Properties 'Container'

General

UserControl_2 ▾ Add

Contained UserControls:

UserControl	Name		
UserControl_1			
UserControl_2			

Background: ▾

Alignment: Horizontally ▾

Touch: Flick ▾

□ Scrollbar visible (not accessible)

☑ Snap to UserControl

□ Automatically hide navigation buttons

□ Animated change

Figure 48: Defining the order of the controls in the table

- Use the Alignment: ▾ drop-down menu to select the direction for the scrollbar. The following options will be available:Horizontally and Vertically.
- Use the Touch drop-down menu to select the type of gesture you want. The following options will be available:Flick, Drag, and No Touch The option you select will determine which additional options will be available.
- If the Scroll bar visible option is ☑, a scroll bar will be shown as appropriate for the selected direction.

The Snap to UserControl option is used to define what the transition between two screens should be like when scrolling.

If the Snap to UserControl option is ☑, the image will jump to the screen that has become the most visible as a result of scrolling.

If the Snap to UserControl option is □, the image will stay where it was when the user stopped scrolling.

You can use the Properties pane for a user control in order to define a parameter that can then be used for individual controls.

This parameter consists of a "Name" and "Type," and must be filled out in order to be used (in a container or as a user control). Once you enter this parameter information, it will be listed in the "Parameters" tab as a name and the corresponding drop-down menu.

A container can reference multiple user controls. In this case, these user controls will be the individual entries in the container. It is not absolutely necessary to assign a name to each entry in order to

uniquely identify it. Instead, each entry has its own parameters that need to be entered here.
In other words, if you were to assign a second user control to a container, you could use the combo box after "Entry" to select the appropriate entry.
Once you do this, the parameters for the entry you want will be shown.

For detailed information on how to work with screens, please refer to:→ section "Using containers", page 294.

💡 Quick access menu - help you to work faster

Right-clicking on the 💡 icon will open an object-specific context menu with select commands that make it possible to work more quickly with the selected object.

This function is ideal for speeding up the process of creating function keys for a visualization object. It can be used with any visualization object on which functions can be carried out. These objects include the following:

- Graph Block
- Parameter list
- Recipe
- Alarm List
- UserControls
- Container
- Screen Change

To use the 💡 context menu, follow the steps below:

▶ Move the cursor over one of the visualization objects listed above.

The 💡 icon will appear on the visualization object's top right corner.

▶ Move the cursor over the 💡 icon.

A list with the most important functions for the selected visualization object will appear.

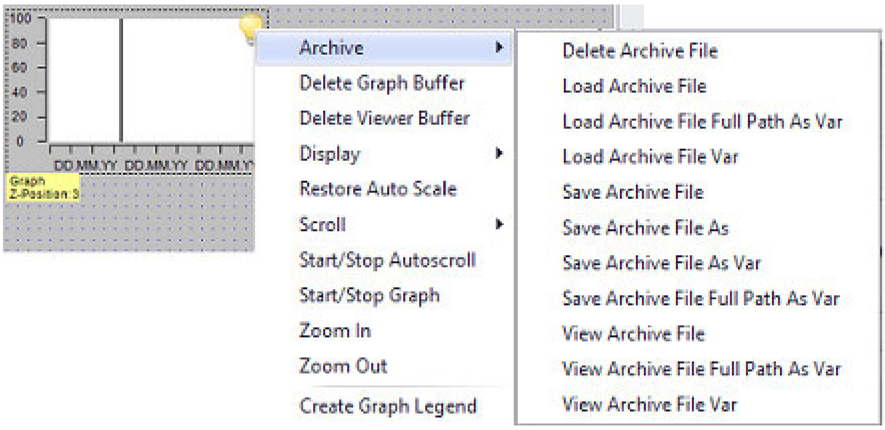
▶ Select one of the functions.

A function key with the function you selected will be created automatically.

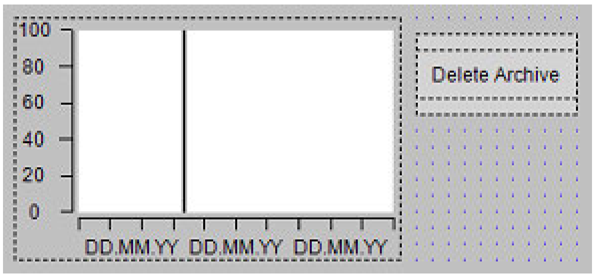
→ The texts for the supported functions for each individual object will be shown in English.

Quick access menu on visualization objects

If you move the cursor over the light bulb icon on a visualization object, all the special functions available for that object will be shown. In order to ensure that they are clearly laid out, these special functions are grouped together as appropriate. For example, for the Graph visualization object, all special functions related to archiving will be grouped under Archive.



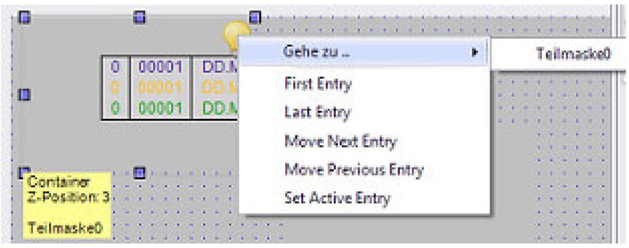
If you select a special function from the context menu, the function will be automatically added as a button on the screen. This is much faster than the alternative, which is to select the Function Key object in the *Controls* tab, place it on the screen, select the Graph group, and then select the function you want, e.g., Delete Archive File.



Quick access menu for UserControls and containers

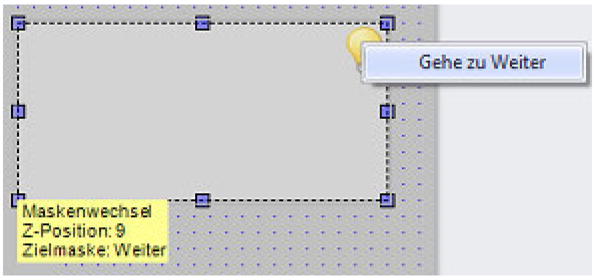
UserControls and containers found on screens also feature a quick access menu allowing for quick navigation.
By using the Jump to... option, you can access the selected UserControl or container from the main screen.
The context menu will also provide special functions for faster positioning on the screen:

```
First Entry
...
Set Active Entry
```



Quick access menu for screen changes

The quick access menu makes it possible to quickly navigate to the defined target screen. In order for this to be possible, however, a target screen must have already been defined for the screen change; e.g., <Next>.



Selecting multiple visualization objects at once

In order to reduce the amount of work involved in configuring visualization objects, GALILEO features the option of selecting and editing multiple visualization objects simultaneously.

If multiple visualization objects are selected at the same time, the quick access pane will only show the properties that that can be configured for all the selected objects at the same time. Position X/Y, Size W/H, and visibility can always be configured simultaneously for all visualization objects. Other configurable properties will depend on the specific objects that have been selected.

If multiple visualization objects of the same type are selected, there will be a lot of properties that can be modified simultaneously, e.g., when configuring several buttons at the same time.

If different types are selected simultaneously, the number of properties that can be modified will be smaller, with the specific number depending on the actual types selected.

Properties that can be edited when multiple buttons are selected

- States
- Accessibility
- Style
- Tag
- Action
- Bit
- Click
- Pushbutton
- Frame

Properties that can be edited when buttons, bargraphs, and sliders are selected at the same time

- Tag
- Frame

To edit multiple visualization objects at the same time, follow the steps below:

- Click on a free spot inside the workspace and draw a rectangular marquee enclosing all the visualization objects you want to select. Please note that only visualization objects that are completely enclosed by the marquee will be selected.
- Alternatively: Press and hold down the Ctrl key and select all the visualization objects you want.
- Now configure all the properties you want that are available on the quick access pane.

Small square-shaped sizing handles will be shown at the outer edges of all selected visualization objects.

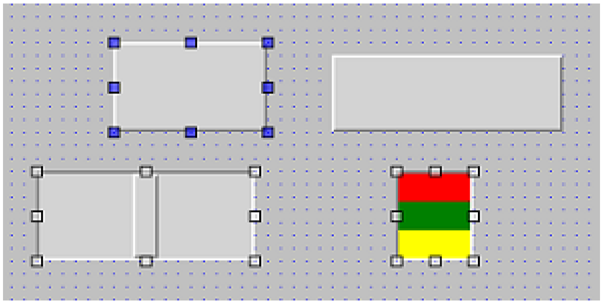


Figure 49: Selected objects

Grouping objects

Grouping visualization objects is a good idea when the objects need to be positioned relative to each other.
When grouped, visualization objects will be linked to each other in such a way that moving any of the objects in the group will move all the other objects in the group as well.

If you want to group two or more visualization objects together, follow the steps below:

- Select all the visualization objects that you want to group together.

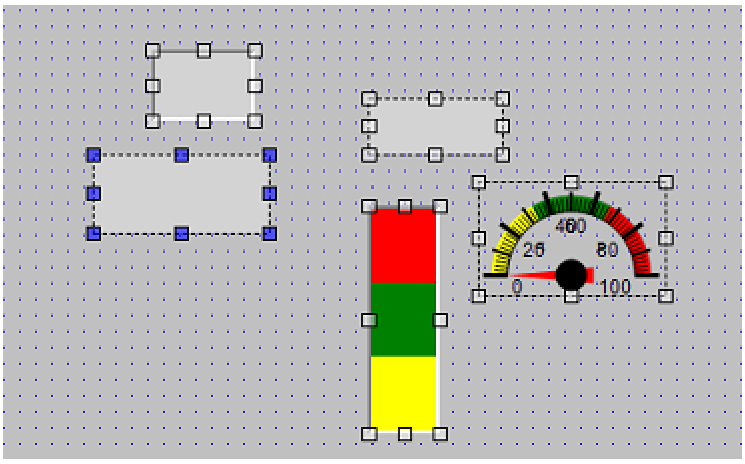


Figure 50: Selecting visualization objects

- Open the "Controls" tab.
- Select the Arrange\Group command.

Controls\Arrange\Group

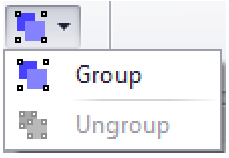
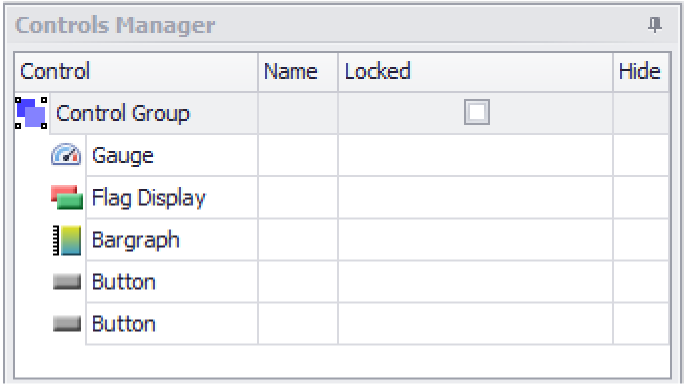


Figure 51: Grouping visualization objects

Grouped visualization objects will be shown as a group in the Controls Manager. In order to have the Controls Manager show the individual objects, follow the steps below:

- Double-click on the group in the Controls Manager

Controls Manager



Controls Manager

Control	Name	Locked	Hide
Control Group		☐	
Gauge			
Flag Display			
Bargraph			
Button			
Button			

Figure 52: Grouped visualization objects being shown in the Controls Manager

If you want to ungroup a grouped set of objects, follow the steps below.

- Select the group.
- Open the Controls tab.
- Click on Arrange\Ungroup.

Controls\Arrange\Ungroup

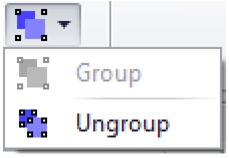


Figure 53: Ungrouping objects

Positioning assistance

The automatic positioning assistance feature helps align visualization objects as soon as they are moved on a screen.

As soon as an outer edge of the visualization object being moved is at the same height as the outer edge of another visualization object, pink lines will appear in order to indicate this.

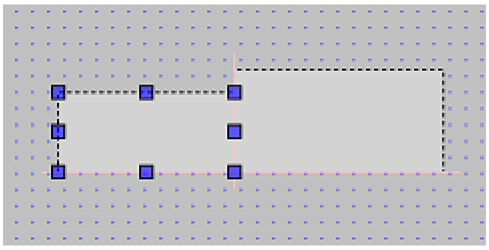


Figure 54: Automatic positioning assistance feature aligning a visualization object with the edge of another visualization object

Snap to grid

In addition, an option for aligning objects along the grid can be enabled. To align a visualization object with the grid, follow the steps below:

- ▶ Open the "Controls" tab.
- ▶ Click on "Snap to grid"

Controls\Snap to grid

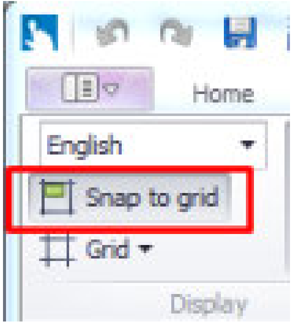


Figure 55: "Snap to grid" positioning assistance

- ▶ Now move the visualization object towards the position you want.

Whenever an outer edge of the visualization object intersects with a grid line, it will snap to it until it is moved to the next grid line.

Grid settings can be configured by going to → section "Grid Settings", page 212.

Described in section:

→ section "Project Configuration", page 196

Project as...

You can create a copy of your project by using the Save Project as...

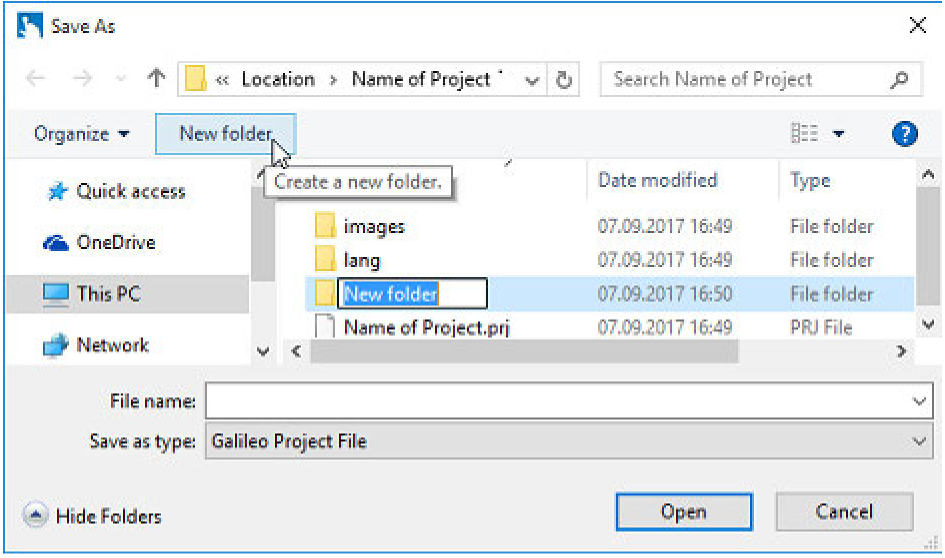
All the data for a project is saved in XML files, making it easy to manage with versioning software.

→ The folder for the "new" project needs to be created before saving the project.

Perform the following instructions :

- Open the project that you want to copy in GALILEO.
- In the project management tab, click on the Save Project as... option to open the Save As dialog box
- Under Organize, go to the location where you want to save the copy.
- Once there, click on New folder.
  A new folder will be created in the location you selected previously.
- Name the folder.

Project Management\Save Project as...



Name	Date modified	Type
images	07.09.2017 16:49	File folder
lang	07.09.2017 16:49	File folder
New folder	07.09.2017 16:50	File folder
Name of Project.prj	07.09.2017 16:49	PRJ File

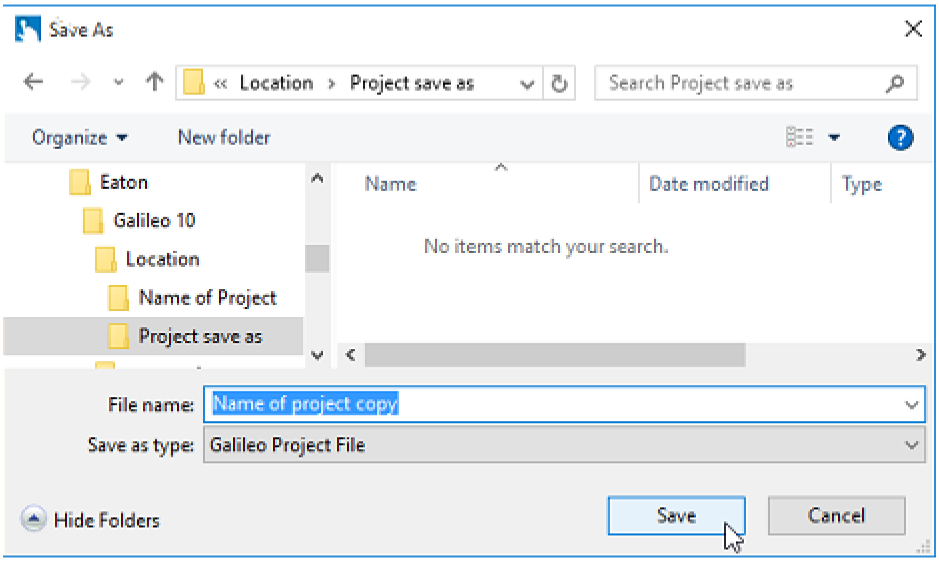
File name:

Save as type: Galileo Project File

Figure 56: Creating an empty folder

- Enter the filename you want.
- Create a copy of the open project by clicking on the Save button.

Project Management\Save Project as...



No items match your search.

File name: Name of project copy

Save as type: Galileo Project File

Figure 57: Creating a copy of the open project

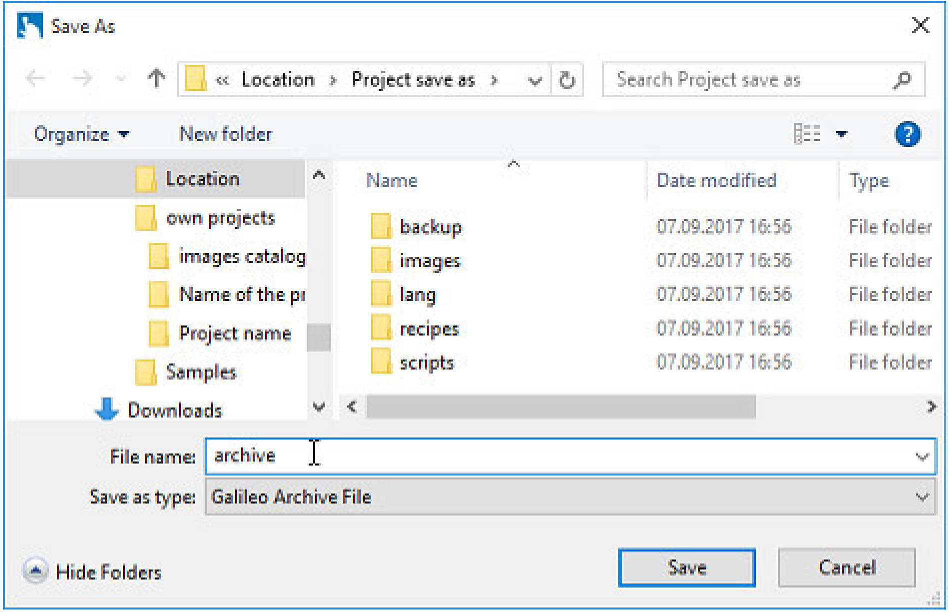
Archive...

The Archive... option in the project management tab can be used to create a *.zip archive of the open project.

Perform the following instructions :

- ▶ Open the project that you want to archive in GALILEO.
- ▶ In the project management tab, click on the Archive... option to open the Save As dialog box

Project Management\Archive...



Name	Date modified	Type
backup	07.09.2017 16:56	File folder
images	07.09.2017 16:56	File folder
lang	07.09.2017 16:56	File folder
recipes	07.09.2017 16:56	File folder
scripts	07.09.2017 16:56	File folder

File name: archive

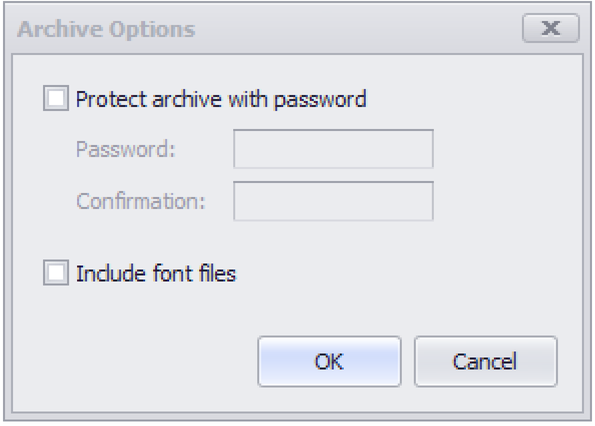
Save as type: Galileo Archive File

Figure 58: Entering the name for the archive

▶ Enter the archive name you want.

Enable the checkboxes (☑)for the options you want.

Project Management\Archive...



Archive Options

□ Protect archive with password

Password:

Confirmation:

□ Include font files

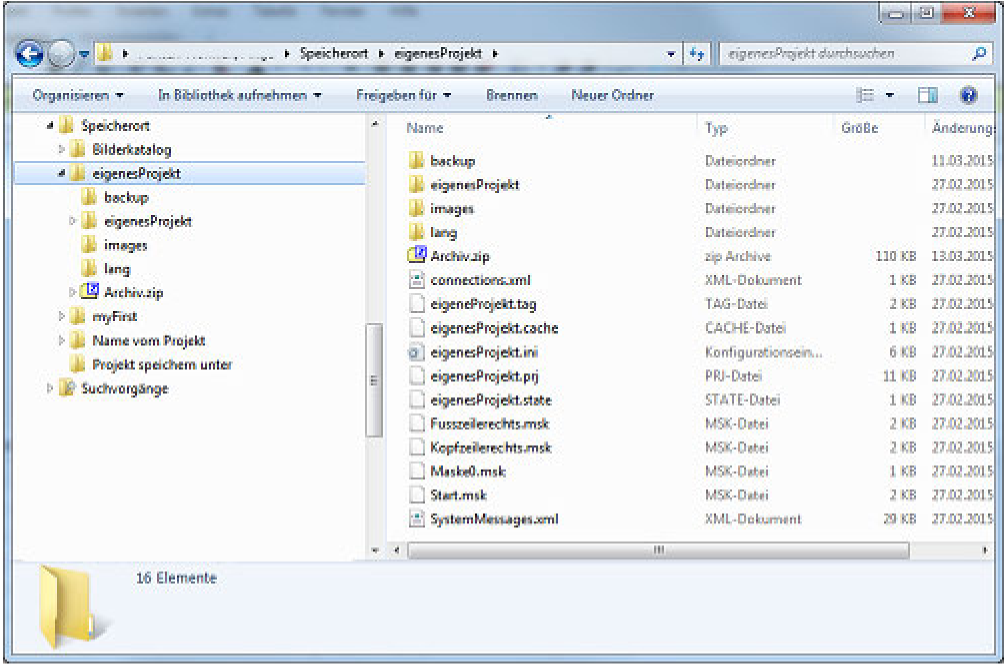
Figure 59: Selecting archive options

☑ Protect archive with password	Conditions for protecting the project file
☑ Include font files	The fonts used in the project will be saved together with the project.

Click on OK.

A prompt saying that the archive has been created successfully will appear.

Storing in the directory structure



Name	Typ	Größe	Änderung
backup	Dateiordner		11.03.2015
eigenesProjekt	Dateiordner		27.02.2015
images	Dateiordner		27.02.2015
lang	Dateiordner		27.02.2015
Archiv.zip	zip Archive	110 KB	13.03.2015
connections.xml	XML-Dokument	1 KB	27.02.2015
eigeneProjekt.tag	TAG-Datei	2 KB	27.02.2015
eigenesProjekt.cache	CACHE-Datei	1 KB	27.02.2015
eigenesProjekt.ini	Konfigurationsein...	6 KB	27.02.2015
eigenesProjekt.prj	PRJ-Datei	11 KB	27.02.2015
eigenesProjekt.state	STATE-Datei	1 KB	27.02.2015
Fusszeilerechts.msk	MSK-Datei	2 KB	27.02.2015
Kopfzeilerechts.msk	MSK-Datei	2 KB	27.02.2015
Maske0.msk	MSK-Datei	1 KB	27.02.2015
Start.msk	MSK-Datei	2 KB	27.02.2015
SystemMessages.xml	XML-Dokument	29 KB	27.02.2015

Figure 60: .zip in folder

Unzip archive...

A Galileo archive ZIP file created in GALILEO can be opened in the project management tab with the Unzip archive... command.

Perform the following instructions :

- ▶ In the project management tab, click on Unzip archive....
- ▶ In the directory structure, select the Galileo archive file (*ZIP) that you want to unzip.
- ▶ Select an empty folder where you want to extract the contents of the file.

Click on OK to unzip the file. This may take a few minutes depending on the size of the project.
After the file has been unzipped, the directory structure will open and show the unzipped project so that you can open it if you want.

What to do when a project crashes - restoring project files

GALILEO will save your project file to a backup file in the background at regular intervals. This <PROJECTNAME>.BAK backup file can then be used to restore corrupted project files, ensuring that you will only lose the work you have done since the last automatic backup.

The backup interval can be adjusted by going to *Home\Application Configuration\Options*, Described in section: → section "Application Configuration", page 135

If you try to open a project for which the project file is corrupted or was not closed properly, a message to this effect will appear.

When this happens, you can use the information specifying the date and time and the file size to decide whether you want to load the original file or the backup file.

→ General recommendation
Usually, you will want to open the backup file, as you can be sure that it has been saved correctly.

Copying between projects - Ctrl+C/Ctrl-V

Elements can be copied between GALILEO projects – both between multiple instances of the Galileo application and between different projects.

There is a special dialog box that can be used to copy visualization objects or screens from one project to another. When using this option, you can specify whether you want to copy referenced styles, texts, tags, and/or images.

To copy a screen from project A to project B, follow the steps below:

- ▶ In the project tree for project A, right-click on the screen that you want to copy in order to open the corresponding context menu.
- ▶ Click on Copy.

Project Tree\Screens\Context menu\Copy

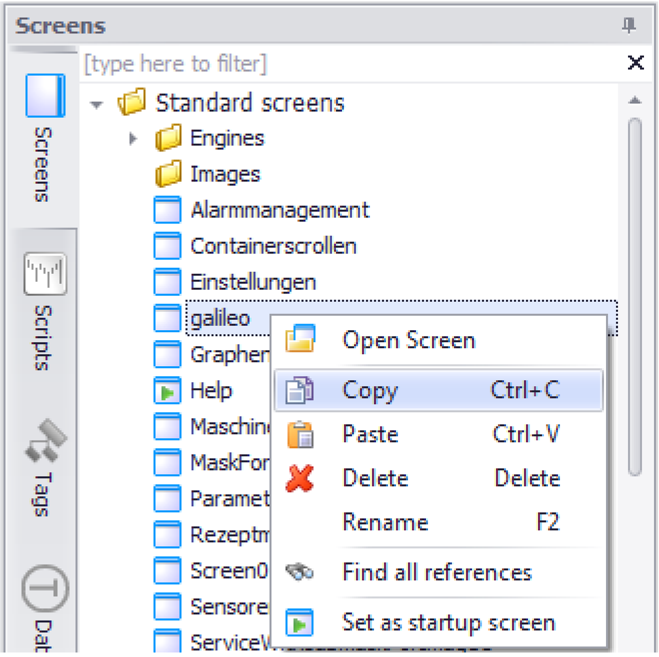


Figure 61: Copying a screen

- ▶ Switch to project B.
- ▶ Open the Screens tab in the project tree for project B and right-click on the subfolder corresponding to the screen type of the screen being copied. The context menu will appear.
- ▶ Click on Paste.

Project Tree\Screens\Context menu\Paste

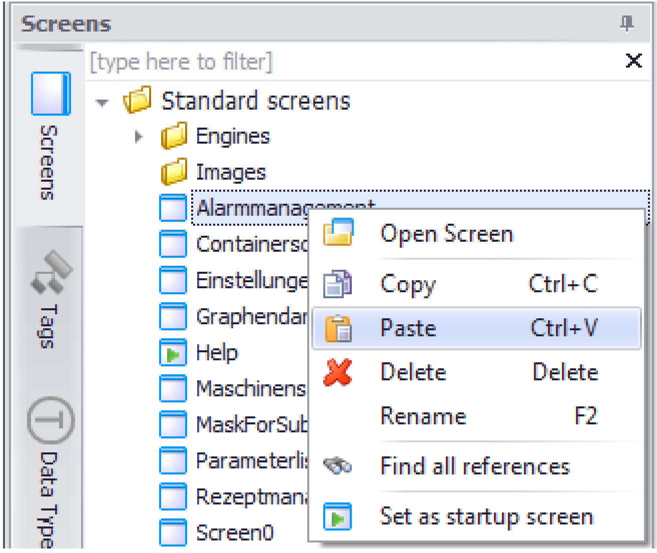


Figure 62: Pasting a screen from a different project

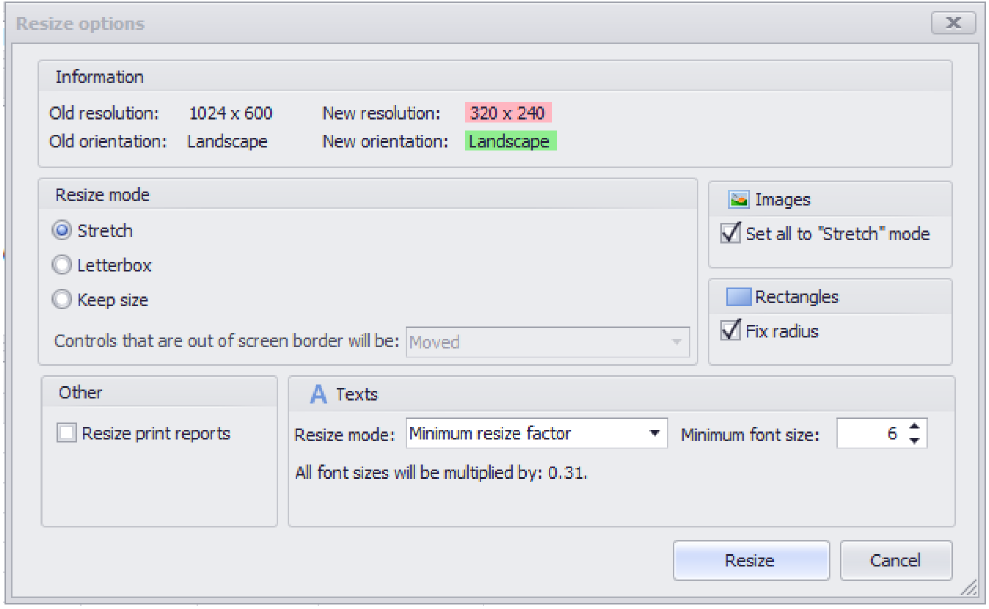
Projects with different display settings

If you try to copy a screen to a project with a different resolution or orientation, a dialog box asking what you want to do with the objects in the screen being copied will appear.

The resize options in the dialog box are described in the communication settings – please refer to:

→ section " Making changes to the display settings", page 226

Device tab\Panel Type



Resize options

Information

Old resolution: 1024 x 600 New resolution: 320 x 240
Old orientation: Landscape New orientation: Landscape

Resize mode

● Stretch
○ Letterbox
○ Keep size

Controls that are out of screen border will be: Moved

Images
☑ Set all to "Stretch" mode

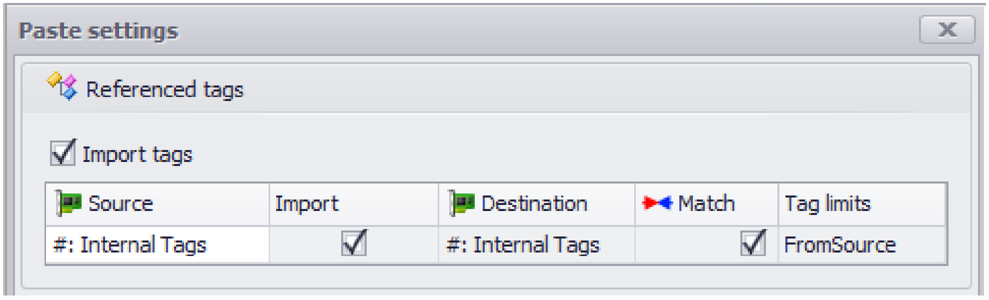
Rectangles
☑ Fix radius

Other
☐ Resize print reports

Texts
Resize mode: Minimum resize factor Minimum font size: 6
All font sizes will be multiplied by: 0.31.

Figure 63: Copying a screen between projects with different display settings

The Paste settings dialog box will appear.



Paste settings

Referenced tags

☑ Import tags

Source	Import	Destination	Match	Tag limits
#: Internal Tags	☑	#: Internal Tags	☑	FromSource

Figure 64: Paste settings dialog box (e.g., screen contents with tags)

- Select the settings you want.
- Click on OK.

The screen will be inserted into the Screens tab tree for project B.

Limitations when copying elements between projects

In order to be able to copy elements between multiple instances of the Galileo application, all the instances must belong to the exact same GALILEO program version.

If you are copying elements between projects that use different panel types, you will additionally need to confirm the new resolution and the associated changes.

→ Please note that the undo function will not be available when pasting elements between multiple instances of the Galileo application or between different projects.

Convert project

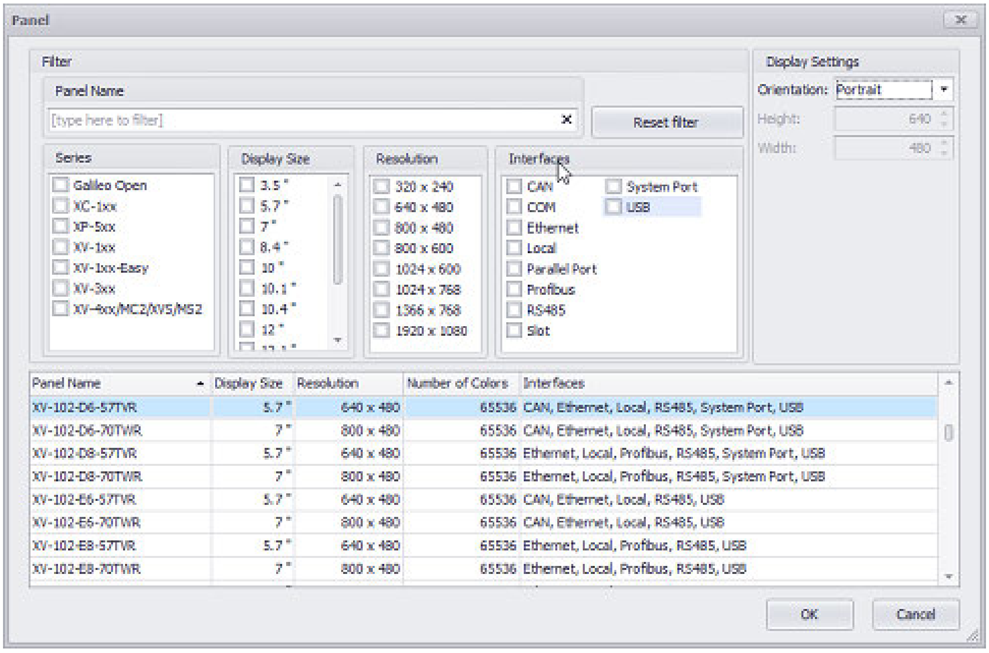
The project can be converted to a panel with a different display size, orientation, or resolution.

When you select a different panel for an open project, the same dialog box can also be used to change the orientation and resolution for the panel.

Perform the following instructions :

- Open a copy of the project that you want to convert in GALILEO.
- Open the Home tab and click on Panel Type, in the Device group.

Device tab\Panel type



Orientation: Portrait
Height: 640
Width: 480

Panel Name	Display Size	Resolution	Number of Colors	Interfaces
XV-102-D6-57TVR	5.7 "	640 x 480	65536	CAN, Ethernet, Local, RS485, System Port, USB
XV-102-D6-70TWR	7 "	800 x 480	65536	CAN, Ethernet, Local, RS485, System Port, USB
XV-102-D8-57TVR	5.7 "	640 x 480	65536	Ethernet, Local, Profibus, RS485, System Port, USB
XV-102-D8-70TWR	7 "	800 x 480	65536	Ethernet, Local, Profibus, RS485, System Port, USB
XV-102-E6-57TVR	5.7 "	640 x 480	65536	CAN, Ethernet, Local, RS485, USB
XV-102-E6-70TWR	7 "	800 x 480	65536	CAN, Ethernet, Local, RS485, USB
XV-102-E8-57TVR	5.7 "	640 x 480	65536	Ethernet, Local, Profibus, RS485, USB
XV-102-E8-70TWR	7 "	800 x 480	65536	Ethernet, Local, Profibus, RS485, USB

Figure 65: Window Select Panel

- Select the panel you want.
- Click on OK.

Changing the screen setting values will open the Resize options dialog box. This dialog box will show the resolution and/or orientation changes highlighted in color and provide additional options.

The Panel dialog box is described in the → section " Select panel", page 221 section.

Working with styles

Instead of assigning properties such as font size and font color to every single individual visualization object, you can combine these properties into styles.

Once you have created a style, you can then apply it to other visualization objects as well, saving both time and effort.

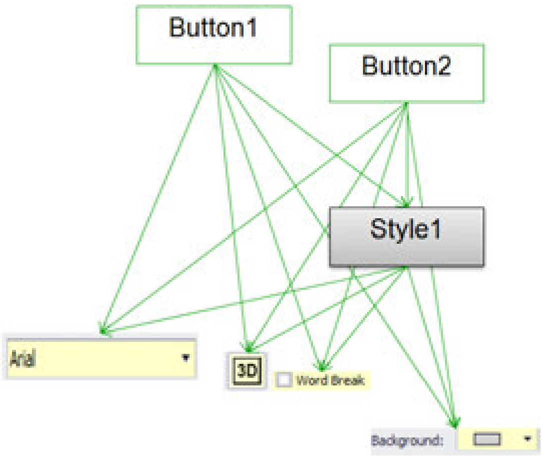


Figure 66: How styles work

For an explanation of how to create styles, please refer to the → "Style management", page 458 section.

Keyboards

GALILEO features system keyboards with styles that can be customized.

Keyboards are created the same way as screens, and are also managed in the project tree's Screens tab.
You can create your own custom keyboards or import system keyboards in order to use the latter as templates for your own custom keyboards.

The term "system keyboards" refers to the keyboards that are displayed in the application when necessary in order to make it possible to enter input or make selections. This means that their behavior is already set and that their actual visual appearance is defined with the use of styles.

For a detailed description of the properties of keyboards, please refer to → "Keyboard Properties", page 273

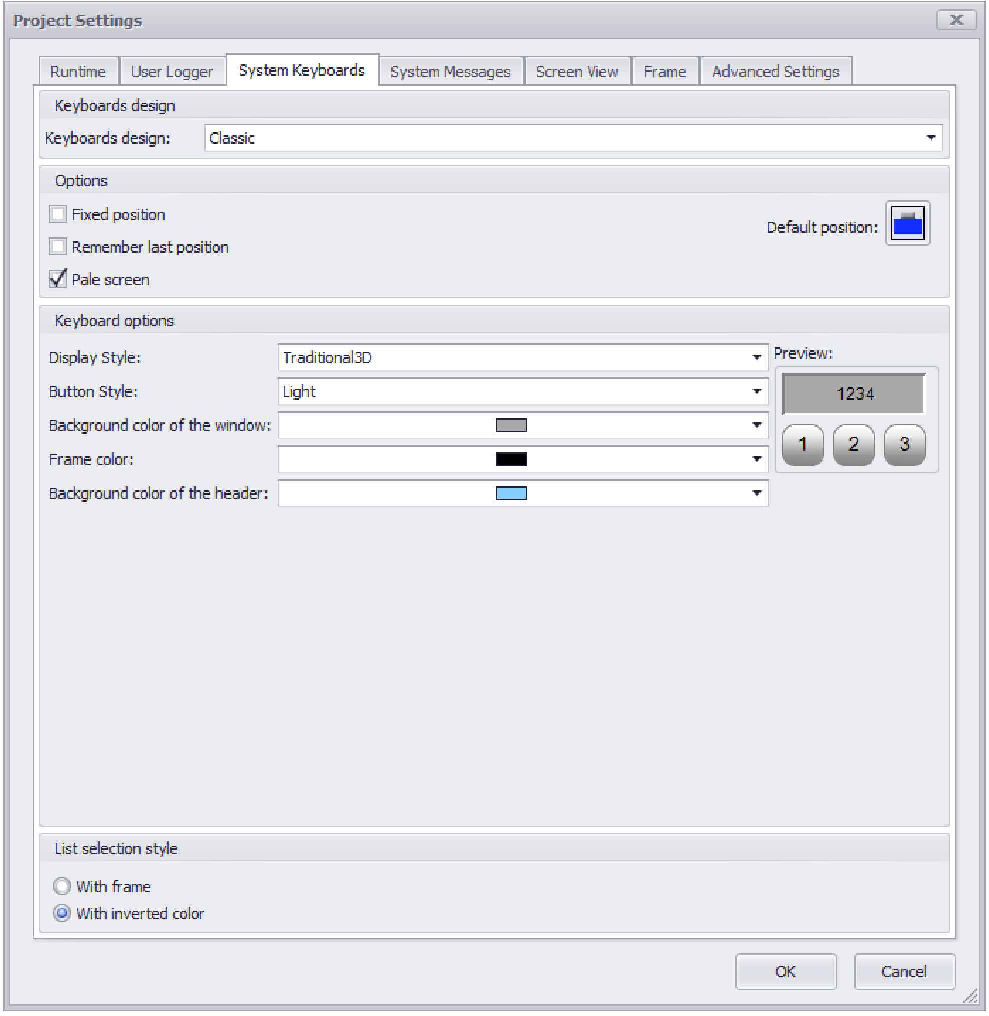
You can use keyboard styles both for the keys in a keyboard and for the way your system keyboard is displayed. These styles can be used to configure the background and font colors you want. Before you can use these styles, however, you will need to create a style for keys and a style for Value Display/Entry objects.

For a detailed explanation of how to create styles, please refer to Described in section: → section "Style management", page 458.

To apply a style you have created to your system keyboard, follow the steps below:

- ▶ Go to *Home\Project Configuration\Settings* and open the System Keyboards tab.
- ▶ Select the styles you want using the Display Style and Button Style drop-down menus.
- ▶ Click on OK.

Home\Project Configuration\Settings tab



Keyboards design

Keyboards design: Classic

Options

□ Fixed position

□ Remember last position

☑ Pale screen

Default position: 🟦

Keyboard options

Display Style: Traditional3D

Button Style: Light

Background color of the window: ▨

Frame color: ■

Background color of the header: 🟦

List selection style

○ With frame

● With inverted color

Figure 67: *Project configuration/Project Settings/System Keyboards*

The style will be automatically applied to the system keyboards.

Value Conversion Manager

Much like styles, conversion groups can be used for the units assigned to tags. In other words, a conversion group can be created once and then assigned to multiple tags, meaning that changes will only have to be made once and not for each tag individually. For an overview of the functions in the Conversion Manager, please refer to the following section: → "Conversions", page 921.

Project organization and overviews

The project tree in GALILEO contains all the screens, media (images, videos), tags, data types, and scripts in a project. It not only makes projects more manageable, but also ensures that you will be able to find the elements you want more easily.

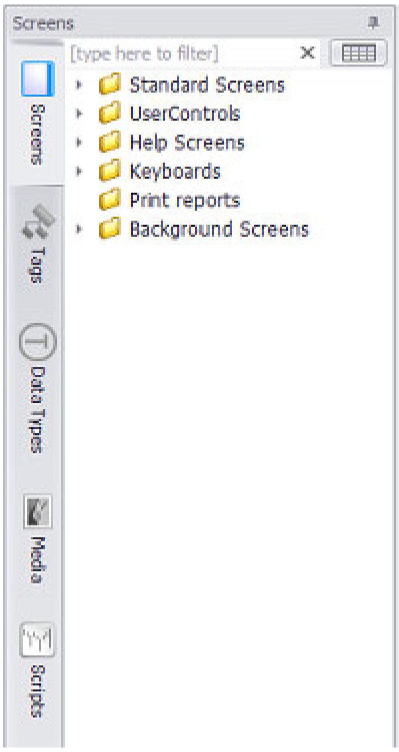


Figure 68: Project tree in GALILEO showing the project's structure

The Screens and Media trees can be structured using folders.

The project tree comes with a filter feature that can be used to quickly filter objects, even if they are in the thousands.

Use the Expand and Collapse context menu commands to expand and collapse the tree structure

Finding specific elements

1.6.1.13 in Project tree

- Enter a search term into the corresponding box.

Or

- Filter by specific types

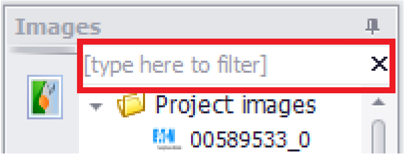


Figure 69: Filter in Project Tree

▦ Table overview

- ▶ Click on ▦ to open the table overview.

The table overview makes it easier to work in the project.

The table overview's content will depend on the element making the call.
For screens and tags, the table can be used to compare and edit most properties

- ▶ Click on ▽ to filter by specific column contents
- ▶ Disable the filter by selecting the (All) option from the drop-down filter menu

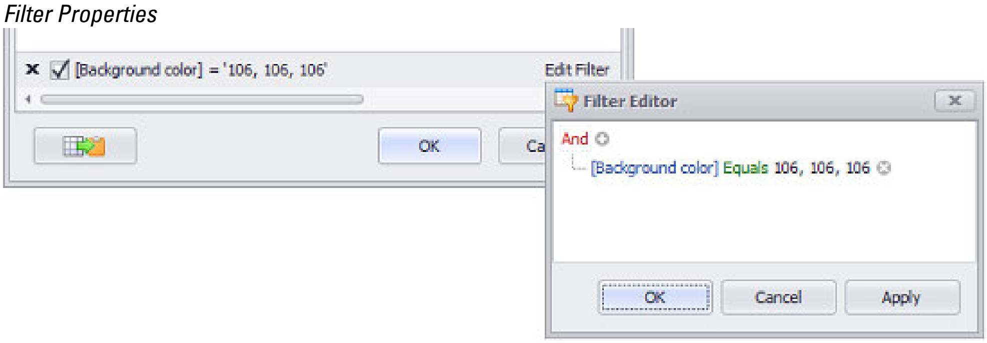
The Recipe Editor lists the recipe tags with the place where they are used. You can adjust the view by clicking on ☰ to expand all and ☰ to collapse all. If all the tags in a structure are used, the checkmark will be black. If, on other other hand, only some of the tags in the structure are used, the checkmark will be gray.

Filter Editor

As soon as a filter is set in the table, you will be able to use the editor to customize the filter with your own custom filter logic.

Set filters can be canceled in the status bar by clicking on ☒.

Filter Properties



× ☑ [Background color] = '106, 106, 106'

And ⊕
[Background color] Equals 106, 106, 106 ⊗

Figure 70: Example of the Filter Editor

Filter	
And	Boolean logic conjunction
[Background color]	Filters by a specific column
is empty	Filters by field content

- ▶ In the editor line, click on one of the colored elements and select or enter the filter value you want.

See also

→ section "Working with the screen table", page 276
→ section "Working with the tags table", page 560

Controls Manager

The Controls Manager is a tool designed to provide a quick overview showing which visualization objects are being used on a screen. These objects will be accordingly sorted vertically. The Controls Manager can be used to select and hide visualization objects, as well as to lock them in place.

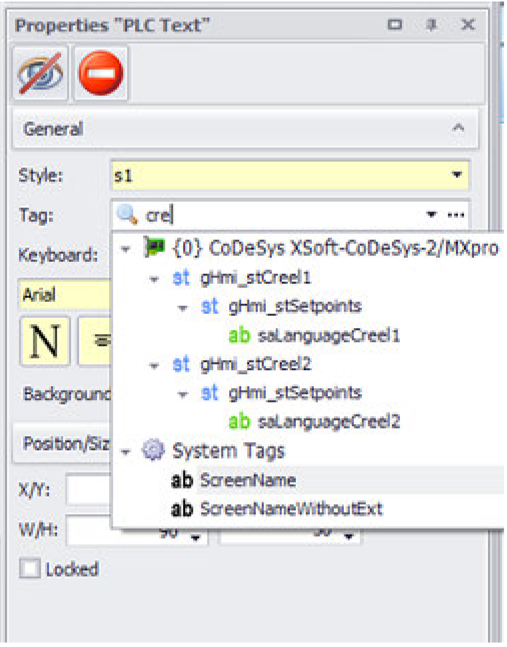
The Controls Manager is described in detail in a different section – please refer to → "Working with the Controls Manager", page 523.

Real-time full-text search

GALILEO features an autocomplete function designed to help you when entering a text or tag into a field by providing existing texts and tags for selection. The following example is intended to illustrate how this function works:

- Create a tag of data type STRING called ScreenName.
- Create a Value Display/Entry visualization object.
- In the Properties 'PLC Text' pane, enter the first few letters of the tag you created into the Tag field, e.g., ScreenName.
- Click on the tag you want after it is shown on the drop-down menu.

Properties pane for the object



Properties "PLC Text"

General

Style: s1

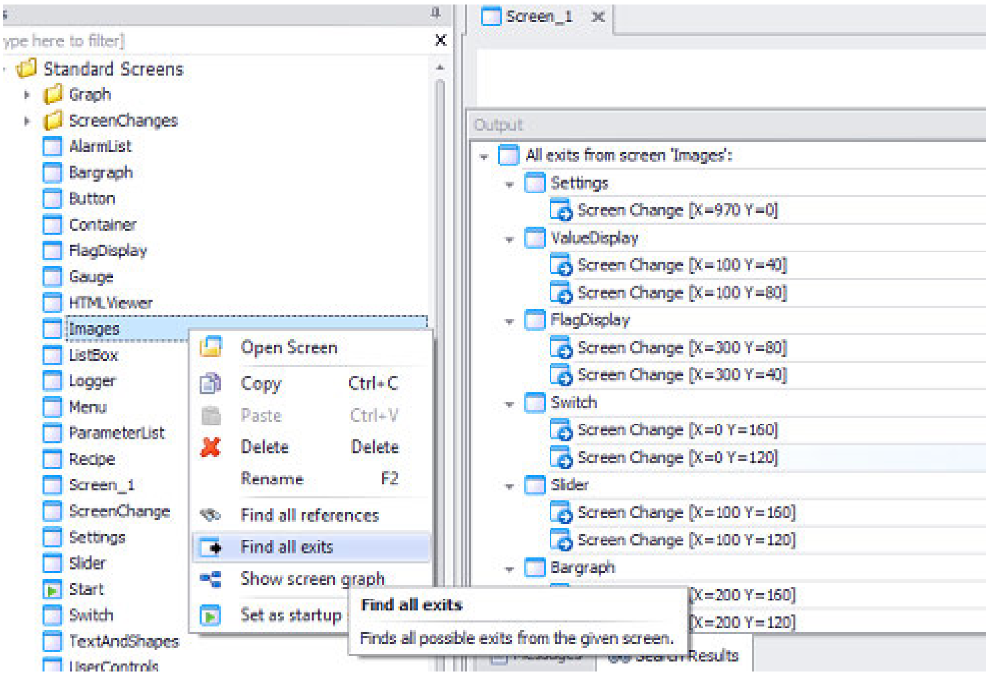
Tag: cre

Figure 71: Full-text tag name search

Find all references

The Find all exits function makes it easier to find links within a GALILEO project. For instance, to find where a specific screen is being used, you would follow the steps below:

- ▶ Go to Project Tree\Screens and right-click on the screen you want in order to open the context menu.
- ▶ Click on Find all exits.



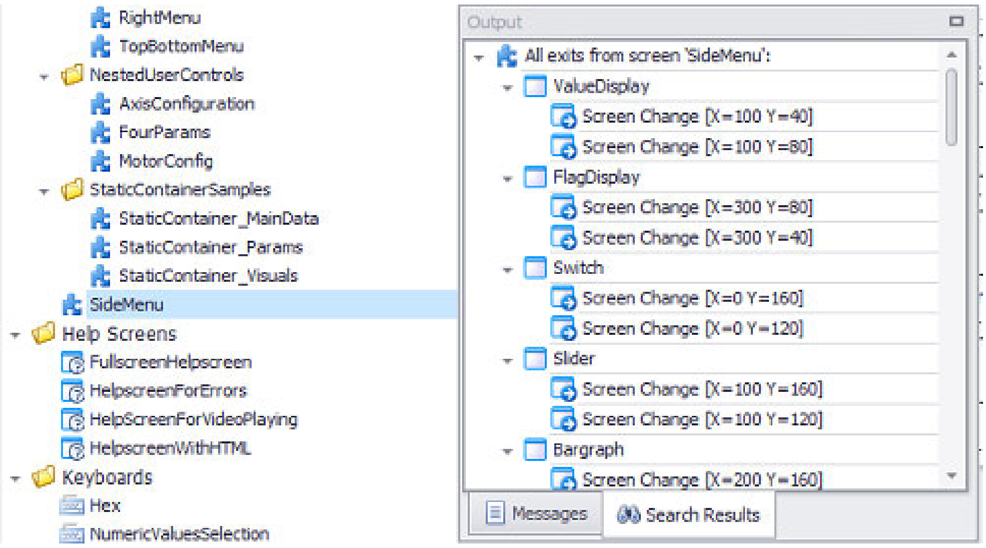
```
All exits from screen 'Images':
  Settings
    Screen Change [X=970 Y=0]
  ValueDisplay
    Screen Change [X=100 Y=40]
    Screen Change [X=100 Y=80]
  FlagDisplay
    Screen Change [X=300 Y=80]
    Screen Change [X=300 Y=40]
  Switch
    Screen Change [X=0 Y=160]
    Screen Change [X=0 Y=120]
  Slider
    Screen Change [X=100 Y=160]
    Screen Change [X=100 Y=120]
  Bargraph
                  [X=200 Y=160]
                  [X=200 Y=120]
```

Find all exits
Finds all possible exits from the given screen.

The Output pane will show all the places where the tag is being used.

Editing a screen exit

- ▶ Double-click on the search result you want.
  The corresponding object used to trigger the screen change will be opened.



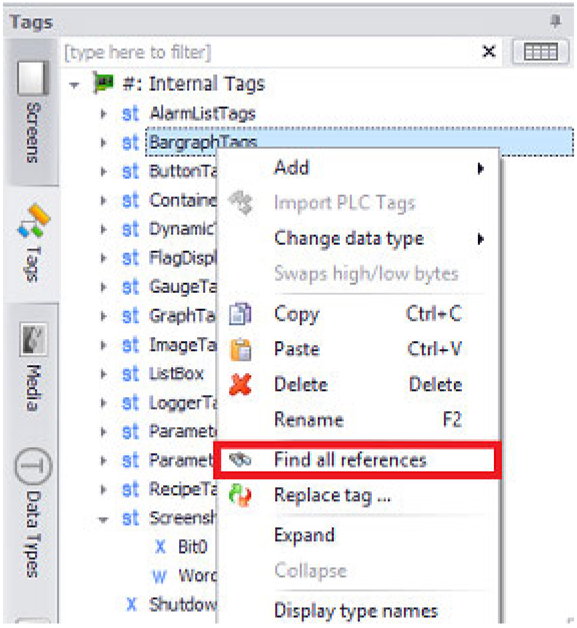
```
All exits from screen 'SideMenu':
  ValueDisplay
    Screen Change [X=100 Y=40]
    Screen Change [X=100 Y=80]
  FlagDisplay
    Screen Change [X=300 Y=80]
    Screen Change [X=300 Y=40]
  Switch
    Screen Change [X=0 Y=160]
    Screen Change [X=0 Y=120]
  Slider
    Screen Change [X=100 Y=160]
    Screen Change [X=100 Y=120]
  Bargraph
    Screen Change [X=200 Y=160]
```

→ You can change the target screen in the Properties pane for the screen change you selected.

Find all exits

The Find all references function makes it easier to find every instance of a specific element being used within a GALILEO project.
For instance, to find where a specific tag is being used, you would follow the steps below:

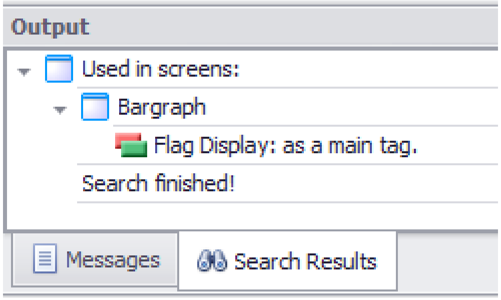
- Go to Project Tree\Tags and right-click on the tag you want in order to open the context menu.
- Click on Find all references.



The Output pane will show all the places where the tag is being used.

Editing the element using the tag

▶ Double-click on the search result.
The element using the tag will be opened.



Output

- Used in screens:
  - Bargraph
    - Flag Display: as a main tag.
  - Search finished!

The → section "Application Configuration", page 135 can be used to enable a search for object references that will be run before an object is permanently deleted.

Window, Assistants (Wizards) and dialog boxes

Most dialog boxes need to be closed or exited by clicking on the OK, Cancel, Next>, or Close button as applicable.

However, there are also certain dialog boxes (e.g., Compiler dialog box) that will remain open like a normal window while you continue using GALILEO.
This is in order to ensure that dialog boxes that provide information needed to use the program will remain open.

Undo and Redo

GALILEO will keep up to 20 actions in its undo stack, provided these actions are carried out during the current session.
If you are not happy with the results of an action or make a mistake by accident, you can use the "Undo" command in the Clipboard group in order to undo the action, as well as any additional actions before it, one by one.
Please note that the undo stack will be cleared when you save your project!

Undo

Can be used to undo your most recent actions, such as moving, enlarging, copying, and deleting objects.
Please note, however, that changes to objects' properties cannot be undone.

This command can be used from the Home tab, the→ "Quick Access Toolbar", page 61, or the context menu.

Redo

Can be used to redo undone actions such as moving, enlarging, copying, and deleting objects. Please note, however, that changes to objects' properties cannot be redone.

This command can be used from the Home tab, the→ "Quick Access Toolbar", page 61, or the context menu.

Tooltips

Tooltips are object descriptions for all objects in GALILEO, e.g. Buttons or System tags These tooltips will appear when you move the cursor over the corresponding objects.

When you move the cursor over screens or image files in the project tree's Screens and Images tab panes, a preview similar to a tooltip will be shown.

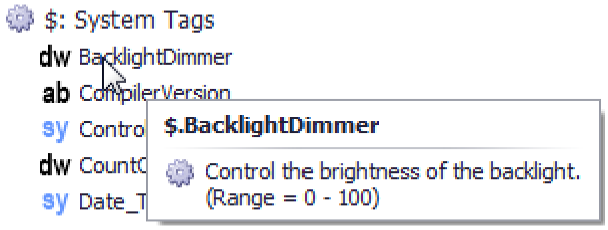


Figure 72: Tooltip for tags

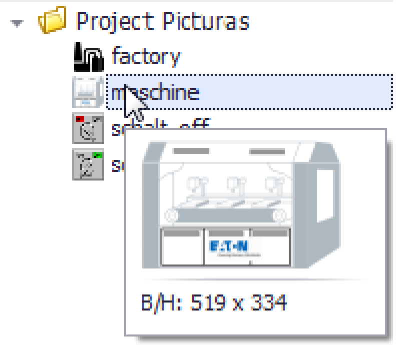


Figure 73: Image preview

Calling dynamic link library (DLL) functions

You can call functions from your own DLLs in GALILEO. The number of parameters you can use is unlimited.

DLL parameters can be "in" or "ref" parameters, i.e., parameters that are passed to the DLL but are not returned or parameters that are both passed and returned.

1.6.1.14 Customizing the application's user interface

GALILEO features a series of functionalities designed to ensure that you will be able to work
with GALILEO comfortably.

The section that goes over → "Running GALILEO for the first time", page 54 explains how to change the language for the application.

The → "Functions on main tabs", page 76 are listed in table format in the section that explains how to use the program.

Customizing your personal workspace

Setting the position of the various panes and screens on the user interface

To undock a pane or screen and move it to the right/left/top/bottom position you want on the screen, simply drag and drop it accordingly.
When holding down the left mouse button after clicking on the corresponding menu's or pane's title bar, a docking zone selector will appear in order to make positioning easier.
The shaded color area that appears during this process is intended to show the currently selected position.

Positioning the Controls Manager pane at the top of the workspace

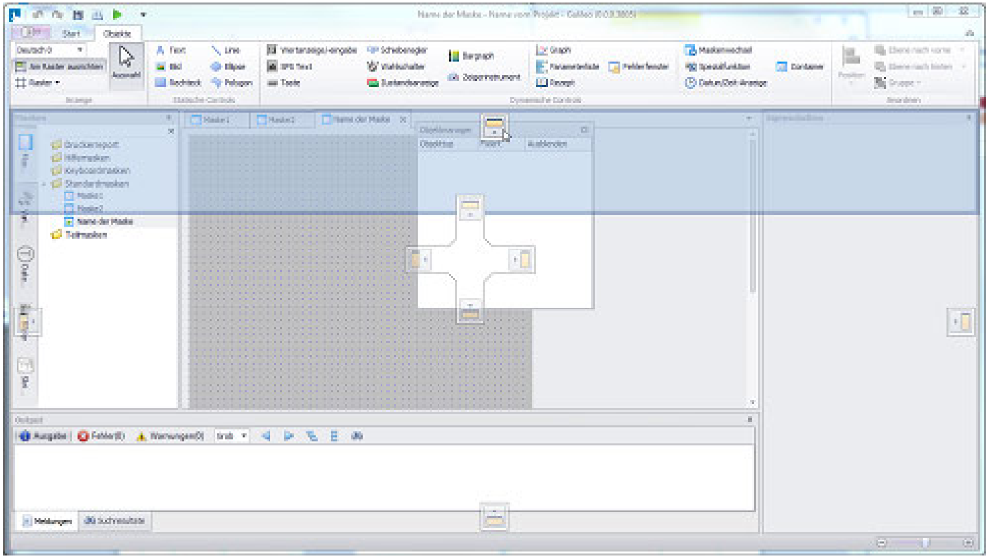


Figure 74: Docking zone selector for positioning the Controls Manager pane

You can move screens within the workspace and/or to a screen area outside the GALILEO application.

Positioning a screen on the workspace

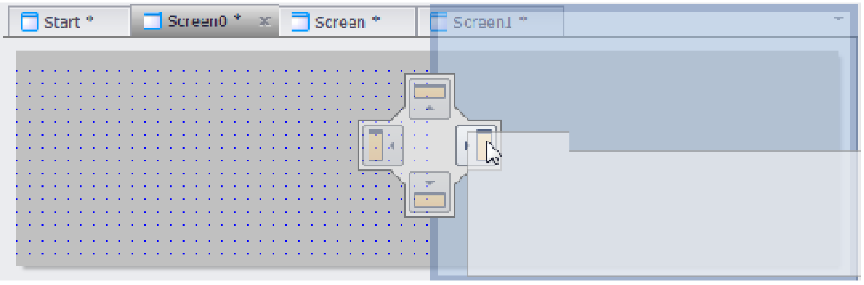


Figure 75: Docking zone selector for positioning screens on the workspace

Placing standard screen <M0> at the bottom by itself

- Click on the tab next to the "M0" screen name and hold the mouse button down (1)

The docking zone selector will appear

- While keeping the left mouse button held down, drag tab "M0" to the bottom position on the docking zone selector (2)
- Release the mouse button (3)

The docking zone selector will disappear and the "M0" screen will be in the position you want.

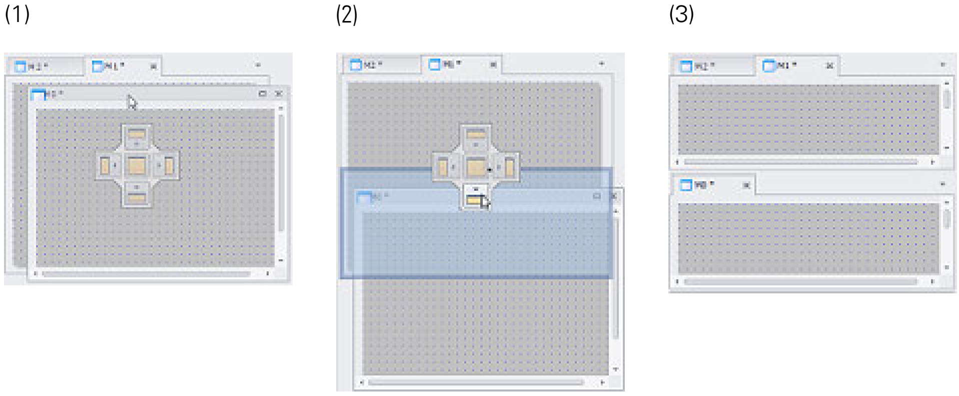


Figure 76: Docking zone selector for positioning screens. In this example, the selector is being used to dock standard screen M0 at the bottom of the start screen by itself

Docking / using the auto hide function for the project tree and for other user interface panes

The project tree pane, as well as the other panes, can be separately shown and hidden.

- Clicking on 📌 will automatically hide the project tree or other corresponding pane at the edge of the user interface and show a corresponding tab.
- Clicking on ⊣ in the active project tree pane or in another pane will dock the pane so that it is not hidden automatically anymore.

Maximizing and restoring the user interface panes

Panes can be maximized within their user interface position or restored to their original size as necessary.

- Clicking on □ will maximize the pane within its user interface area
- Clicking on ⧉ will restore the pane to its previous size

Displaying, editing, and saving screens on the workspace

There are several options available for editing screens after they have been created in the project tree.

Opening a screen

- Double-click on the name of the screen you want after opening the appropriate screen type folder in the *Project Tree/Screens tab pane*

Showing a screen

- Click on the tab showing the open screen's name

Or

- Click on ▾ to open a list of all open screens and select the screen you want.

The screen you selected will be moved to the foreground.

Once a screen is in the foreground, you can edit it using the various available objects.

Once you make changes to a screen, an asterisk (*) will be shown after the screen name until you save the project. If you close the screen while still editing the project, the changes you have made will be kept temporarily until you either save the project or close it without saving.

→ Screens are not saved as standalone files independently of the project, i.e., they are only saved when the project is saved!

Saving a project

- Click on the 💾 icon in the toolbar.

Context menu for<screen name> tab

The context menu that can be accessed by right-clicking on the <screen name> tab can be used to customize the view.

In order for all of the context menu's functions to be available, multiple screens need to be open.

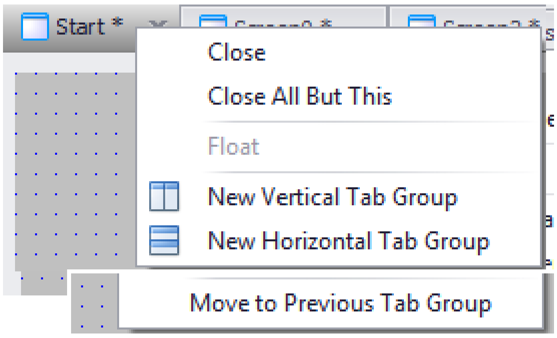


Figure 77: Context menu for <screen name>tab

Close	Closes the active screen in the foreground
Close All But This	Closes all screens except for the current one
Float	Opens the screen in its own pane

Work pane splitting

If you select one of the two options, the other one will cease to be available, i.e., you can use vertical tab groups or horizontal tab groups, but not both

New vertical tab group	Work pane splitting The active screen will be placed in a new view next to the other open screens
New horizontal tab group	Work pane splitting The active screen will be placed in a new view underneath the other open screens
Move to next tab group	If the work pane is split, the selected screen will be moved to the next horizontal or vertical section

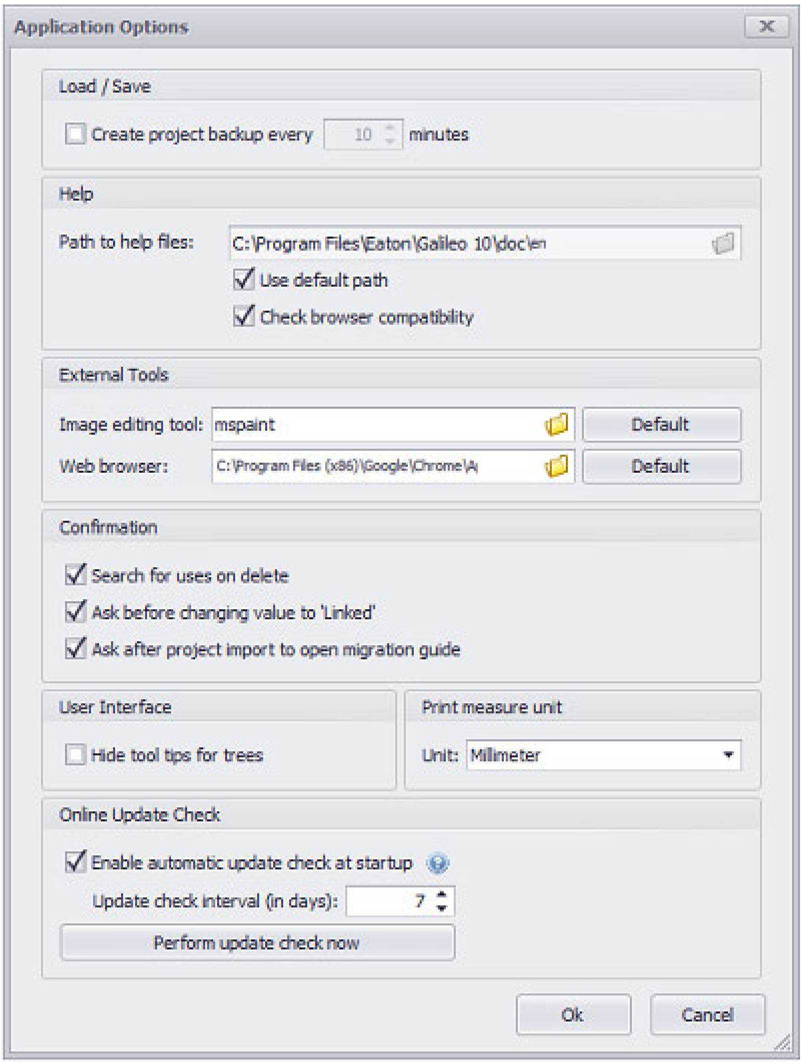
Application Configuration

Window Application Options

This dialog box contains options that apply to GALILEO itself as a program.

In other words, the settings in this dialog box will apply to all projects on the computer on which the Automatization Software GALILEO program has been installed.

Home\Application Configuration\Options



Application Options

Load / Save

☐ Create project backup every 10 minutes

Help

Path to help files: C:\Program Files\Eaton\Galileo 10\doc\en

☑ Use default path

☑ Check browser compatibility

External Tools

Image editing tool: mspaint

Web browser: C:\Program Files (x86)\Google\Chrome\A

Confirmation

☑ Search for uses on delete

☑ Ask before changing value to 'Linked'

☑ Ask after project import to open migration guide

User Interface

☐ Hide tool tips for trees

Print measure unit

Unit: Millimeter

Online Update Check

☑ Enable automatic update check at startup

Update check interval (in days): 7

Figure 78: Application Options dialog box

Load / Save

☑Create project backup every

When this setting is enabled, the project will automatically
be saved together with all screens every x minutes, where x is the configured interval.

Help

Path to help files

This field will show the folder where the GALILEO User Help has been stored.
You can set the online help path here if the online help components were installed only once when installing GALILEO or if the same online help needs to be available for multiple GALILEO versions.

☑Use default path

Uses the path set during installation

☑Check browser compatibility

During installation, checks whether the browser installed on the computer supports the GALILEO User Help's full range of features. If applicable, a system message will be shown in order to provide the user with the result.

External tools

Image editing tool

Used to specify the graphics editor on the computer that will be opened to edit images outside GALILEO when the corresponding option is selected in *Project Tree\Media*.

If you click on Default, mspaint – the Microsoft Paint image editing program that comes with Microsoft Windows – will be used.

Web Browser

You can use this field to select your preferred browser for web visualization purposes. You can even select a browser other than the operating system's default browser.

Default will clear the browser you selected and the default browser used by the operating system will be used instead.

Confirmation

☑ Search for uses on delete

Before individual objects are deleted, the program will look for references to them in the open project in order to prevent references from being accidentally deleted.

☑Ask before changing value to 'Linked'

Only relevant for projects with various visualization groups (web).
A prompt will appear before a specific property value for a visualization group is applied to all groups.

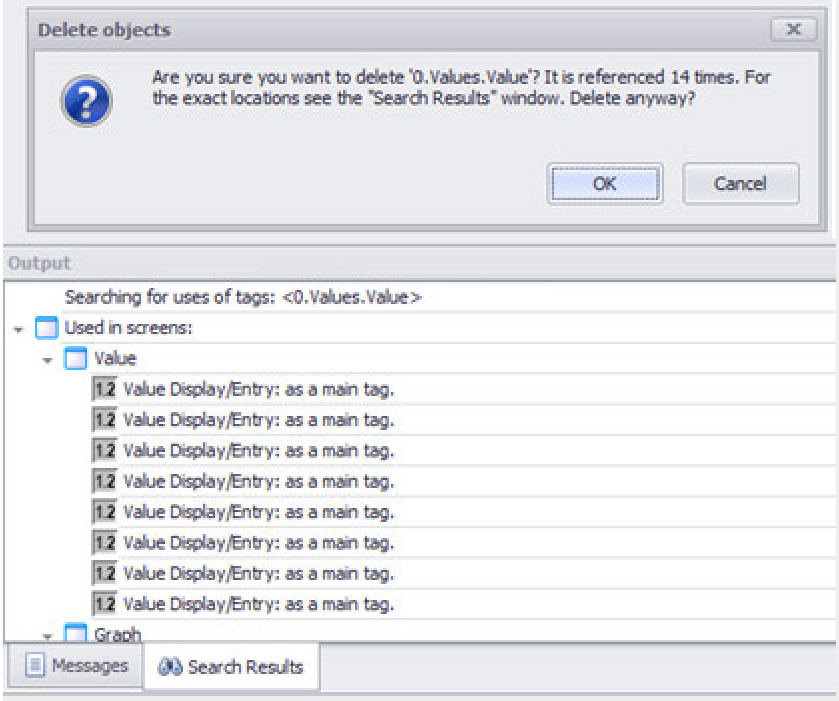
☑Ask after project import to open migration guide

After importing projects created with GALILEO 8.1, modifications and setting changes for the

functions in GALILEO may be required.
The Migration Guide document contains instructions that will assist you with the process. The Migration Guide can be accessed at any time in the tab by clicking on Help and Tools. The document is only available in English.

Object deletion example

Assume that the search is enabled. In this case, when you attempt to delete tag "X", the program will search the currently open project for any tag "X" references (everywhere, e.g., in screens, recipes, scripts, etc.).
Before the tag is permanently deleted, the program will show a warning indicating that tag "X" is referenced n times, and the Output pane will list the exact locations where it is referenced.



Delete objects

Are you sure you want to delete '0.Values.Value'? It is referenced 14 times. For the exact locations see the "Search Results" window. Delete anyway?

Searching for uses of tags: <0.Values.Value>
Used in screens:
Value
1.2 Value Display/Entry: as a main tag.
1.2 Value Display/Entry: as a main tag.
1.2 Value Display/Entry: as a main tag.
1.2 Value Display/Entry: as a main tag.
1.2 Value Display/Entry: as a main tag.
1.2 Value Display/Entry: as a main tag.
1.2 Value Display/Entry: as a main tag.
1.2 Value Display/Entry: as a main tag.
Graph

Figure 79: Object deletion example

User Interface

Hide tool tips for trees	☑ If this checkbox is enabled, the preview function when moving the cursor over individual objects in the project tree will be disabled. □ Preview function enabled

Print measure unit

Unit:	Used to select a length unit

	for when printing the project on a plotter/printer The available options are Millimeter and Inch
Online update check	
☑ Enable automatic update check at startup	If this option is ☑ enabled, GALILEO will use an existing Internet connection to check for updates at the specified interval whenever it is started.
Update check interval (in days):	Used to set the interval at which GALILEO will automatically search for newer program versions when started.

1.6.2 Example projects

When originally installing the program, you specified

- Whether the sample projects would be stored on your computer
- And the folder where they would be stored

(Please refer to the → section "Now select the component you want out of the ones that are available.", page 47)

The default folder is C:\Users\Public\Documents\Eaton\Galileo\Samples.

Both demo projects are executable, meaning they can be transferred to your touch panel.

Installation directory\XV300Demo\DemoApplication2.prj

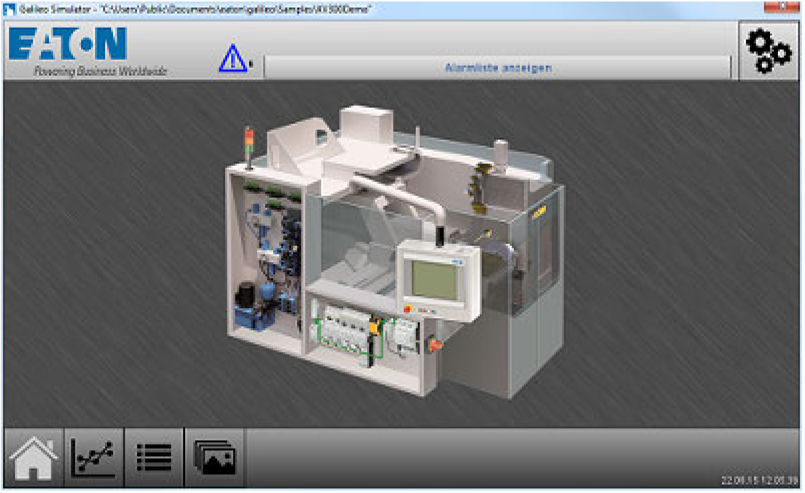

Galileo Simulator - "C:\Users\Public\Documents\eaton\galileo\Samples\XV300Demo"
EATON
Powering Business Worldwide
Alarmliste anzeigen

Figure 80: Demo project for use on Eaton XV300 touch panels

Some of the features implemented in this demo project, named DemoApplication2.prj, are listed below:

- Graph functions
- Parameter customization
- Gesture-based control with GALILEO

Installation directory\XV100Demo\demo.prj

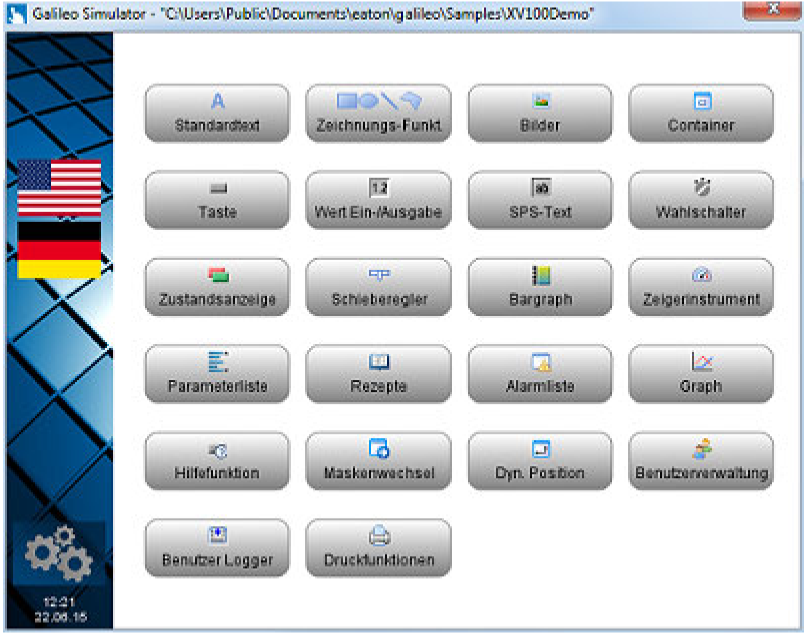


Figure 81: Demo project for use on Eaton XV100 touch panels

This demo project, named demo.prj, provides an overview of the various static and dynamic control options and their implementation in the various types of screens.

1.6.3 Getting started with tutorials

In addition to this GALILEO User Help user help and the

various application examples saved on your computer; please refer to → section "Example projects", page 139

you can also use the various individual tutorials in order to learn more about the functionalities in GALILEO.

These tutorials are currently only available in English.

To access a tutorial directly in GALILEO, go to the tab and click on Help and Tools.

Or

EatonVideos on YouTube
General Query Link
[Youtube.com/user/EatonVideos/search?query=galileo+10]

Table 2: Available tutorials

	Name of tutorial	Contents
🌐	First Steps How to Tutorial	Takes you through the steps required in order to create your very first simple HMI project
🌐	User Interface Walkthrough	Provides a quick overview ofGALILEO's user interface.
🌐	Screen Designer Tools and Hints	A number of tools and hints that will enable you to work more productively with the screen designer.
🌐	General Productivity Tools and Hints	Shows a number of general tools and hints that you can use to increase productivity.
🌐	Styles Tutorial	How to work with styles: Creating new styles, assigning styles, editing existing styles.
🌐	Tag Import	Explains the integrated tag import functionality, including repeat imports with modified data.
🌐	Background Screens	Goes over how to use background screens in order to create a unique appearance for your project.
🌐	Controls Manager	A useful tool that can simplify a number of screen designer tasks and that you can use, for example, to temporarily hide the controls you want.
🌐	Control Sets	Organize and categorize your controls on the screen.
🌐	Parameterized Subscreen	Explains how to use parameters in order to reuse a subscreen as a user control when displaying various data.
🌐	Parameterized Subscreens (Advanced)	Shows how to use custom data types as parameters in user controls
🌐	User Defined Keyboards	Goes over how to import system keyboards, edit system keyboards, and use custom system keyboards in your project
🌐	Graphs	Covers how to display values in a graph and the required settings
🌐	Multi-Language	Shows how to create a project with multiple languages, including relevant tips and functions
🌐	GALILEO: New Features	An introduction to the new features and functions in Galileo 10.2 and higher

1.6.4 Your own first project

Follow the steps below to become familiar with GALILEO and how GALILEO is designed to conveniently assist you throughout your project.

→ "Getting started with ", page 144 This example will get you started by having you create your own application and test it on your computer.
This project explains how to use the Automatization Software GALILEO program wherever necessary.

1.6.4.1 Overview of your own first project

Following is a list of the steps involved in this first project:

→ "Preparation", page 143 before getting started with your first project

1. Creating a new project

The New Project... Wizard will guide you through the process.

When you create a new project, the standard "Start" screen will be created automatically.

2. Configuring project settings

Configuring the grid and the objects' alignment

3. Configuring the start screen's properties

Setting a background color.

4. Creating an additional standard screen

Setting a standard screen as the start screen.

5. Adding an image

Adding an image to the project tree and placing it on the screen in the workspace.

6. Creating a user control

Selecting the user control that will be inserted

7. Adding a button with a label

Adding a button object and creating the text for the corresponding label

8. Adding screen changes

"Connecting" the various screens to each other.

9. Testing the project

Simulating the project on the computer on which GALILEO is installed

1.6.4.2 Preparation

Before getting started with your first project

GALILEO works best with a structured approach. Because of this, all the objects used in GALILEO should be organized in the project tree before or while working on the project and then used on the various screens.

Because of this, you should consider the following questions before creating a project:

- How should the project be named? Where should it be stored?
  - Name and location.-
- Do you want the project to have multiple languages?
  - Add project languages -
- Which types of screens will be needed? How should they be named?
  - Standard screens, help screens, user controls, ...-
- How will the screens be connected to each other?
  - Which screens are accessed from the start screen? -
- What should be displayed on the screens?
  - Buttons, images, values, indicators, ...-
- Which values should be shown? How should they be named?
  - Button labels, indicators, variables-

See also

→ "Notes on creating a Project", page 192

→ GALILEO is only able to import images (graphs, icons, logos, etc.), i.e., these images cannot be edited in the program.
If you want to make modifications to images, you will need to do so externally with an appropriate graphics editor that can be used to directly open the images by using the context menu in *Project\Media tree*.
The graphics editor that will be used by default can be selected in the → section "Window Application Options", page 135.

▶ Start by outlining the structures that you want to use to organize your project.

▶ Start GALILEO and set a user language.
→ section "Language", page 82

▶ Familiarize yourself with the user interface.
→ section "The user interface", page 58's user interface

Continue to the following step with your first project:

→ "Step 1: Creating a new project", page 158

1.6.4.3 Getting started with

You can use the myFirst exercise to learn how to familiarize yourself with GALILEO. This exercise will have you create a project on your computer step by step.

→ The following description does not come with additional explanations.
For more detailed information, please refer to
→ section "Using the Automatization Software program", page 58
or
→ section "Your own first project", page 141

Machine controller exercise

The application created in this exercise is intended to allow touch panel operators to switch from the Start screen to another screen and back.
To enable them to do this, both screens need to have a button with a label that can be used to switch between them. The button will have an icon on the Start screen. Meanwhile, the second screen will represent a machine controller. This target screen will have a schematic diagram of the machine in the background and a button that can be used to go back to the Start screen. In addition, this screen will feature a button that can be used to change the value for a bit tag. During actual operation, the touch panel operator would use this button to switch the machine on and off.

This example is based on a 10" XV300 touch panel using the CODESYS V2 communication method.

The steps below will guide you as you implement this exercise as an application in GALILEO:

1. → section "Creating the <MyFirst> project", page 147
2. → section "Importing all Media that will be used for the project ", page 147
3. → section "Creating the <MachineController> screen", page 148
4. → section "Adding a screen change (to the Start screen) for transitioning to the target screen", page 149
5. → section "Adding a screen change (to the target screen) for transitioning back to the Start screen", page 151
6. → section "Adding an image as a background to the target screen", page 152
7. → section "Adding a button with a label to the target screen", page 153
8. → section "Creating a bit tag and linking it to the button", page 154
9. → section "Linking the bit tag to images representing the switch position", page 156
10. → section "Testing the project in the Galileo Simulator", page 157

Preview of end result in the Galileo Simulator

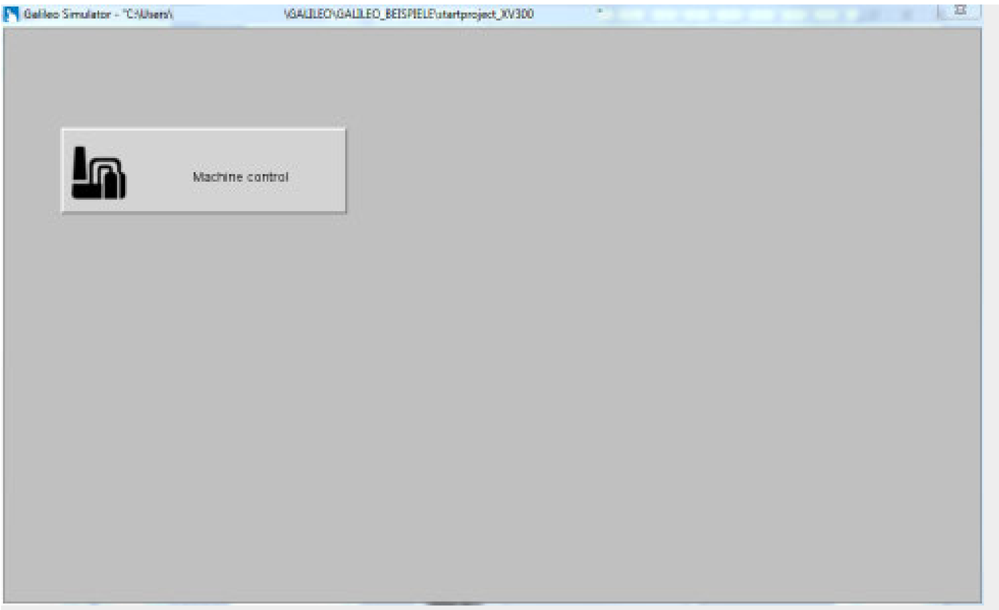

Galileo Simulator - "C:\Users\ \GALILEO\GALILEO_BEISPIELE\startproject_XV300
Machine control

Figure 82: touch panel view for the Start screen

Clicking on the icon will make the application switch to the MachineController screen:

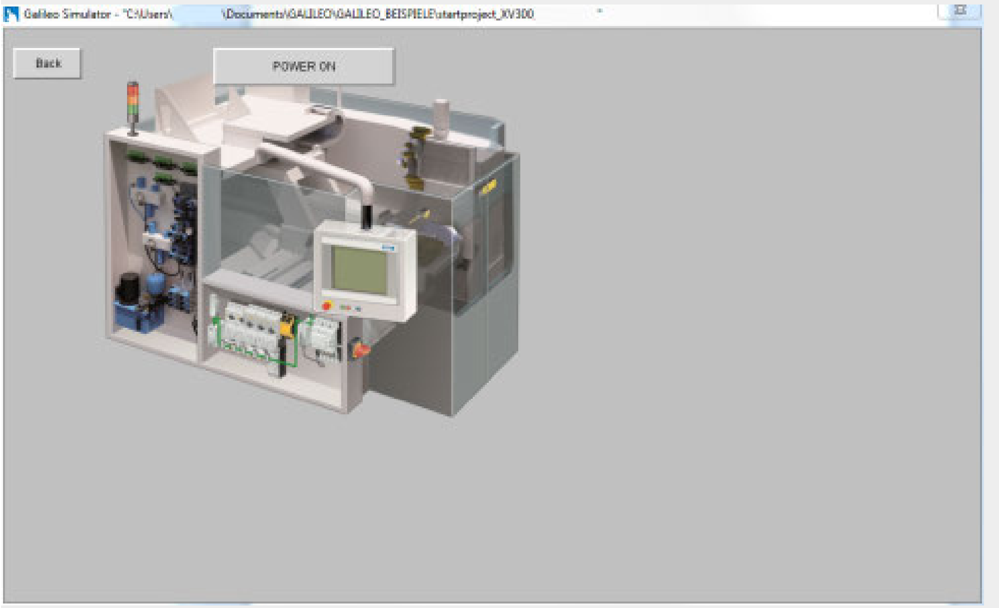

Galileo Simulator - "C:\Users\ \Documents\GALILEO\GALILEO_BEISPIELE\startproject_XV300
Back
POWER ON

Figure 83: touch panel view for the MachineController screen in the OFF state

Clicking on Back will make the application switch back to the Start screen.

Clicking on the SWITCH ON button will set the start_stop bit to 1 and the switch position will change. This effect is achieved by hiding and showing the schalt_off and schalt_on images.

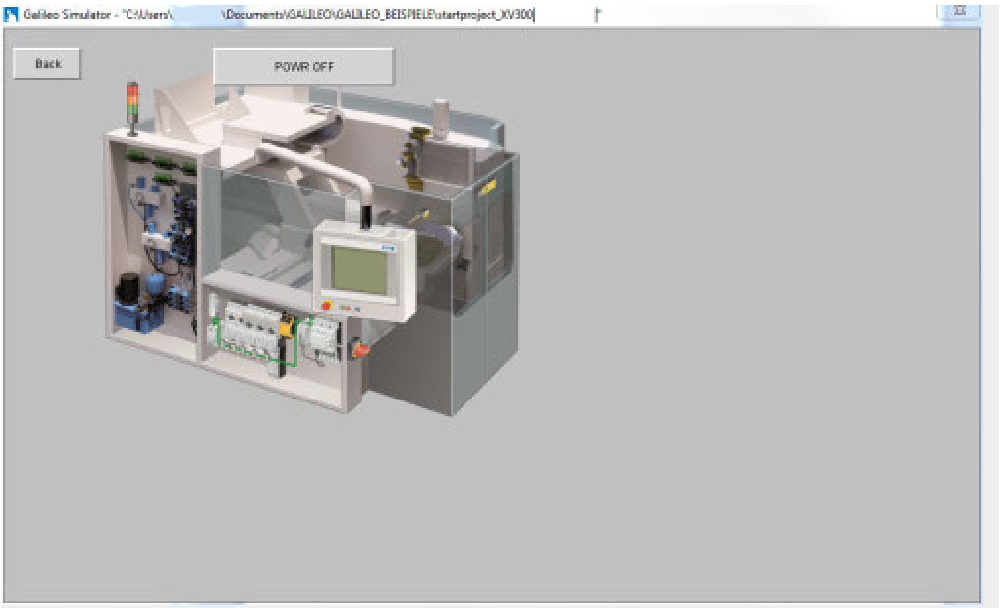

Galileo Simulator - "C:\Users\ \Documents\GALILEO\GALILEO_BEISPIELE\startproject_XV300
Back
POWR OFF

Figure 84: touch panel view for the MachineController screen in the ON state

Creating the <MyFirst> project

- Click on the project management tab on the start screen and then click on New Project...

The corresponding wizard will open in a new dialog box on your screen.

- Enter <MyFirst> into the Project name field
- Enter the storage location for the project.
  If you want, you can enter the corresponding path directly, e.g., *C:\Users\...\StorageLocation*
- Select the ☑ <XV-3xx> series and a ☑ <10''> display size
- Select a by double-clicking on it in the Panel Nametouch panel table.
- Select the <CODESYS V2> communication method with <Ethernet> as the interface by double-clicking on it.
- Click on Next twice.
- Click on Finish.

You have now created the <MyFirst> project. The Start standard screen will be automatically created and displayed on the workspace.

Exercise step 1

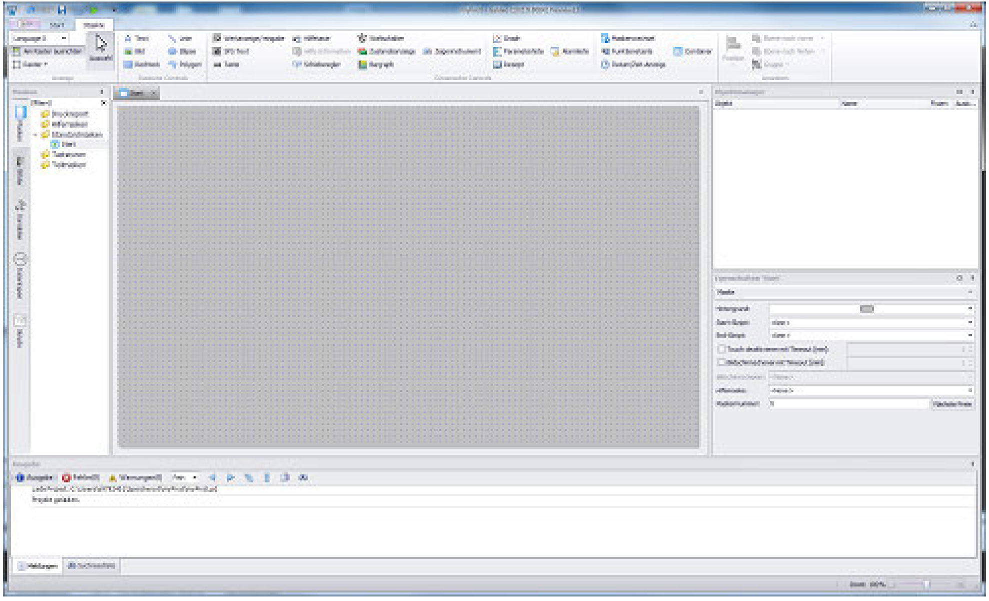


Figure 85: New project after being created

Importing all Media that will be used for the project

To do this, follow the steps below:

- Go to the project tree.
- Open the Media tab.

The Project images folder will have already been created.

- ▶ Right-click on the Project images folder to open the corresponding context menu.
- ▶ Click on Import images.
- ▶ Select the required media files and click on OK

You can use the images from the DemoApplication2.prj demo project. These images will be found in the following installation folder: DemoApplication2.prj\images.

Import the following images, for example:

- factory.png for the icon on the start screen
- schalt_on.png for the image for the "ON" switch position
- schalt_off.png for the image for the "OFF" switch position
- maschine klein.png for the machine image on the target screen

Once you import them, the images will be found in the Project images folder.

Exercise step 2

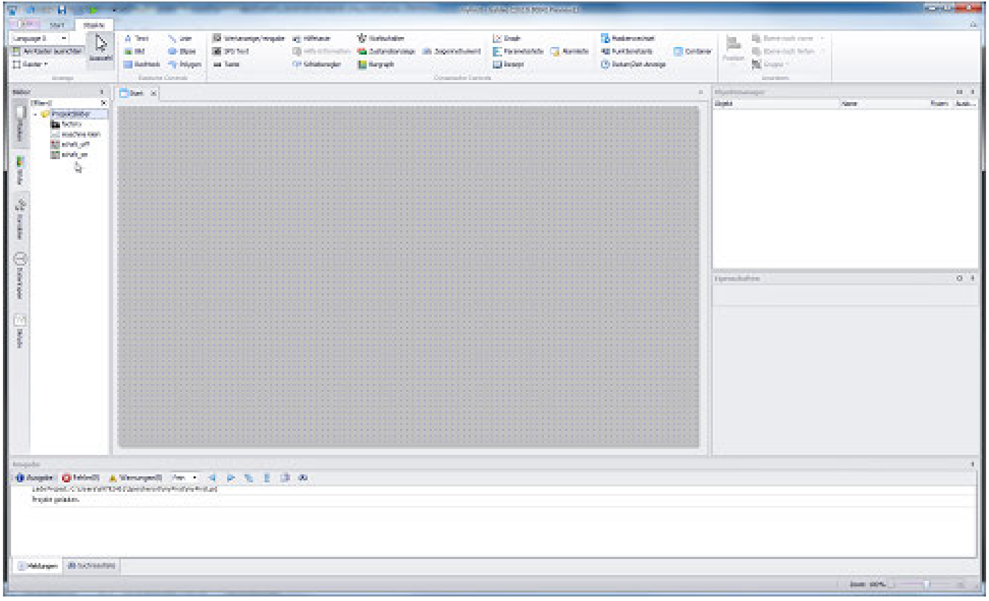


Figure 86: Images after being imported

Creating the <MachineController> screen

Create an additional standard screen called <MachineController>.

- ▶ Go to the project tree.
- ▶ Click on the Screens tab.

- ▶ Open the context menu by right-clicking on the Standard screens folder
- ▶ Click on Add new screen.
- ▶ Change the name from Screen0 to <MachineController>.

The MachineController standard screen has now been created.

Exercise step 3

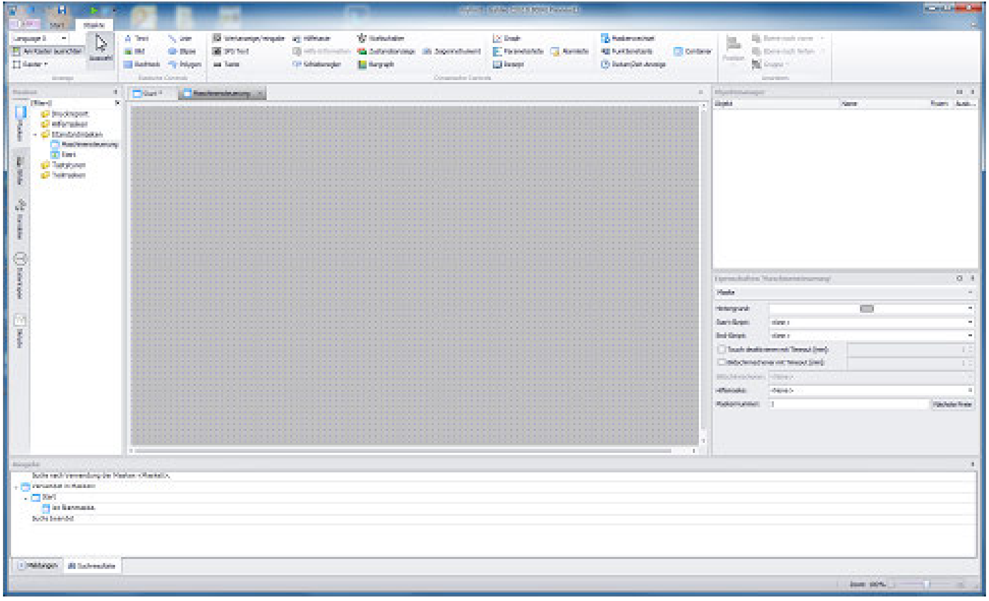


Figure 87: The MachineController screen after being created

Adding a screen change (to the Start screen) for transitioning to the target screen

- Switch to the Start screen.
- Click on *Controls\Screen Change*.
- Position the object
  Move the cursor to the screen and draw the area for the visualization object.

Screen Change

- In the Properties 'Screen Change' pane, click on the drop-down menu arrow for the Target Screen field and select the <MachineController> screen.

In order to ensure that touch panel users will know what the screen change button does, you will be adding an icon to it.

- Double-click on the screen change and open the States tab.
- Click on the Off state.
- Instead of Text as foreground, select the Use an icon as foreground option.
- Select the icon you want from the Project images folder:
  factory, <for example>.
- □ Disable the Stretched option.
- Click on OK.

Now add a descriptive text next to the button.

- Click on *Controls\Text*.
- Position the object
  Move the cursor to the screen and draw the area for the visualization object.

Text

- Go to the Properties 'Text' pane and type Machine Controller into the <Text> field.
- Press the Enter key when you are done.

Exercise step 4

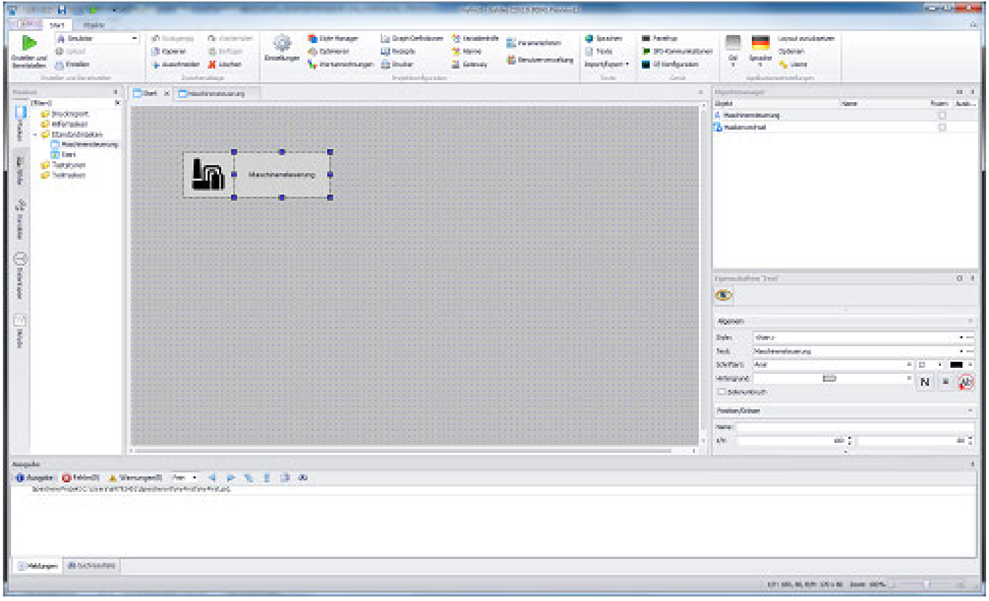


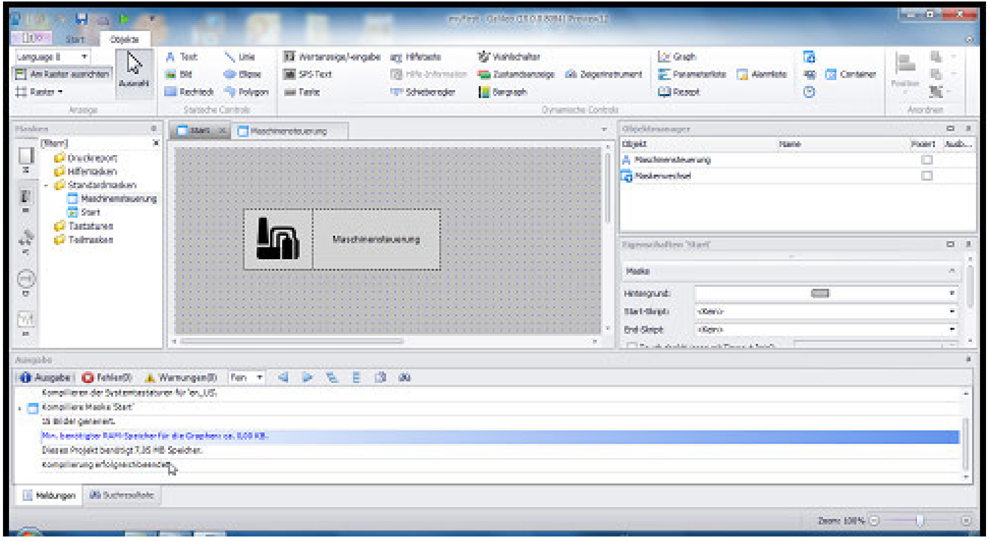
Figure 88: The screen change and text added to the MachineController screen

The Start screen now features a screen change object shown as a button with an icon, as well as a descriptive text next to this button.

You can test the project at this point already.

- Start the Galileo Simulator with function key F5 or by clicking on Home\ . Build.

Exercise step 5



```
Kompilieren der Systemtastaturen für 'en_US'.
Kompiliere Maske 'Start'
15 Bilder generiert.
Min. benötigter RAM-Speicher für die Graphen ca. 8,00 KB.
Dieses Projekt benötigt 7,35 MB Speicher.
Kompilierung erfolgreich beendet.
```

Figure 89: The Output pane logs the results from the test.

✓ Compilation finished – If you get this message, you did everything correctly!

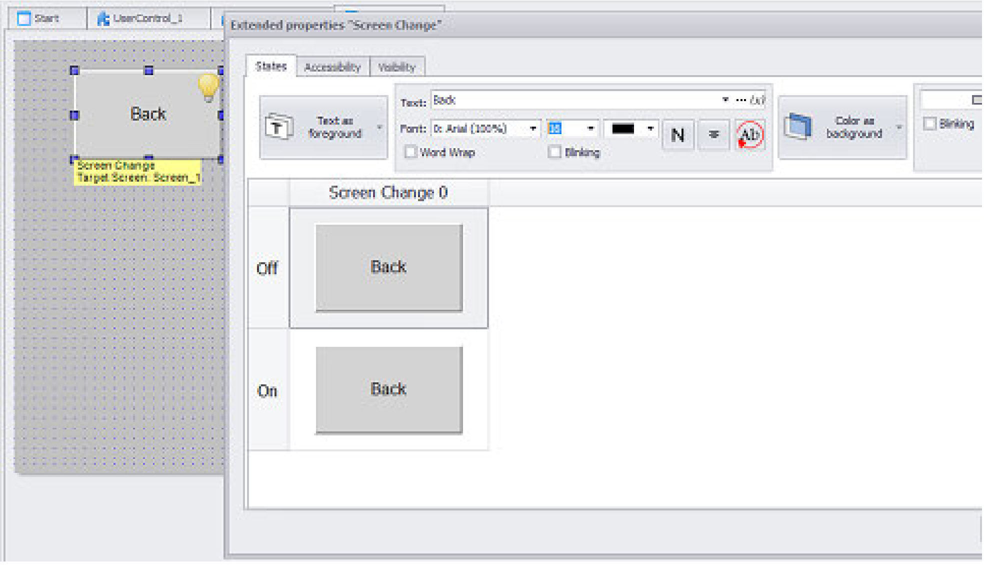
Adding a screen change (to the target screen) for transitioning back to the Start screen

- Switch to the <MachineController> screen.
- Click on *Controls\Screen Change*.
- Position the object
  Move the cursor to the screen and draw the area for the visualization object.
- Go to the Properties 'Screen Change' pane and click on the drop-down menu arrow for the Target Screen field in order to select the Start screen.

Make sure to label the various objects so that the touch panel operator will later on know what each button does.

- Double-click on the screen change object.
- Click on the States tab, type <Back> into the Text field, and center-align the text.

Exercise step 6



Extended properties "Screen Change"

Text: Back

Font: 0: Arial (100%)

Figure 90: Adding a screen change with text by using the Extended properties dialog box

- ▶ Click on OK.

The MachineController target screen now features a screen change object for switching back to the Start screen.

Adding an image as a background to the target screen

- ▶ Switch to the <MachineController> screen.
- ▶ Click on *Controls*\ *Image*.
- ▶ Position the object
  Move the cursor to the screen and draw the area for the visualization object.
- ▶ Go to the Properties 'Image' pane and, in the Image field, click on the drop-down menu arrow to select the <maschine klein> image.

An image will be added to the MachineController screen.

Exercise step 7

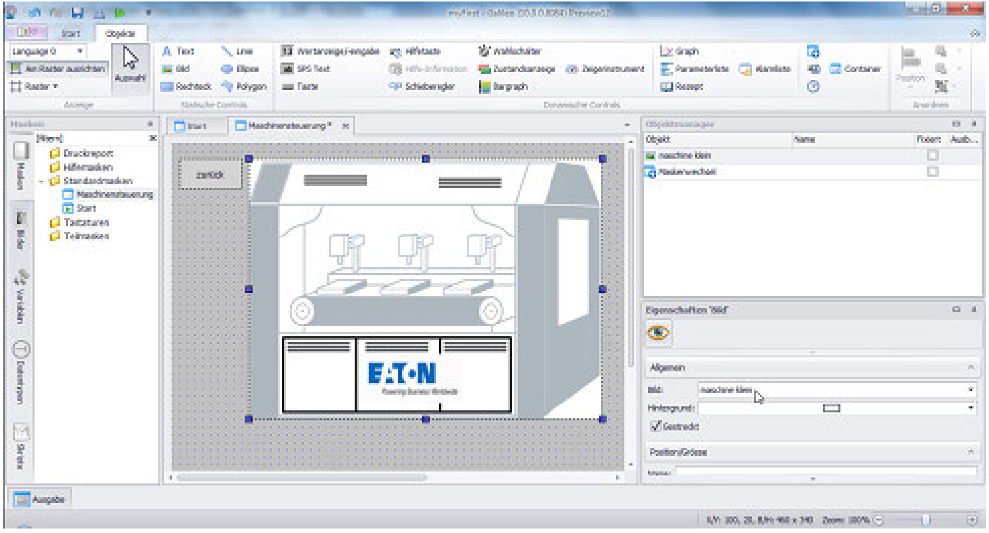


Figure 91: Background image added to the screen

Adding a button with a label to the target screen

- Switch to the <MachineController> screen.
- Click on *Controls\ Button*.
- Position the object
  Move the cursor to the screen and draw the area for the visualization object.

Make sure to label the various objects so that the touch panel operator will later on know what each button does.

- Double-click on the button visualization object.
- Under the States tab, select the Off state.
- Type the label for the OFF state into the input line, e.g., <SWITCH ON>.
- Under the States tab, select the On state.
- Type the label for the OFF state into the input line, e.g., <SWITCH OFF>.
- Click on OK.
- In the Properties 'Button' pane, click on the □ button and select the 3D 3D frame option
  so that the button will look like an actual physical button.

Exercise step 8

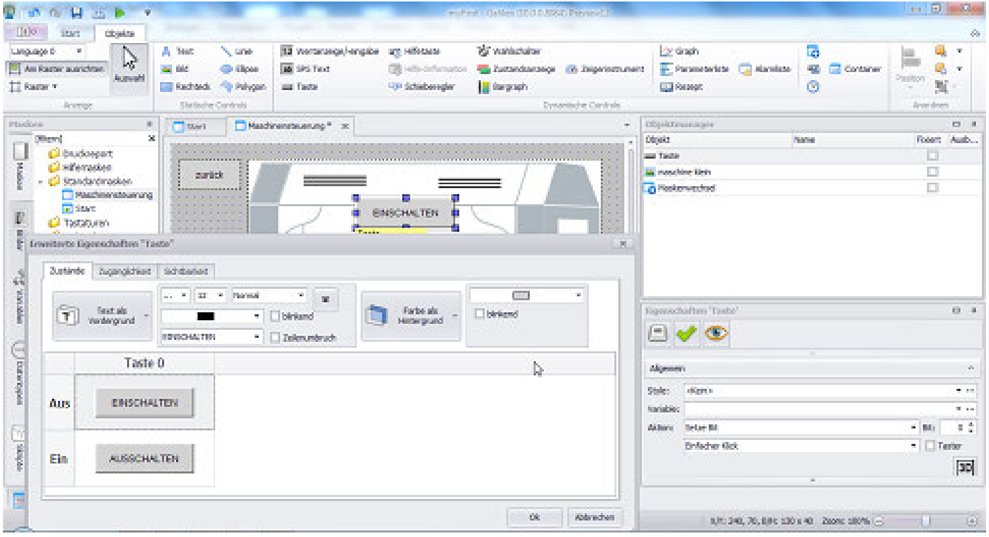


Figure 92: Button added on top of background image

You have now configured the actual button details for the button object. When the button is pressed, its appearance will change according to the corresponding state.

→ At this point, pressing the button will have no effect on the controller, as the button has not been linked to any tags yet.

Creating a bit tag and linking it to the button

- Go to the project tree.
- Click on the Tags tab.
- Right-click on the 0:CODESYS V2 folder to open the corresponding context menu
- Select the *Add\x Bit* option.

A tag called Bit0 will be created.

- Change the tag's name from Bit0 to <start_stop>.
- Click on OK.

Exercise step 9

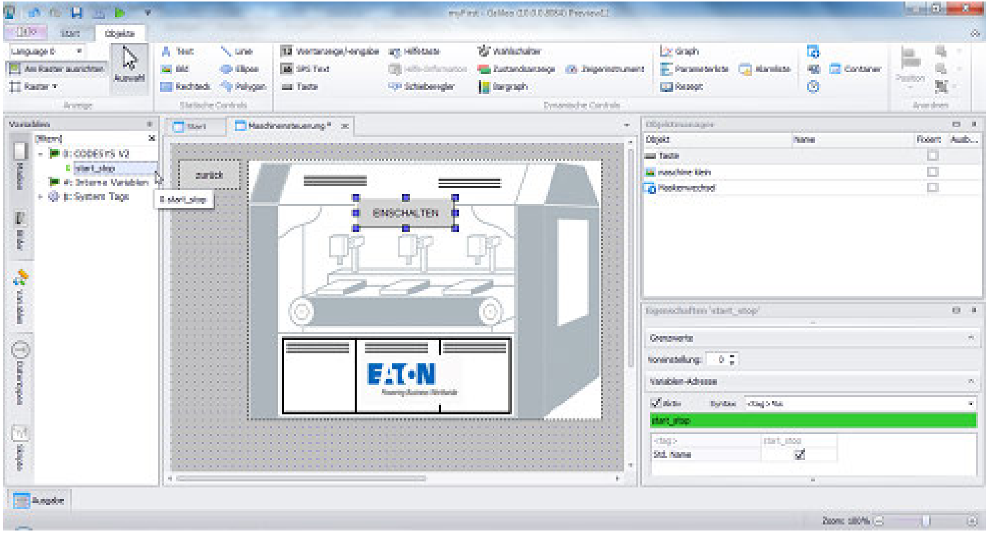


Figure 93: Tag in the project tree after having been created

- Switch to the <MachineController> screen.
- Make sure that the SWITCH ON <button> visualization object is selected on the screen.
- In the Properties 'Button' pane, click on the drop-down menu arrow for the Tag field to select the tag you just created, i.e., start_stop, from the <CODESYS V2> folder.
- Make sure that the Action field has the <Set Bit> option selected and that the Bit field is set to <0>.

Exercise step 10

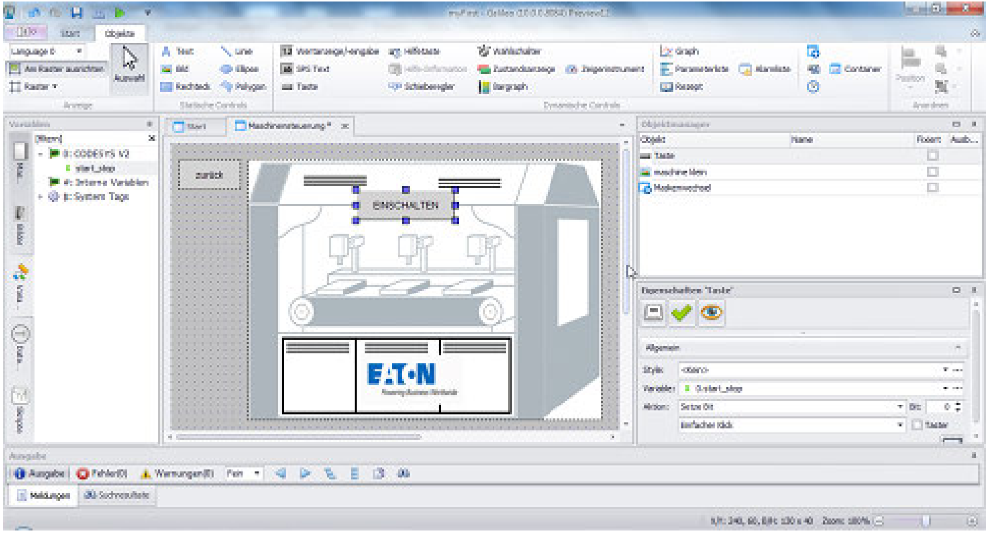


Figure 94: Tag after being assigned to the button

When the button is clicked on, the tag will be set to a value of <1>. Clicking on it again will set the tag back to <0>.

At this point, the button on the screen will show<SWITCH ON> as its text regardless of the tag's value.

Linking the bit tag to images representing the switch position

In order to visually show the machine controller's state so that the touch panel user can see it, you will be using the <start_stop> bit's state to show different images.
To do this, follow the steps below:

- Go to the project tree.
- Open the Images tab.
- Open the Project images folder.
- Drag and drop the <schalt_on.png> and <schalt_off.png> images from the project tree onto the <MachineController> screen.
- Select the <schalt_on.png> image that should be shown when the machine controller is switched on.
- In the Properties 'Image' pane, click 👁 to open theVisibility tab.
- Click on Add condition. Select the <start_stop> bit tag as the tag for the condition.
- In the Logic column, select the is OFF (=0) condition.

This means that the schalt_on.png image will be hidden when <start_stop> = 0.

- Now select the < schalt_off.png> image that should be shown when the machine controller is switched off.
- In the Properties 'Image' pane, click 👁 to open theVisibility tab.
- Click on Add condition. Select a bit tag as the tag for the condition, e.g., <start_stop>.
- In the Logic column, select the is ON (=1) condition.
- Click on OK.

This means that the schalt_off.png image will be hidden when <start_stop> = 1.

Exercise step 11

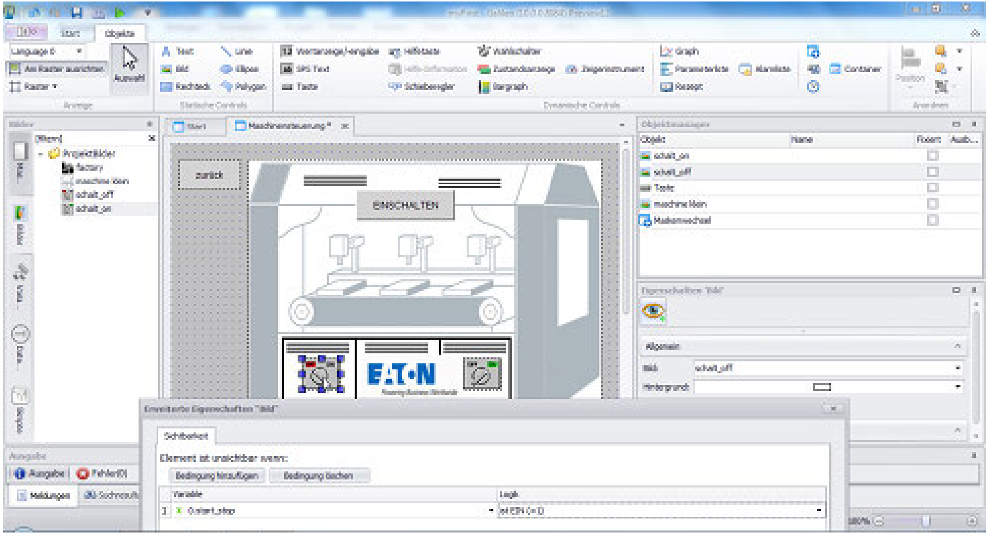


Figure 95: Bit tag assigned to the images

- ▶ Finally, move the two visualization objects on top of each other on the screen so that it will look like the switch moves when the controller is switched.

Testing the project in the Galileo Simulator

You can now test the project for this exercise.

- ▶ Start the Galileo Simulator by pressing F6 or by clicking on Home\ ▶ Build and Deploy.

The <myFirst> project will be compiled and opened on your computer with the Galileo Simulator.

Exercise step 12

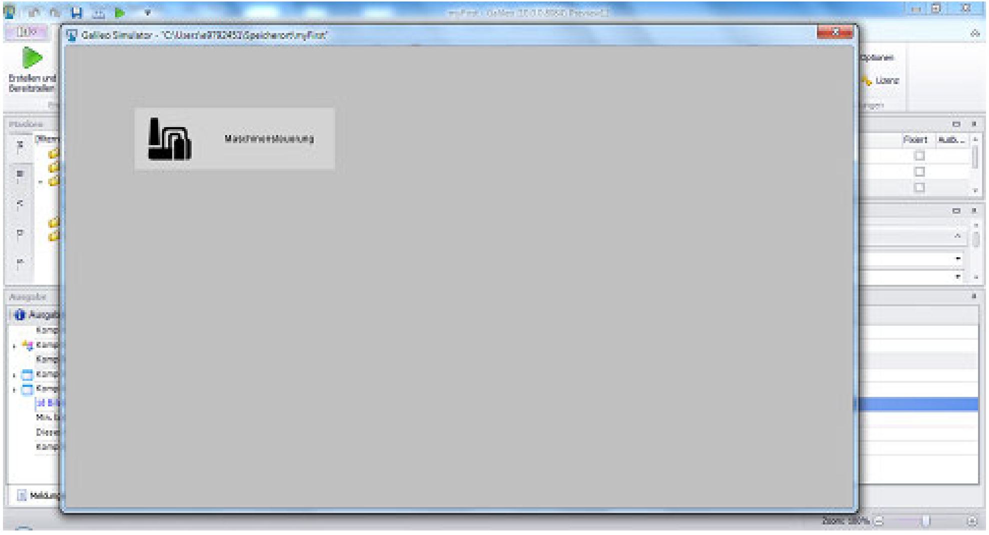

Erstellen und Bereitstellen
Galileo Simulator - 'C:\Users\...\myFirst'
Maschinensteuerung
Ausgabe
Meldungen

Figure 96: The Output pane logs the results from the test and the project is opened

- Check that the project works correctly by clicking on the corresponding elements.

1.6.4.4 Step 1: Creating a new project

To create your first project, use the New Project... Wizard

The wizard will take you through a number of dialog boxes (four or six depending on your selection) used to obtain the required information and configure the necessary settings.

- Click on the project management tab on the home screen
- Click on New.
- Under New, select Project.

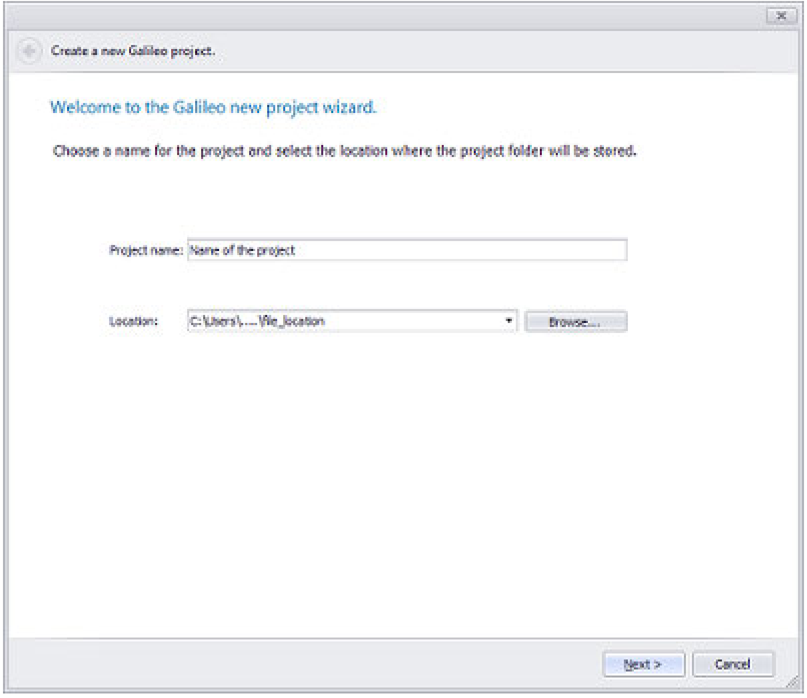
The wizard will open in a new dialog box on your monitor.

Substep 1. Specifying a project name and storage location

Give your project a name and specify the storage location you want on your system.

- Enter the project name you want into the **Project name** field
- Enter the storage location for your project.
  To do so, you can either enter the path directly, e.g.,**C:\Users\..\StorageLocation**
  or
  You can click on Browse... to browse for a folder on your computer's hard drive or on your network.
  Clicking on Make New Folder will create a new folder in the path being shown.
  Click on OK.

Project Management\ New Project ...



Create a new Galileo project.

Welcome to the Galileo new project wizard.

Choose a name for the project and select the location where the project folder will be stored.

Project name: Name of the project

Location: C:\Users\.....\file_location

Figure 97: Wizard New Project, step 1

- ▶ Click on Next >.

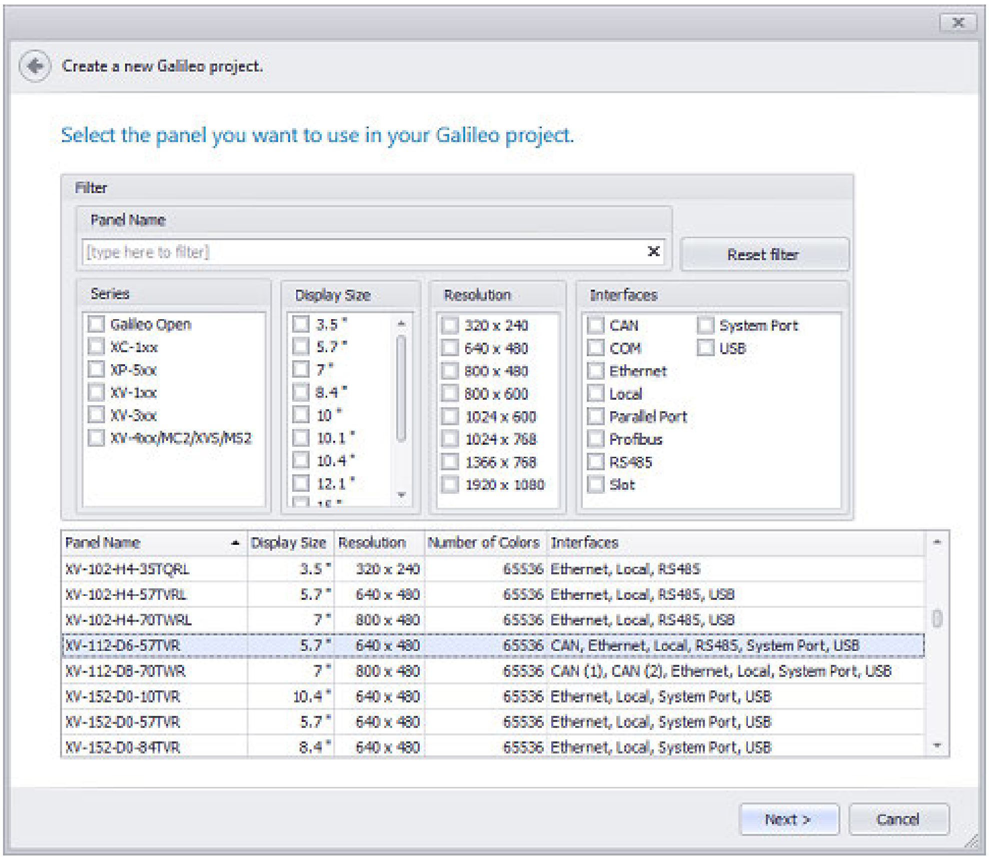
The dialog box will show the next step.

Substep 2. Selecting a panel model

The purpose of the project you will be creating is to transfer it to a specific panel so that it will function as a user interface there. In order to help you work more efficiently, the wizard will ask you for the panel name at this point when creating a new project. The name will contain the panel's display size, resolution, and available ports and interfaces in coded format.
The reason it is a good idea to select a panel model at this point is that if, for instance, the display size is set later on, it will be necessary to adjust all screens as necessary for the display size, which can end up being rather time-consuming.

Only one panel can be selected per project.

Project Management\ New Project ...



Create a new Galileo project.

Select the panel you want to use in your Galileo project.

Filter

Panel Name

[type here to filter]

Series

□ Galileo Open
□ XC-1xx
□ XP-5xx
□ XV-1xx
□ XV-3xx
□ XV-4xx/MC2/XVS/MS2

Display Size

□ 3.5 "
□ 5.7 "
□ 7 "
□ 8.4 "
□ 10 "
□ 10.1 "
□ 10.4 "
□ 12.1 "
□ 15 "

Resolution

□ 320 x 240
□ 640 x 480
□ 800 x 480
□ 800 x 600
□ 1024 x 600
□ 1024 x 768
□ 1366 x 768
□ 1920 x 1080

Interfaces

□ CAN
□ COM
□ Ethernet
□ Local
□ Parallel Port
□ Profibus
□ RS485
□ Slot
□ System Port
□ USB

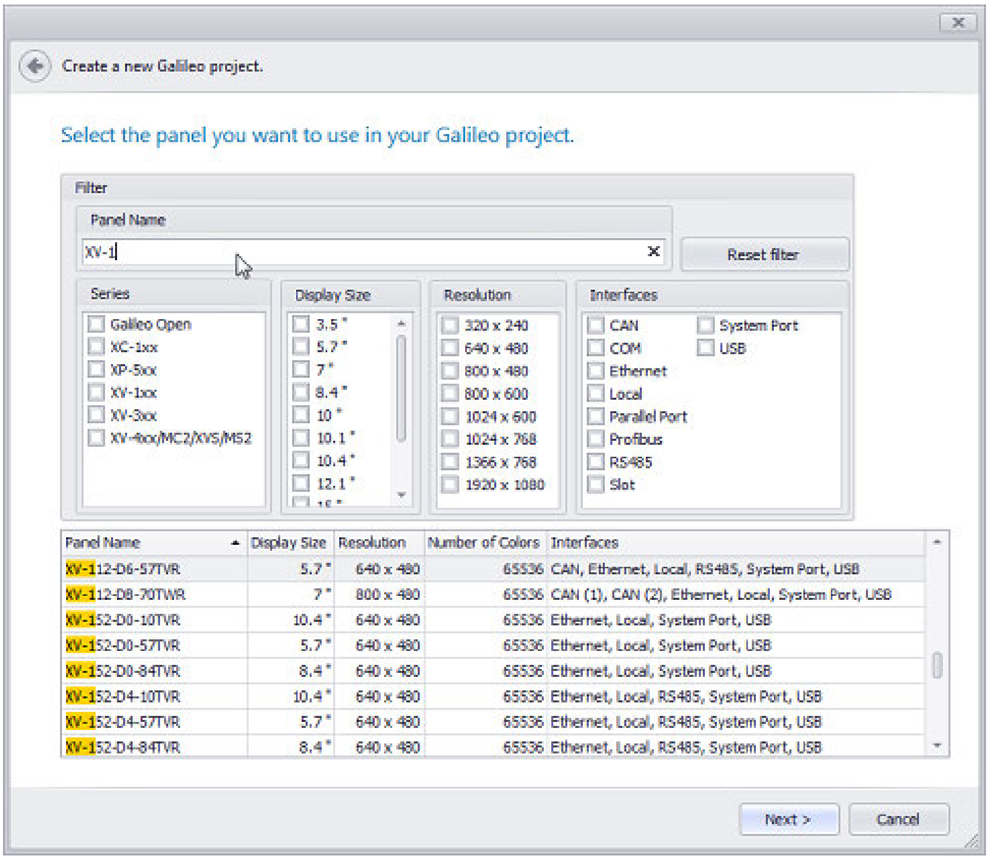
Panel Name	Display Size	Resolution	Number of Colors	Interfaces
XV-102-H4-35TQRL	3.5 "	320 x 240	65536	Ethernet, Local, RS485
XV-102-H4-57TVRL	5.7 "	640 x 480	65536	Ethernet, Local, RS485, USB
XV-102-H4-70TWRL	7 "	800 x 480	65536	Ethernet, Local, RS485, USB
XV-112-D6-57TVR	5.7 "	640 x 480	65536	CAN, Ethernet, Local, RS485, System Port, USB
XV-112-D8-70TWR	7 "	800 x 480	65536	CAN (1), CAN (2), Ethernet, Local, System Port, USB
XV-152-D0-10TVR	10.4 "	640 x 480	65536	Ethernet, Local, System Port, USB
XV-152-D0-57TVR	5.7 "	640 x 480	65536	Ethernet, Local, System Port, USB
XV-152-D0-84TVR	8.4 "	640 x 480	65536	Ethernet, Local, System Port, USB

Figure 98: New Project Wizard, step 2

If you do not know which panel you will be using yet, you can change it later on.

Please follow the instructions for this example for your first project.

▶ Enter **XV-1** into the input field for the panel name



Create a new Galileo project.

Select the panel you want to use in your Galileo project.

Filter

Panel Name: XV-1

Series
- □ Galileo Open
- □ XC-1xx
- □ XP-5xx
- □ XV-1xx
- □ XV-3xx
- □ XV-4xx/MC2/XVS/MS2

Display Size
- □ 3.5 "
- □ 5.7 "
- □ 7 "
- □ 8.4 "
- □ 10 "
- □ 10.1 "
- □ 10.4 "
- □ 12.1 "
- □ 15 "

Resolution
- □ 320 x 240
- □ 640 x 480
- □ 800 x 480
- □ 800 x 600
- □ 1024 x 600
- □ 1024 x 768
- □ 1366 x 768
- □ 1920 x 1080

Interfaces
- □ CAN
- □ COM
- □ Ethernet
- □ Local
- □ Parallel Port
- □ Profibus
- □ RS485
- □ Slot
- □ System Port
- □ USB

Panel Name	Display Size	Resolution	Number of Colors	Interfaces
XV-112-D6-57TVR	5.7 "	640 x 480	65536	CAN, Ethernet, Local, RS485, System Port, USB
XV-112-D8-70TWR	7 "	800 x 480	65536	CAN (1), CAN (2), Ethernet, Local, System Port, USB
XV-152-D0-10TVR	10.4 "	640 x 480	65536	Ethernet, Local, System Port, USB
XV-152-D0-57TVR	5.7 "	640 x 480	65536	Ethernet, Local, System Port, USB
XV-152-D0-84TVR	8.4 "	640 x 480	65536	Ethernet, Local, System Port, USB
XV-152-D4-10TVR	10.4 "	640 x 480	65536	Ethernet, Local, RS485, System Port, USB
XV-152-D4-57TVR	5.7 "	640 x 480	65536	Ethernet, Local, RS485, System Port, USB
XV-152-D4-84TVR	8.4 "	640 x 480	65536	Ethernet, Local, RS485, System Port, USB

Figure 99: New Project Wizard, step 2

The options available for selection in the lower pane will change automatically to reflect the options you select above and any text you enter into the Panel Name field.

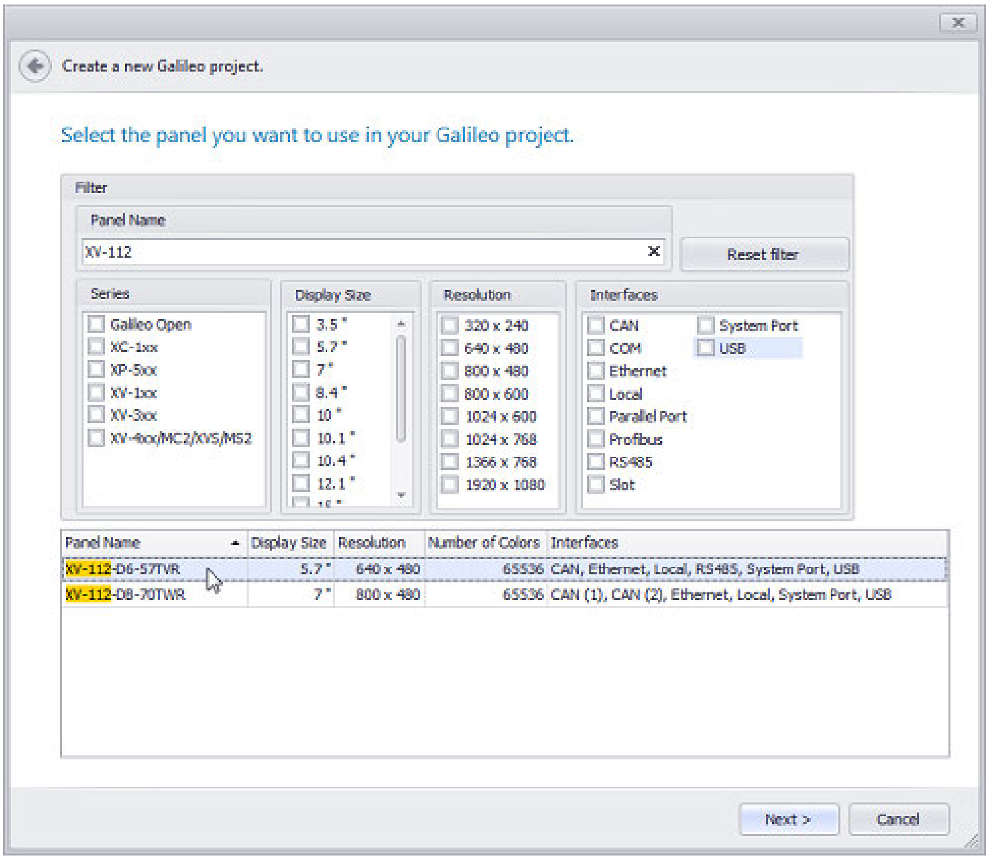
Clicking on a panel will apply the corresponding parameters internally. However, the various options on the dialog box will not be changed.

Double-clicking on a panel will select the panel model and cause the dialog box to go to the next project creation step.

▶ Enter **XV-112** into the input field for the panel name

The available selection of panels will be narrowed further.

▶ Click on **XV-112-D6-57TVR** to select the corresponding model.



Create a new Galileo project.

Select the panel you want to use in your Galileo project.

Filter

Panel Name: XV-112

Series: □ Galileo Open □ XC-1xx □ XP-5xx □ XV-1xx □ XV-3xx □ XV-4xx/MC2/XVS/MS2

Display Size: □ 3.5 " □ 5.7 " □ 7 " □ 8.4 " □ 10 " □ 10.1 " □ 10.4 " □ 12.1 " □ 15 "

Resolution: □ 320 x 240 □ 640 x 480 □ 800 x 480 □ 800 x 600 □ 1024 x 600 □ 1024 x 768 □ 1366 x 768 □ 1920 x 1080

Interfaces: □ CAN □ COM □ Ethernet □ Local □ Parallel Port □ Profibus □ RS485 □ Slot □ System Port □ USB

Panel Name	Display Size	Resolution	Number of Colors	Interfaces
XV-112-D6-57TVR	5.7 "	640 x 480	65536	CAN, Ethernet, Local, RS485, System Port, USB
XV-112-D8-70TWR	7 "	800 x 480	65536	CAN (1), CAN (2), Ethernet, Local, System Port, USB

Figure 100: New Project Wizard, step 2

The item you click on will be highlighted in blue.

- Click on Next >.

Substep 3. Selecting communication methods

Every project can be configured for up to eight different communication methods via PLC interfaces.

Select at least one communication method for your project.
If you want, you can use the Interface field to select an interface, which will narrow the list of communication methods available for selection.
You can also select a communication method directly from the list if you want.

Add the selected communication method to your project either by double-clicking on it or by clicking on the Add button.

The selected communication methods can be changed and/or added to later on by clicking on Device\ PLC Communication

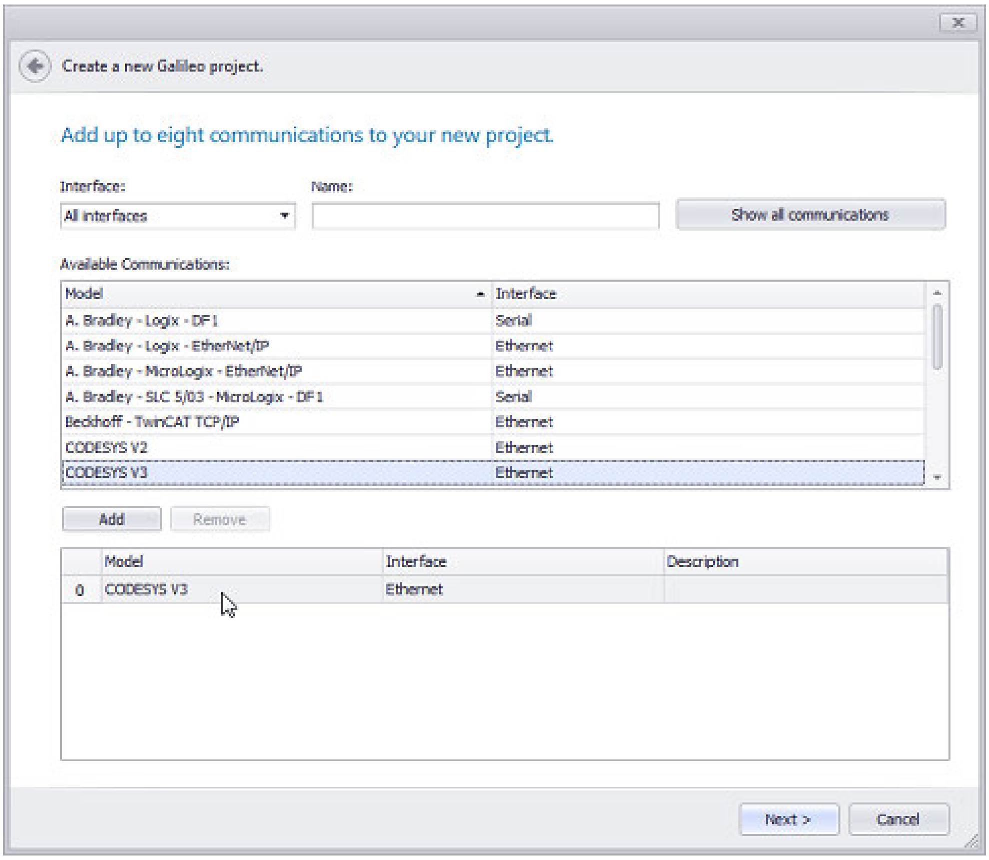
- Select CODESYS V3 for your first project.
- Click on Add.

CODESYS V3 will be added to the list of selected communication methods (Nos. 0–7)

▶ Click on Next >

The selected communication methods can be changed and/or added to later on by clicking on Home\Device\ PLC Communication

Project Management\ New Project ...



Create a new Galileo project.

Add up to eight communications to your new project.

Interface: All interfaces

Name:

Available Communications:

Model	Interface
A. Bradley - Logix - DF1	Serial
A. Bradley - Logix - EtherNet/IP	Ethernet
A. Bradley - MicroLogix - EtherNet/IP	Ethernet
A. Bradley - SLC 5/03 - MicroLogix - DF1	Serial
Beckhoff - TwinCAT TCP/IP	Ethernet
CODESYS V2	Ethernet
CODESYS V3	Ethernet

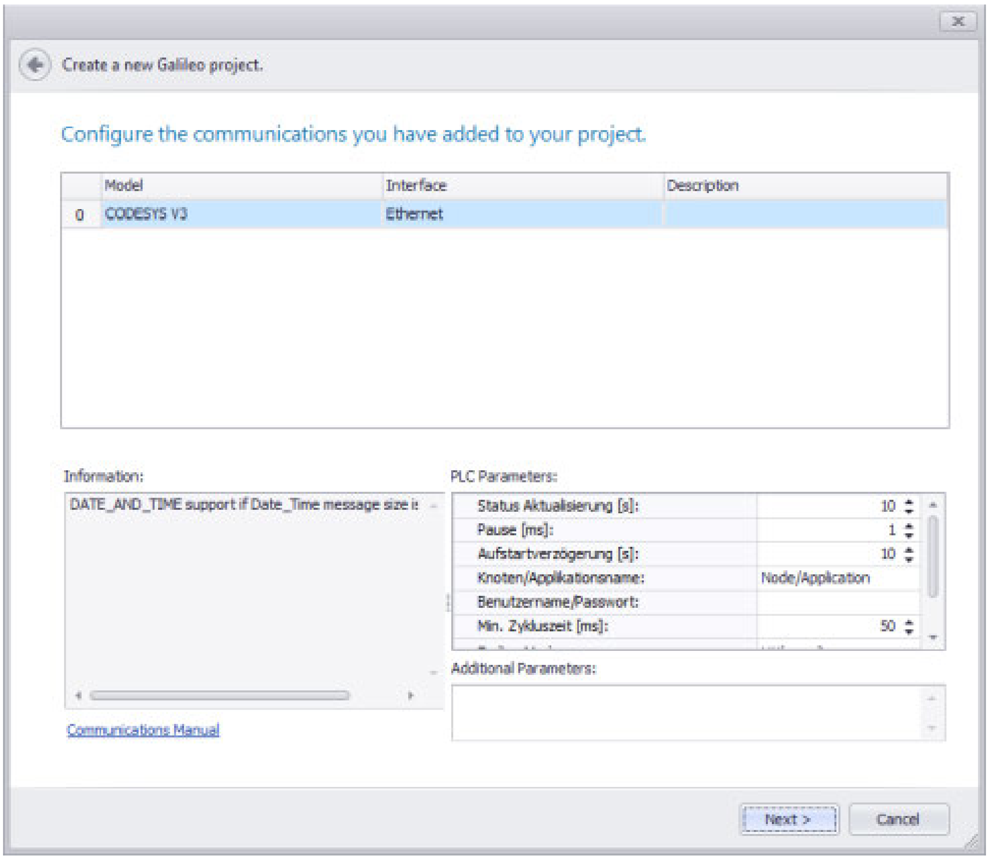
	Model	Interface	Description
0	CODESYS V3	Ethernet	

Figure 101: New Project Wizard, step 3 – adding communication methods to the project

▶ Configure the communication methods you added in the PLC Parameters pane.

→ You can skip this step for your first project.
For more information, please refer to the following section: → "Select Communication", page 581

Project Management\ New Project ...



Create a new Galileo project.

Configure the communications you have added to your project.

	Model	Interface	Description
0	CODESYS V3	Ethernet	

Information:

DATE_AND_TIME support if Date_Time message size i:

Communications Manual

PLC Parameters:

Status Aktualisierung [s]:	10
Pause [ms]:	1
Aufstartverzögerung [s]:	10
Knoten/Applikationsname:	Node/Application
Benutzername/Passwort:	
Min. Zykluszeit [ms]:	50

Additional Parameters:

Figure 102: New Project Wizard, step 3 – configuring communication methods

▶ Click on Next >

Substep 4. Selecting a style (optional)

Applies only to panels belonging to the XV300 and XP500 product groups, as well as to Galileo Open projects

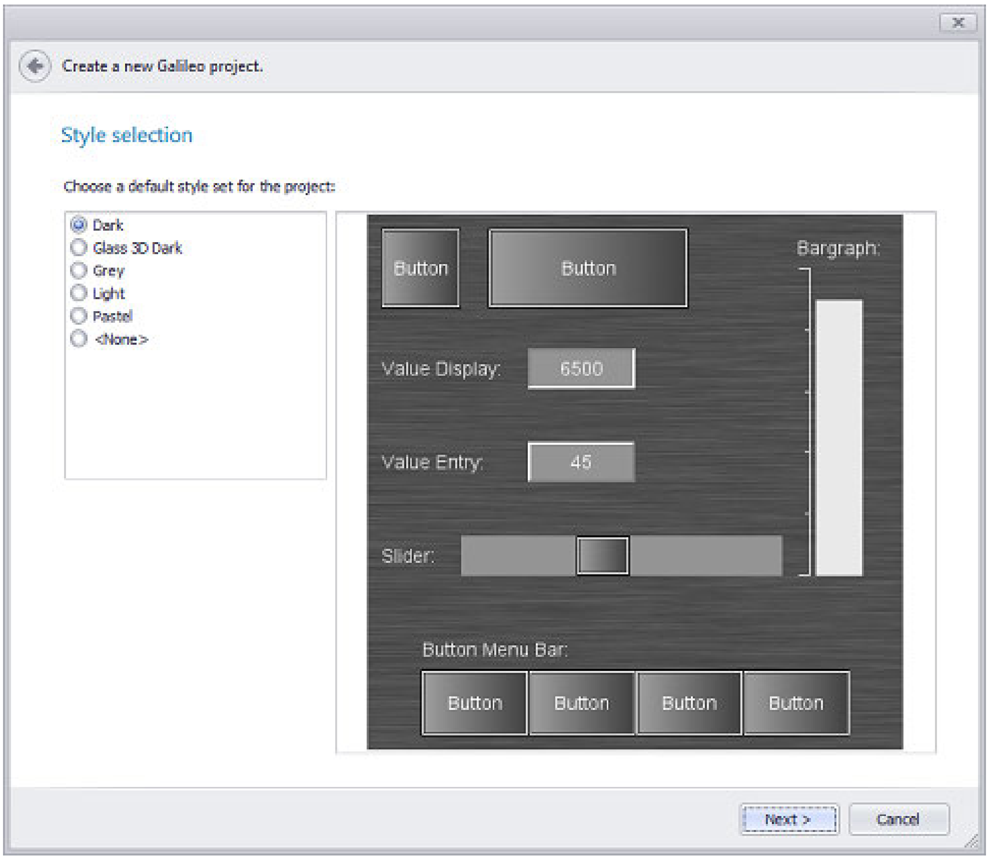
If the panel you select supports certain functionalities, you will be able to select a basic option for the project's appearance in this dialog box.

A pre-defined default style set for dynamic controls can be selected for each project.

The preview on the right side will give you an idea of how the objects will be displayed in your project.

→ This option is not available for your first project, which uses the XV-112-D6-57TVR.
For more information, please refer to the following section: → section "Style management", page 458

Project Management\ New Project ...



Create a new Galileo project.

Style selection

Choose a default style set for the project:

- ● Dark
- ○ Glass 3D Dark
- ○ Grey
- ○ Light
- ○ Pastel
- ○ <None>

Figure 103: New Project Wizard, step 4 (option) Selecting a style set

▶ Click on Next >

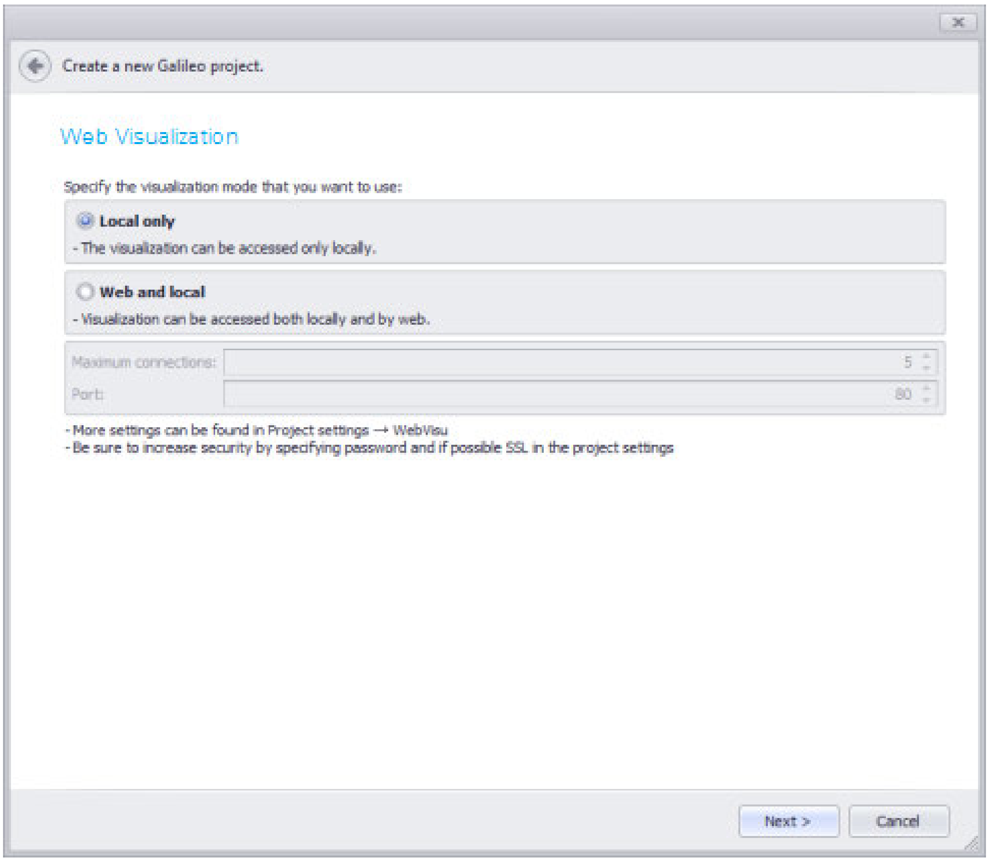
Substep 5. Selecting the type of visualization

GALILEO features a service that

makes it possible to display and control the GALILEO application not just locally on a touch panel, but also on other touch panel or PCs with the use of web clients.

Use this dialog box to select the visualization mode you want to use.

Project Management\ New Project ...



Create a new Galileo project.

Web Visualization

Specify the visualization mode that you want to use:

● Local only
- The visualization can be accessed only locally.

○ Web and local
- Visualization can be accessed both locally and by web.

Maximum connections: 5
Port: 80

- More settings can be found in Project settings → WebVisu
- Be sure to increase security by specifying password and if possible SSL in the project settings

Figure 104: New Project Wizard, step 5 – selecting a visualization mode

You can change and/or expand the mode you select later on by going to Home\ Project Configuration\ Settings\WebVisu

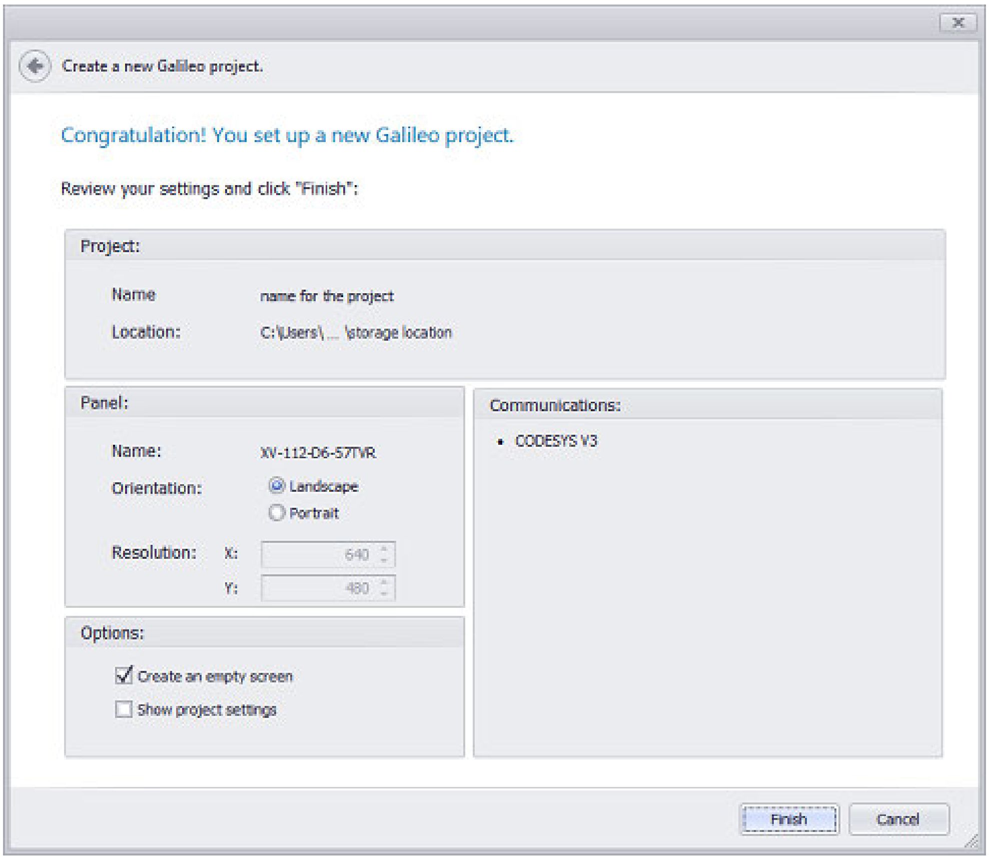
- Click on Next >

Substep 6. Verification of the settings

- Congratulation! You have created a new Galileo project.
- Review your settings and click "Finish":

If you make any mistakes or change your mind while configuring the settings in the wizard, you can click on the arrow on the top left of the dialog box to go back step by step and make the necessary changes.

Project Management\ New Project ...



Create a new Galileo project.

Congratulation! You set up a new Galileo project.

Review your settings and click "Finish":

Project:

Name name for the project

Location: C:\Users\ ... \storage location

Panel:

Name: XV-112-D6-57TVR

Orientation: ● Landscape ○ Portrait

Resolution: X: 640 Y: 480

Options:

☑ Create an empty screen

□ Show project settings

Communications:

- CODESYS V3

Finish Cancel

Figure 105: New Project Wizard, step 6 – reviewing project settings

Alignment:

If you already know that you will be using your panel in portrait mode, you can change the screens' orientation here directly. You can also change this orientation later in the project by using the Panel dialog box.

→ section "Display Settings", page 225

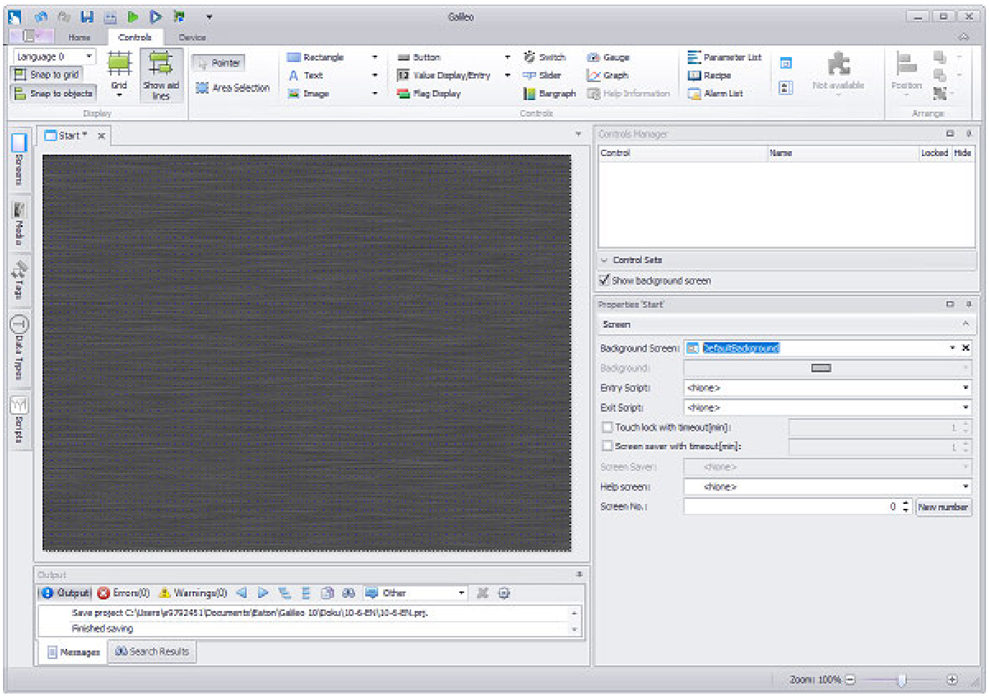
☑ If you enable the Show project settings option, the Project Settings dialog box will be opened immediately after you click on Finish.
→ section "Project Configuration", page 196

▶ Click on Finish.

The GALILEO user interface will appear and will already contain the first standard screen, which will automatically be named Start.

The project tree can be used to change screen names and set a different screen as the start screen.
→ section "Project tree", page 64

View after closing the New Project... Wizard



☑ Show background screen

Properties 'Start'

Screen

Background Screen: DefaultBackground
Background:
Entry Script: <None>
Exit Script: <None>
☐ Touch lock with timeout[min]: 1
☐ Screen saver with timeout[min]: 1
Screen Saver: <None>
Help screen: <None>
Screen No.: 0 New number

Save project C:\Users\pS792451\Documents\Eaton\Galileo 10\Doku\10-6-EN\10-6-EN.prj.
Finished saving

Figure 106: Start standard screen in new project

Before starting to work on your project, check the basic project settings in the next step.

Continue to the following step with your first project:

→ "Step 2: Change Project Settings", page 168

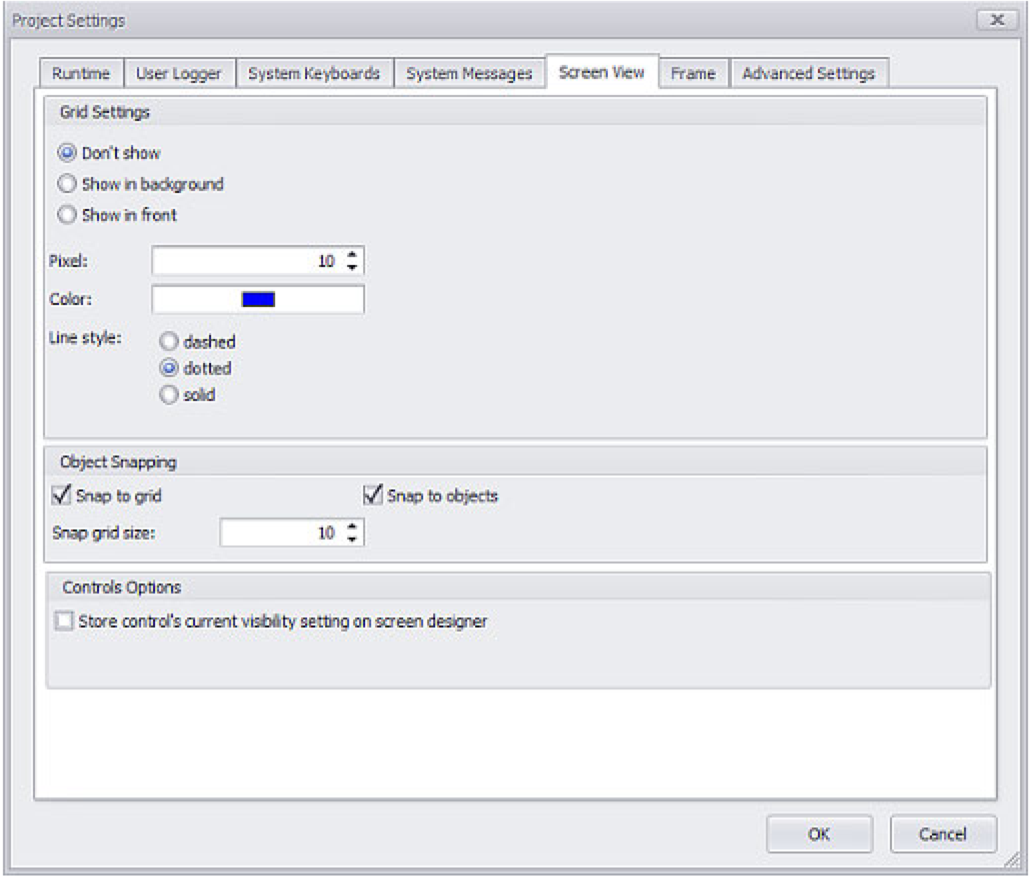
1.6.4.5 Step 2: Change Project Settings

For this first project, you will only be configuring the options for the grid and for object snapping.

→ For more information on the other project configuration options, which are not yet needed for your first project,
please refer to section:
→ section "Project Configuration", page 196

▶ Click on Home\ Settings.
▶ Click on the Screen View tab.

Home\ Settings tab



Project Settings

Runtime | User Logger | System Keyboards | System Messages | Screen View | Frame | Advanced Settings

Grid Settings

● Don't show
○ Show in background
○ Show in front

Pixel: 10

Color: 🟦

Line style:
○ dashed
● dotted
○ solid

Object Snapping

☑ Snap to grid ☑ Snap to objects

Snap grid size: 10

Controls Options

□ Store control's current visibility setting on screen designer

Figure 107: *Project Settings\Screen View*

Grid Settings

- Enable the Show in background ◉ option in the Grid Settings section
- Set the grid to 10 pixels.
- Select blue as the color.
- Select ◉ dotted as the line style

Object Snapping

- Enable the enabled Snap to grid option in the ☑ section
- Enable the enabled Snap to objects option in the ☑
- Set the snap grid size to 10 pixels.

Continue to the following step with your first project:

→ "Step 3: Configuring the start screen's properties", page 169

1.6.4.6 Step 3: Configuring the start screen's properties

For every screen and every object placed on the screen, the Properties pane will show the corresponding configuration options.

▶ Click on the "Start" screen in the *Project Tree\Screens\Standard screens folder*.

Project Tree\Screens

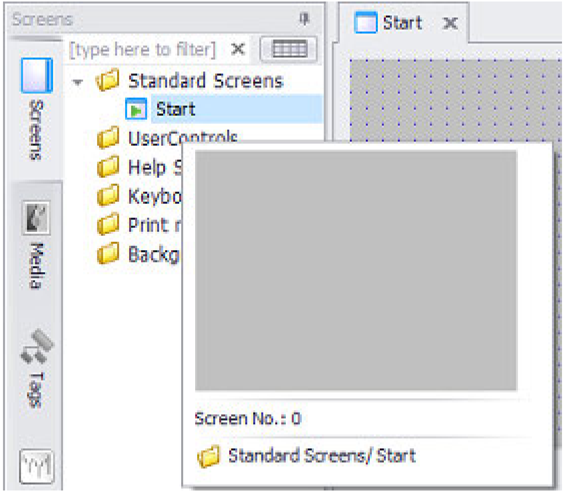
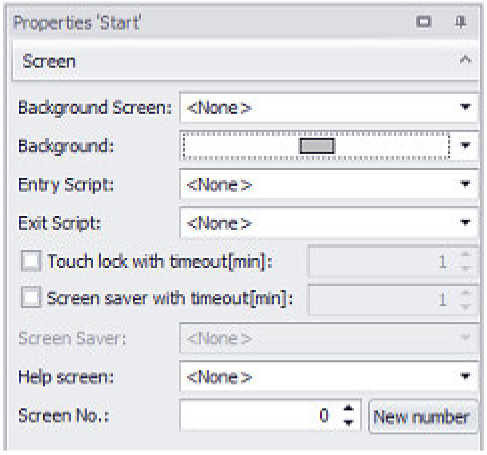


Figure 108: Viewing the "Start" screen

The Properties 'Start' pane will be filled out with the settings for the screen.

Properties pane for the Start screen



Properties 'Start'

Screen

Background Screen: <None>

Background:

Entry Script: <None>

Exit Script: <None>

□ Touch lock with timeout[min]: 1

□ Screen saver with timeout[min]: 1

Screen Saver: <None>

Help screen: <None>

Screen No.: 0 New number

Figure 109: Start screen properties

▶ Select the following background color: light gray [RGB 233;233;233].

Properties pane for the Start screen

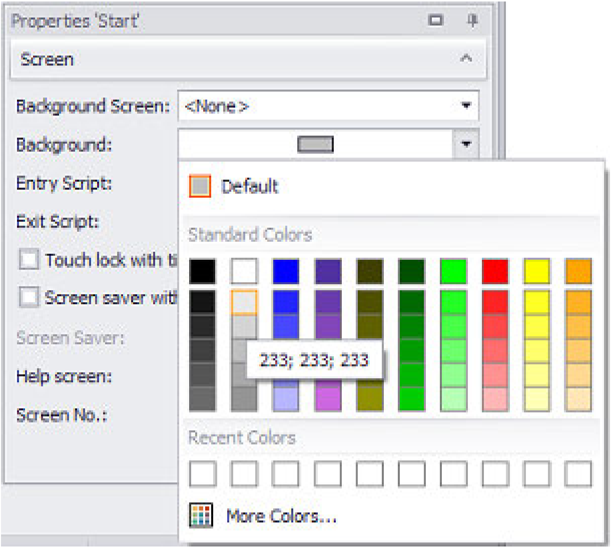


Figure 110: Viewing the "Start" screen

The screen's background will be shown in the selected color.

Save the project

Every time after you make changes to the screen, an asterisk (*) will be shown after the screen name until you save the project.

To save the project, use the quick access toolbar.

Screenshot showing how the first project should look at this point

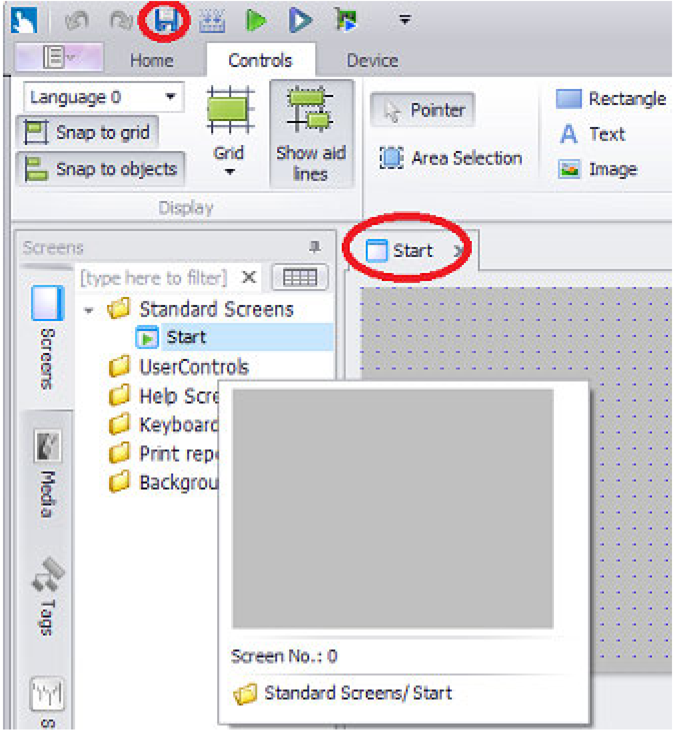


Figure 111: Start screen changes not saved yet

- ▶ Click on the 💾 icon.

Your first project will be saved in the specified storage location using its original name.

Continue to the following step with your first project:

→ "Step 4: Creating an additional standard screen", page 172

1.6.4.7 Step 4: Creating an additional standard screen

All screen types are created and managed in the project tree's Screens tab pane. This pane has a separate folder for each screen type, helping make projects more manageable.

- ▶ Go to the project tree.
- ▶ Click on the Screens tab.
- ▶ Open the context menu by right-clicking on the Standard screens folder
- ▶ Click on Add new screen

Project Tree\Screens

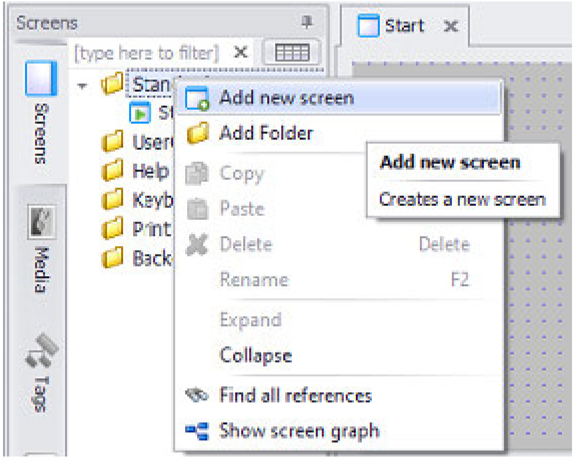


Figure 112: Creating a new standard screen

A new screen will be created and opened

The name <Screen> will be active in Project Tree\Screens\Standard screens so that you can rename the screen.

This can also be done by using the corresponding option in the context menu for the screen icon.

Project Screen\Rename F2

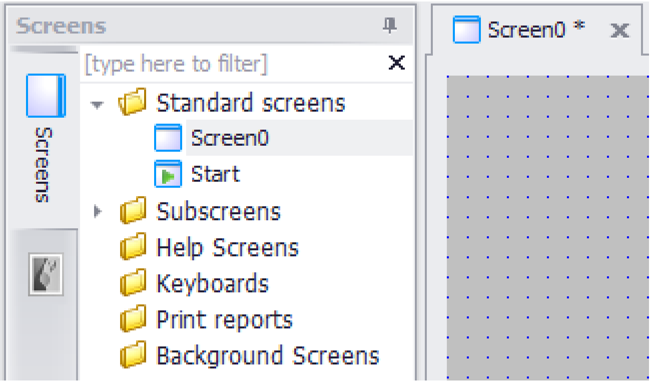


Figure 113: Creating a new standard screen

- ▶ Give the screen a <name>; spaces and special characters are not permitted.

If you click somewhere else without renaming the screen, the <Screen0> name assigned by the application will be kept.

Project Screen\Rename F2

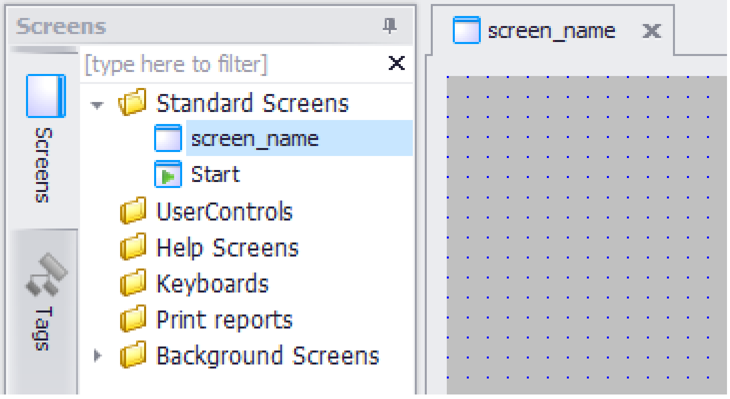


Figure 114: Renaming a standard screen

Setting a standard screen as the start screen

It is necessary to specify which screen will be the start screen, i.e., the project start screen with which the project will start on the panel.

The start screen will be identified with the ▶ icon placed before the screen's name.

→ Please note that only one of the standard screens can be the project start screen.

- ▶ Go to the project tree.
- ▶ Click on the Screens tab.
- ▶ Right-click on the icon for the screen that you want to set as the start screen in order to open the corresponding context menu.
- ▶ Click on ▶ Set as startup screen.

The ▶ icon will be shown before the defined start screen. All other standard screens will be shown with the □ icon

Similar procedure for creating other types of screens

- Go to the project tree's Screens tab pane and select the folder for the type of screen you want.
- Right-click on the folder to add a new screen.
- Press F2 to rename the screen.

Continue to the following step with your first project:

→ "Step 5: Adding an image", page 174

1.6.4.8 Step 5: Adding an image

Prerequisites

You must have an image file on your computer already, e.g., your company's logo.
The project image must be in one of the following formats: *.png, *.jpg/*.jpeg , *.tiff/*.tif, *.gif , *.svg or *.bmp,
image sequences with the *.gif format.
The project video must be in one of the following formats: *.mp4, *.mov or *.avi.
Icon library images are in .svg format.

Before you can add images to a screen with the Properties pane, the images must be imported as a medium into the project using the project tree. Once you have imported the images, you will be able to insert them into the screen freely or in previously positioned boxes with a specific size.

Substep 1: Adding an image (medium) to the project tree

- Go to the project tree.
- Open the Media tab.

When a project is created, a Project images folder will be automatically created as well.

There are two options for adding images:

Dragging&dropping:

- Open your computer's directory structure and look for the pictures that you want to display inside the container.
- Drag the images from the directory structure and drop them onto the Project images folder

or import:

- Right-click on the Project images folder to open the corresponding context menu.
- Click on Import images.

Project\Media

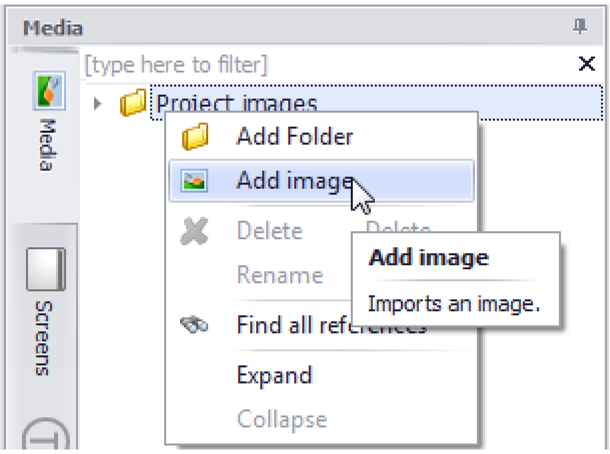


Figure 115: Importing images into the project using the context menu

Use the Open dialog box that appears to look for the image in the directory structure on your computer's hard drive or on your network.

The project image must be in one of the following formats: *.png, *.jpg/*.jpeg , *.tiff/*.tif, *.gif , *.svg or *.bmp,

image sequences with the *.gif format.

The project video must be in one of the following formats: *.mp4, *.mov or *.avi.

Icon library images are in .svg format.

For demonstration purposes, a folder called Image Catalog containing subfolders for the various possible image formats has been created for the first project at the corresponding storage location.

- ▶ Select one of your images, e.g., in JPG format.
- ▶ Click on Open to import the image into the project tree.

Project\Media

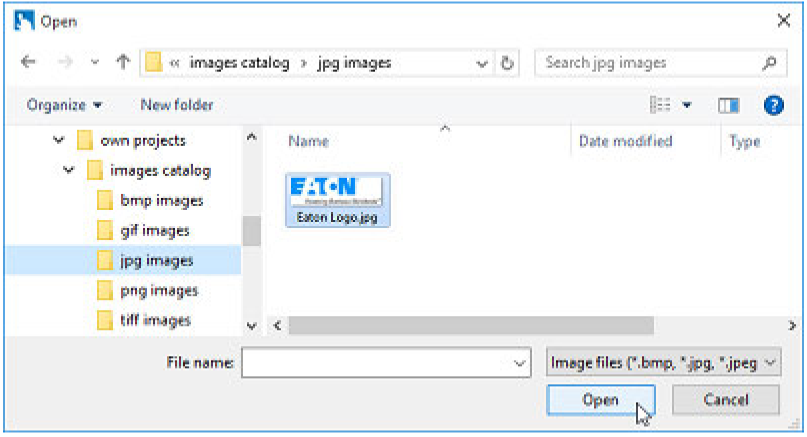


Figure 116: Selecting an image in the directory structure

When you click on "Open," the image will be converted to PNG format. GALILEO uses only png-images in the project

The imported image will be shown with a thumbnail in the Project images tab section of the project tree.
If you move the cursor over this thumbnail, a larger thumbnail will be shown together with the image width and height in pixels.

Images marked with the ⚠ icon are corrupted or do not have the right format.

Project\Media

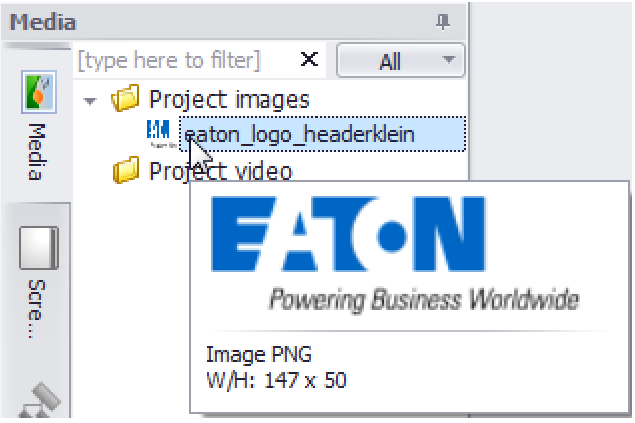


Image PNG
W/H: 147 x 50

Figure 117: Viewing the converted image in the project tree

→ You can drag
media from the project tree in its original size and drop it onto the screen in the position you want.

The image will be added in the Controls Manager and

the Properties 'Image' pane will show the settings and information for the inserted image.

Project\Media

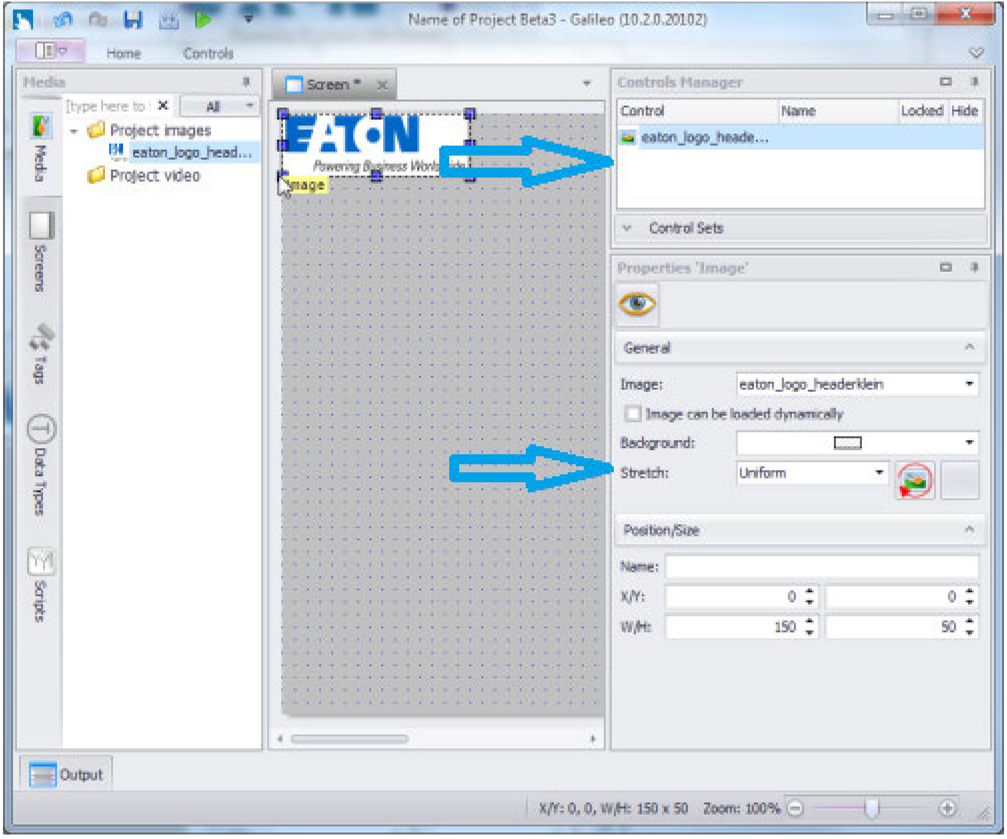


Image: eaton_logo_headerklein

☐ Image can be loaded dynamically

Background:

Stretch: Uniform

Name:

X/Y: 0 0

W/H: 150 50

X/Y: 0, 0, W/H: 150 x 50 Zoom: 100%

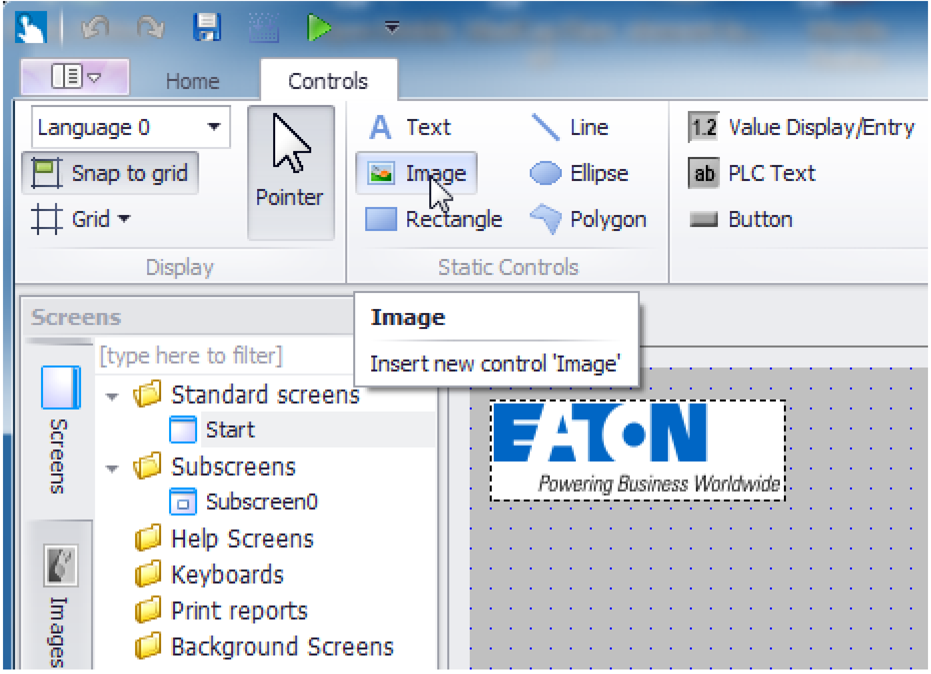
Figure 118: Adding an image by dragging&dropping it

Substep 2: Adding a placeholder box for an image to the screen

If you do not want to use the image in its original size on the screen, use a placeholder box.

- Click on the "Start" screen on the workspace.
- Click on the "Controls" tab.
- In the Controls group, click on 🖼 Image.

Controls\ image



Image

Insert new control 'Image'

Figure 119: Inserting a box for an image into the "Start" screen

- ▶ Click on the location you want on the screen, e.g., on the top left.
- ▶ While holding the mouse button down, drag the mouse to draw the box.

Controls\ image

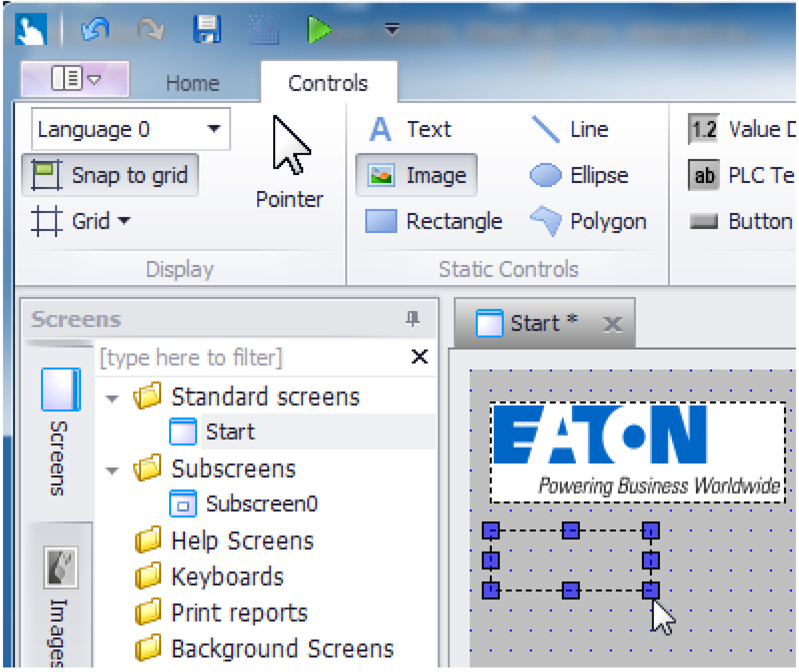
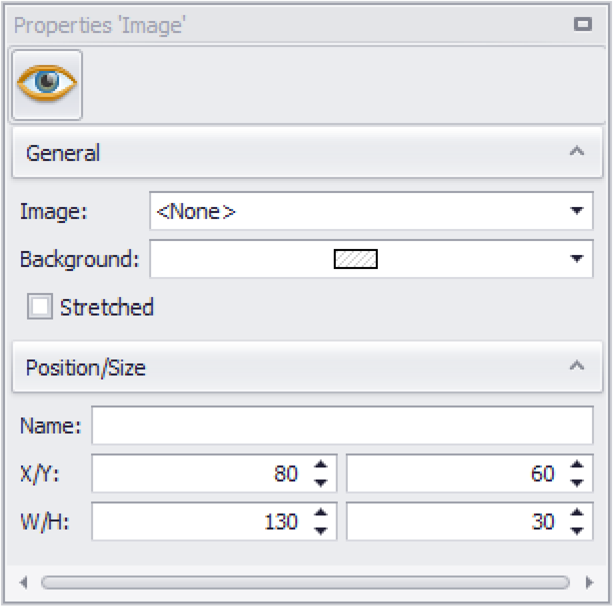


Figure 120: Drawing a box for an image on the "Start" screen

The Properties 'Image' pane will be filled out with the properties for the empty box, and the box's coordinates and pixel sizes will be shown and change automatically if you move or resize the box.

- Enter the following position information (X/Y coordinates) into the Properties 'Image' pane: 0/60.
- Enter the following size information (width/height) into the Properties 'Image' pane: 82/26.

Properties 'Image'



Properties 'Image'

General

Image: <None>

Background: ▨

□ Stretched

Position/Size

Name:

X/Y: 80 60

W/H: 130 30

Figure 121: Properties 'Image' for the example above

Substep 3. Inserting the image into the box using a specific size

- Use the Image drop-down menu to select your image.
- Click on the box you drew for an image.

The image will be inserted into the frame.

For our example

Properties 'Image'

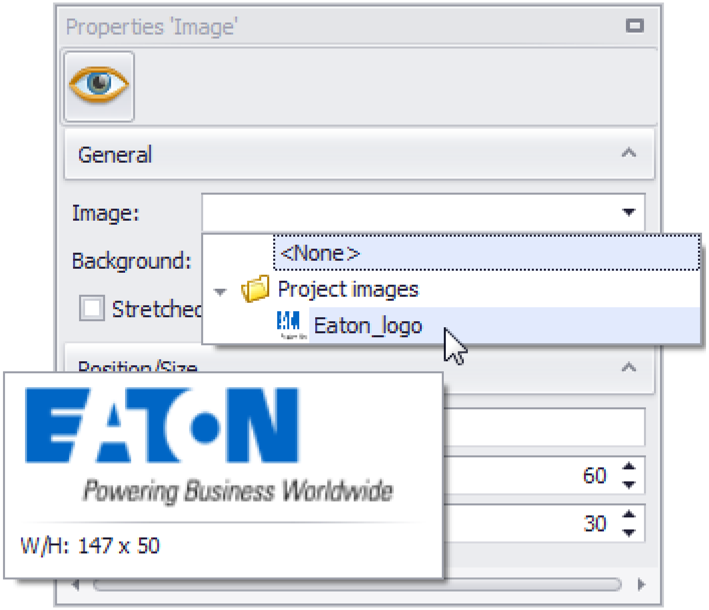


Figure 122: Properties 'Image' for the example above

Position/Size

The box for an image was drawn at the following position on the screen and with the following size:

X/Y	0/0 Upper left corner
W/H	164 pixels wide / 51 pixels tall

The basic properties for the image box under General indicate the following:

- The image that will be shown on the screen
- That the box's background is transparent
- That the selected image will be stretched to fit the box
- Select Stretch: Uniform

Start screen

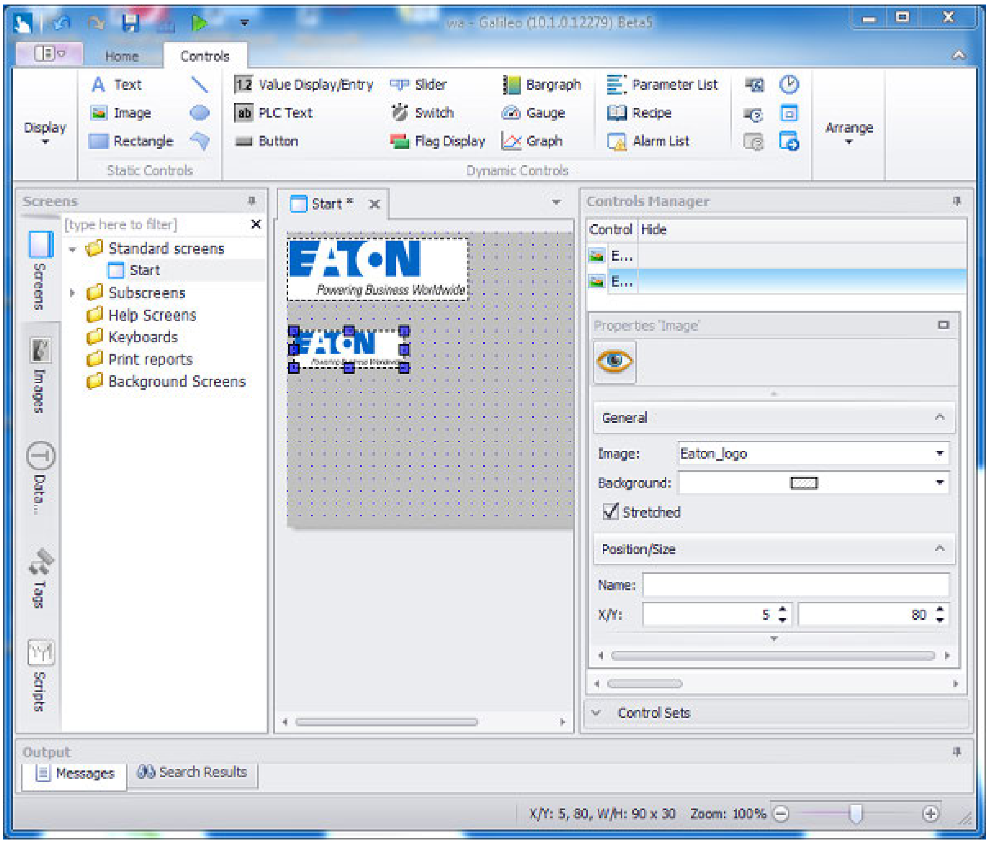


Image: Eaton_logo
Background:
☑ Stretched
Name:
X/Y: 5 80
X/Y: 5, 80, W/H: 90 x 30 Zoom: 100%

Figure 123: Example: Image inserted into the Start screen

The screenshot above shows the image that was dragged&dropped into the screen with its original size and the image inserted into a placeholder box with a specific size.

Continue to the following step with your first project:

→ "Step 6: Working with a user control", page 181

1.6.4.9 Step 6: Working with a user control

User controls are used in order to have the same content available on multiple screens.
Typical applications include a company logo, the date and time, etc., as well as areas used to show machine controller messages.
These controls are also used to implement touch controls for PCT devices.

UserControls are ideal for grouping together objects that are used at several places in a single project, e.g., a header bar, a status bar, and recurring object groups.

→ The advantage of using a user control is that later changes only need to be made to a single place in the project, i.e., to the user control.
The modification will then be automatically applied to all instances of the user control in other screens, in other user controls, and in containers.

To use a UserControl on another screen, follow the substeps below.

Step 1: Creating a user control

▶ Go to the project tree.

▶ Click on the Screens tab.

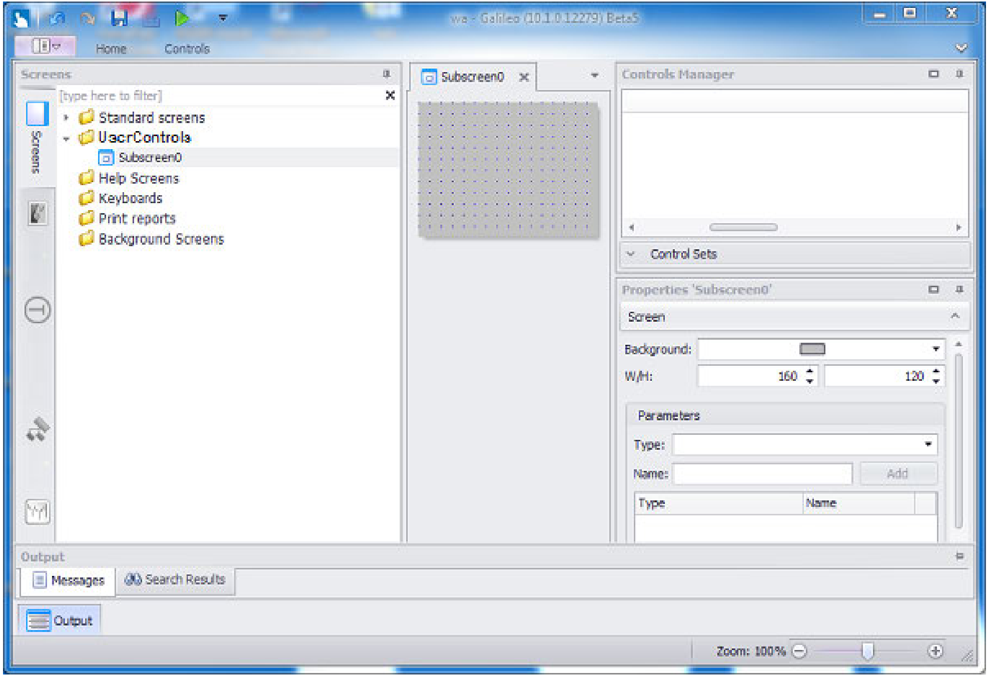
▶ Right-click on the UserControls folder to open the corresponding context menu

▶ Click on Add new UserControl

A User Control named **UserControl** will be created and opened on the workspace.

▶ Rename the control (**<Subscreen0>**, for example)

Project Tree\Screens



Background:
W/H: 160 120

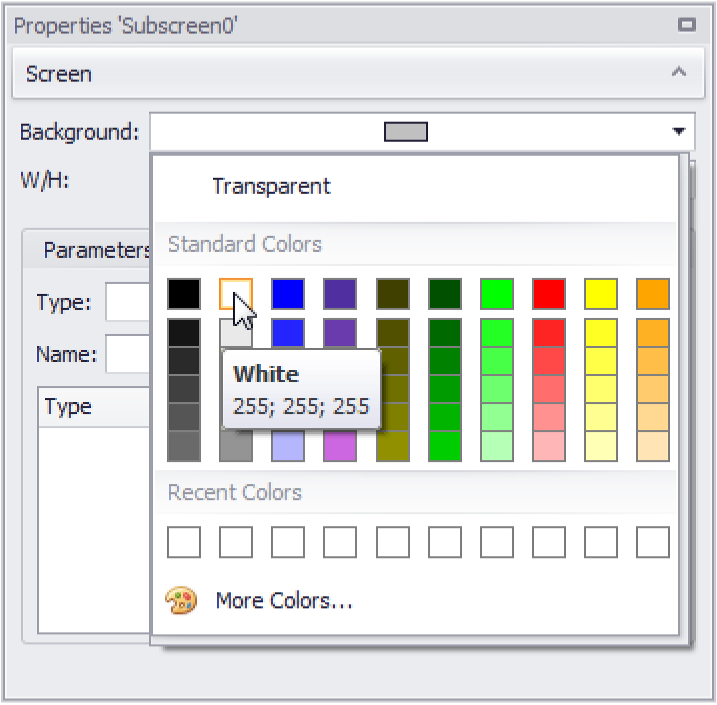
Figure 124: Creating a new user control

Step 2: Setting the background color and size for the user control

The Properties <Subscreen0> pane will be filled out with the properties for the subscreen.

▶ Select the following as the background color: White [RGB 255;255;255].

Properties pane for the UserControl



Properties 'Subscreen0'

Screen

Background:

W/H:

Transparent

Standard Colors

White
255; 255; 255

Recent Colors

More Colors…

Figure 125: The Properties pane for the **<Subscreen0>** user control

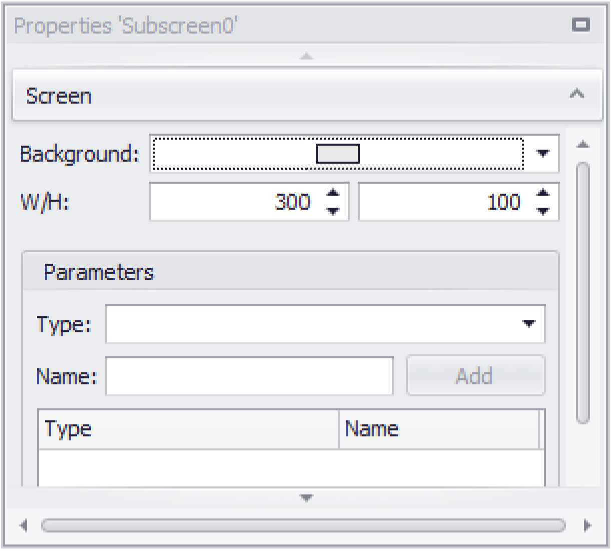
The screen's background will be shown in the selected color.

In this example, the color and size for the user control are set just for illustration purposes – they can always be changed later.

Adjust the size (width/height) of the user control as required for the standard screen.

▶ Enter a width of 320 pixels and a height of 80 pixels.

Properties pane for the UserControl



Properties 'Subscreen0'

Screen

Background:

W/H: 300 100

Parameters

Type:

Name:

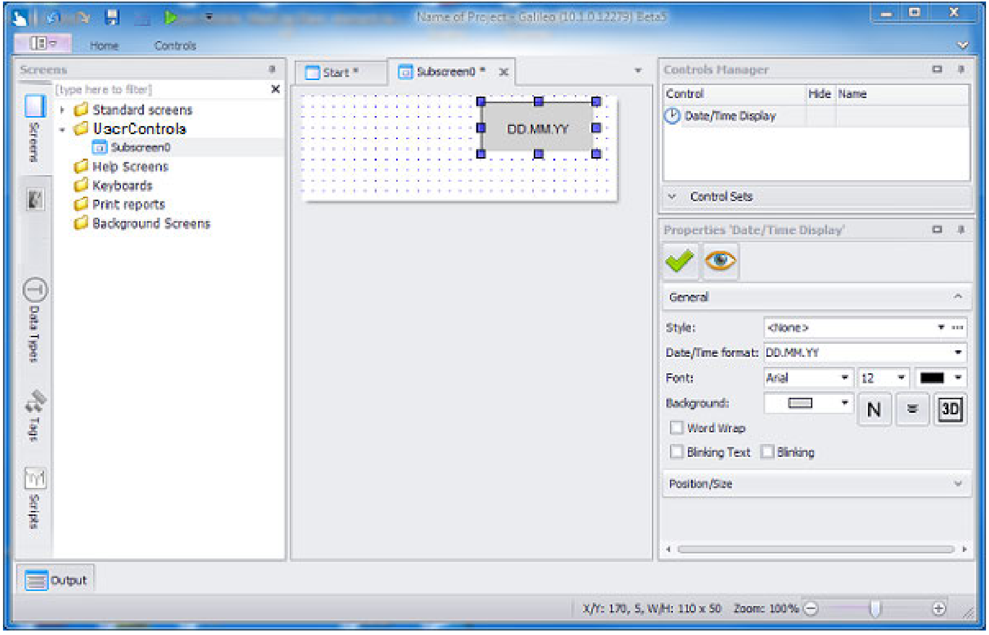
Type	Name

Figure 126: UserControl properties

Step 3: Adding a date-time display to the user control (as an example)

- Click on the user control on the workspace
- Click on the Controls tab.
- In the Controls group, click on 🕒 Date/Time Display.
- Draw a placeholder box on the UserControl.

Project Tree\Screens



Style: <None>
Date/Time format: DD.MM.YY
Font: Arial 12
Background:
□ Word Wrap
□ Blinking Text □ Blinking

Figure 127: Date/time display added

The Date/Time Display object will be added to the Controls Manager and

the Properties 'Date/Time Display' pane will show the properties for the object you added.

- Change the properties for the Date/Time Display object in the Properties pane as necessary.
- Click on the 💾 icon in order to save the **<Subscreen0>** user control.

Step 4: Inserting the user control into another screen

- Go to the project tree.
- Click on the Screens tab.
- Open the Start screen on the workspace.
- Click on the UserControl **<Subscreen0>** subscreen in the project tree.

Project Tree\Screens

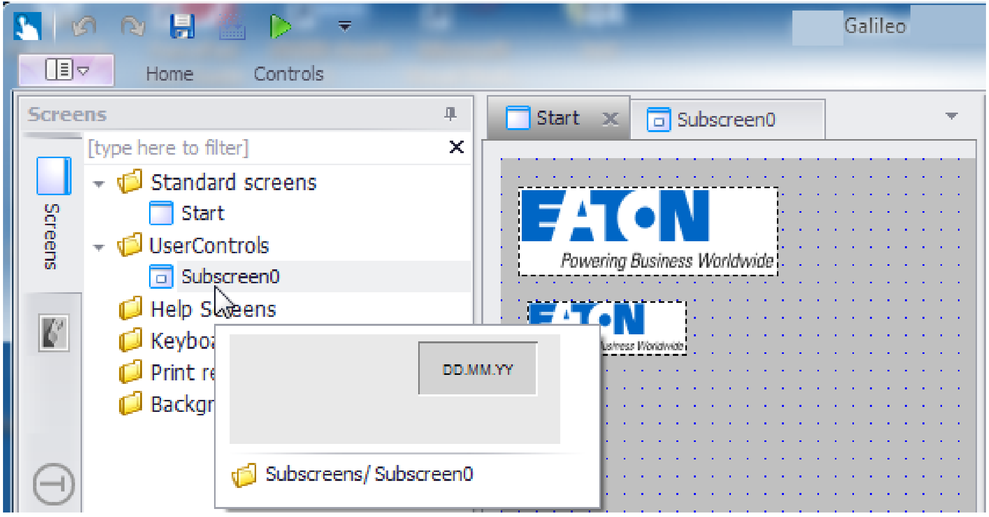


Figure 128: Selecting the user control that will be inserted

→ You can drag & drop
 the UserControl onto the screen in the position you want and adjust it as necessary.

The UserControl will be shown as an object in the Controls Manager pane.

Project Tree\Screens

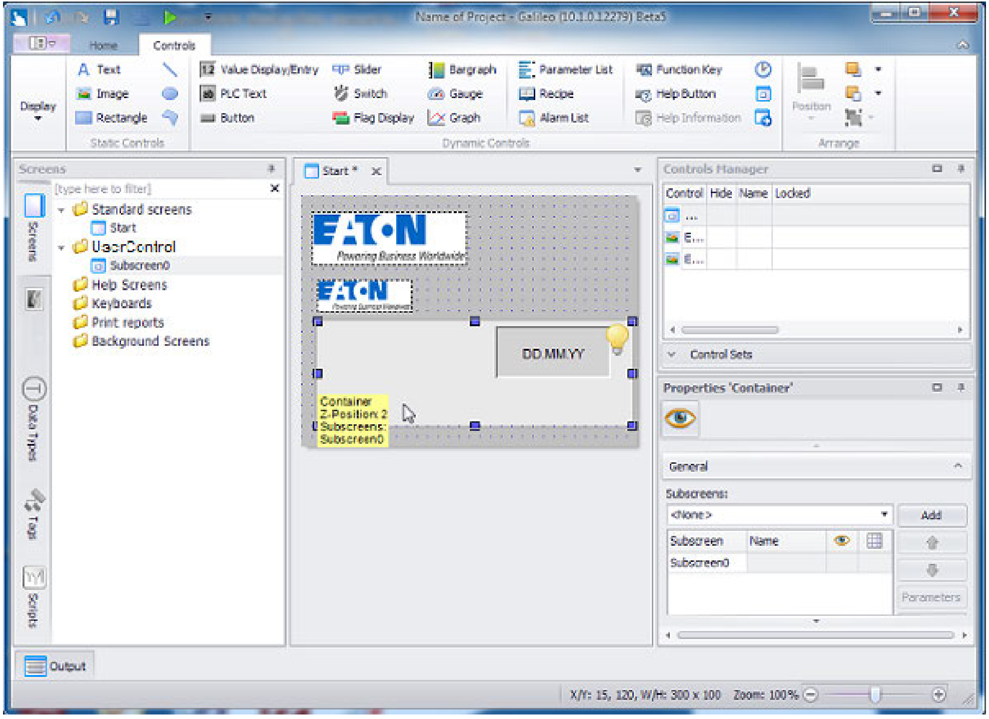
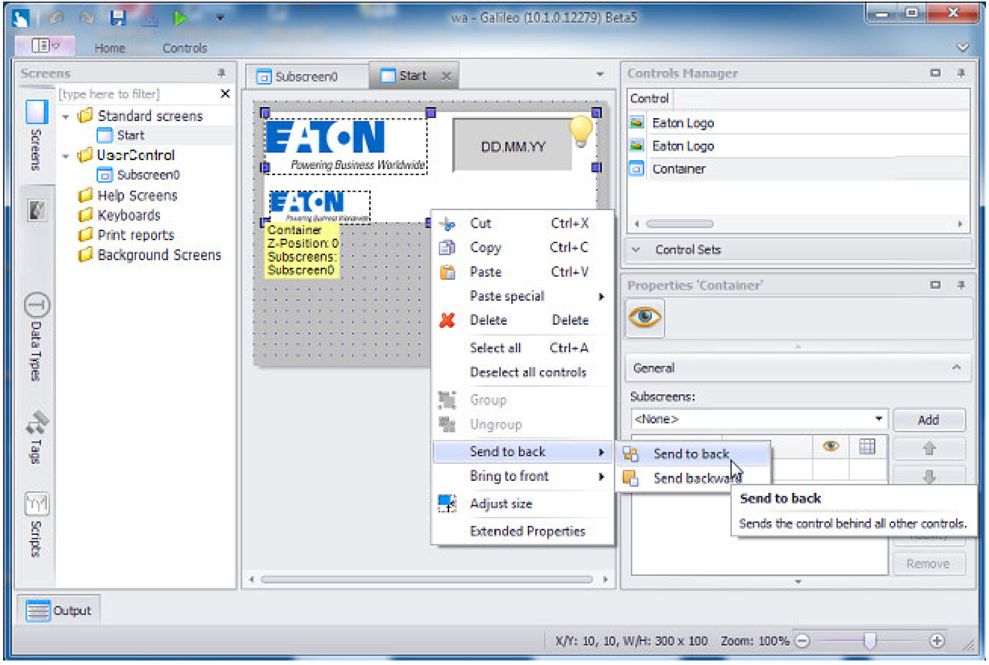


Figure 129: User control inserted

💡	Quick access menu as context menu
✥	Used to move the entire object

Additional user control adjustments



Send to back
Sends the control behind all other controls.

Figure 130: Context menu for user control on the screen

Continue to the following step with your first project:

→ "Step 7: Adding a button with a label", page 186

1.6.4.10 Step 7: Adding a button with a label

A distinction needs to be drawn between the following when using buttons:

- The actual button, for which a function needs to be defined
- A label or image used to ensure that users will be able to recognize what the button is meant for

To add a button to a screen, follow the substeps below.

→ "Substep 1: Creating a button", page 187
→ "Substep 2: Adding a label to the button", page 188
→ "Substep 3: Adding a function to the button", page 189

Substep 1: Creating a button

- Go to the project tree.
- Click on the Screens tab.
- Open the subscreen you created previously (Subscreen0)

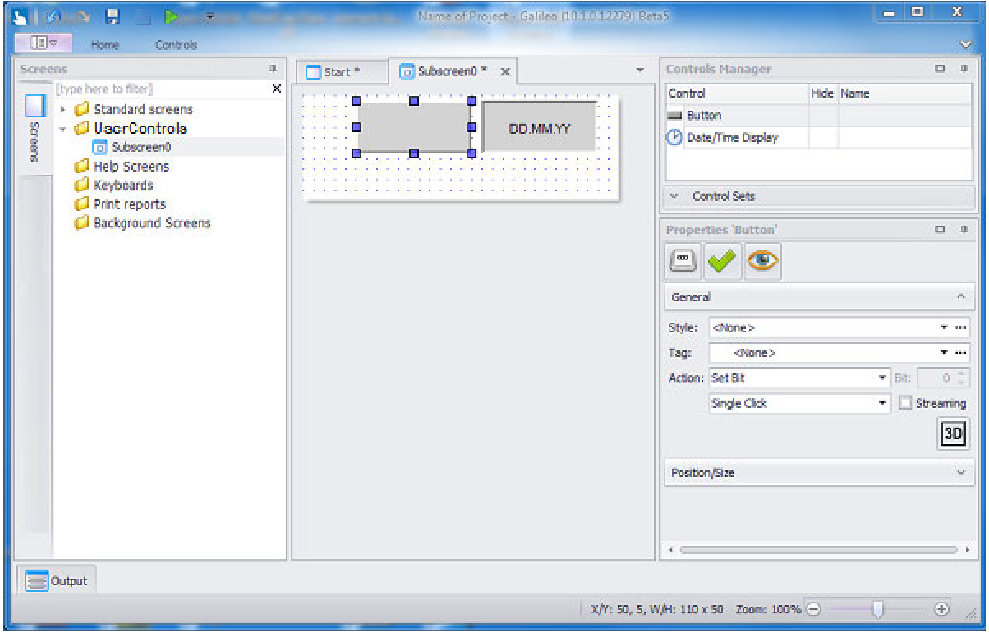
A user control named **<Subscreen0>** will be created and opened on the workspace.

→ If you have been following the previous steps in chronological order, the screen will already be open.

Adding a button object and creating the text for the corresponding label

- Click on *Controls\Button*.
- Position the object
  Move the cursor to the screen and draw the area for the visualization object.

Adding a dynamic button object to the user control



Style: <None>
Tag: <None>
Action: Set Bit Bit: 0
Single Click

Figure 131: Inserted button

- Position the Button object next to the Date/Time Display object.
- In the Properties 'Button' pane, click on the □ button and select the 3D 3D frame option so that the button will look like an actual physical button.

Substep 2: Adding a label to the button

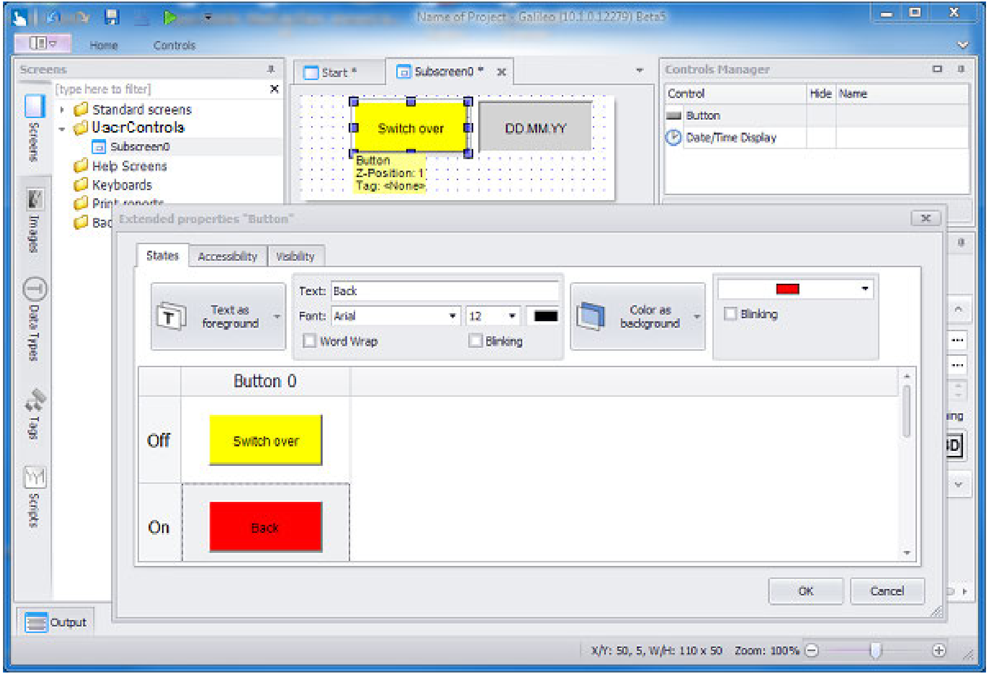
There are various options available for visually designing buttons

Creating a text for use as a label on the button

Make sure to label the various objects so that the touch panel operator will later on know what each button does.

- Double-click on the Button object.
- Under the States tab, select the Off state.
- Type Change into the <Text> field.
- Now select a background color.
- Under the States tab, select the On state.
- Type Back into the <Text> field.
- Now select a background color.

***<Subscreen0>** with inserted button object*



Text: Back
Font: Arial 12
☐ Word Wrap ☐ Blinking

	Button 0
Off	Switch over
On	Back

Figure 132: Extended button properties for labeling

- Click on OK.

You have now configured the actual button details for the button object.

Substep 3: Adding a function to the button

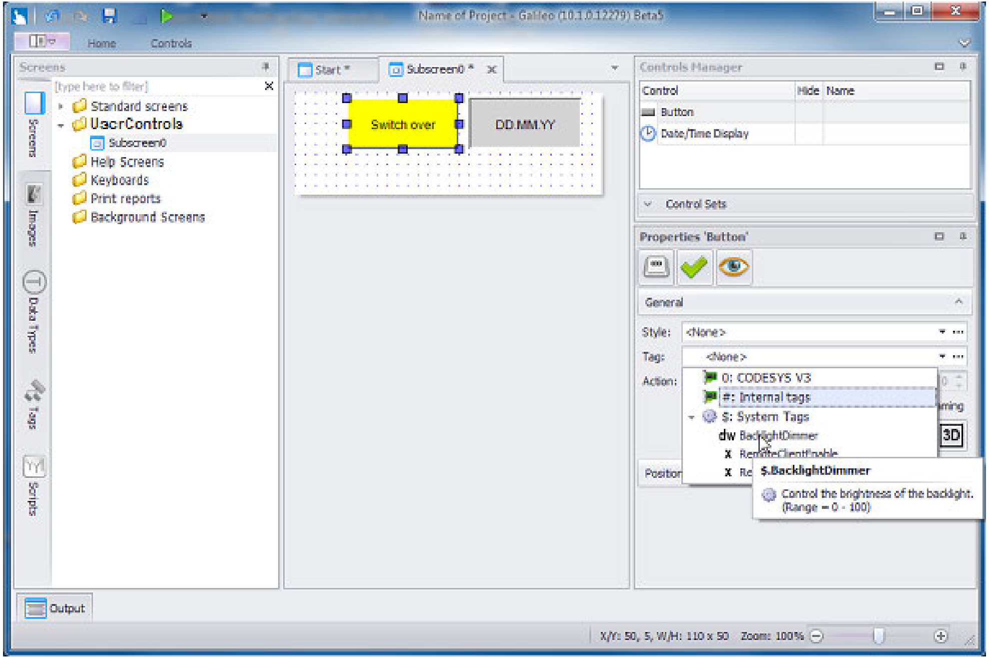
To have the button trigger a specific function, you need to be familiar with the use of tags.

For purposes of this example, we will be using a function for adjusting the display brightness.

▶ Click on the Change button.

▶ In the General section of the Properties 'Button' pane, select$ System Tags\ dw BacklightDimmer in the Tag: field.

User control with added object button



$.BacklightDimmer
Control the brightness of the backlight. (Range = 0 - 100)

Figure 133: Selecting a tag in the Properties 'Button' pane

Continue to the following step with your first project:

→ "Step 8: Adding screen changes", page 189

1.6.4.11 Step 8: Adding screen changes

The screen change object can be added to any type of screen.

If applicable, you will also need to add a screen change to the screen to which you are switching so that all screens will be interconnected and accessible.

Screen changes always define the transition from the screen on which they are to the screen that should be reached, i.e., the target screen.

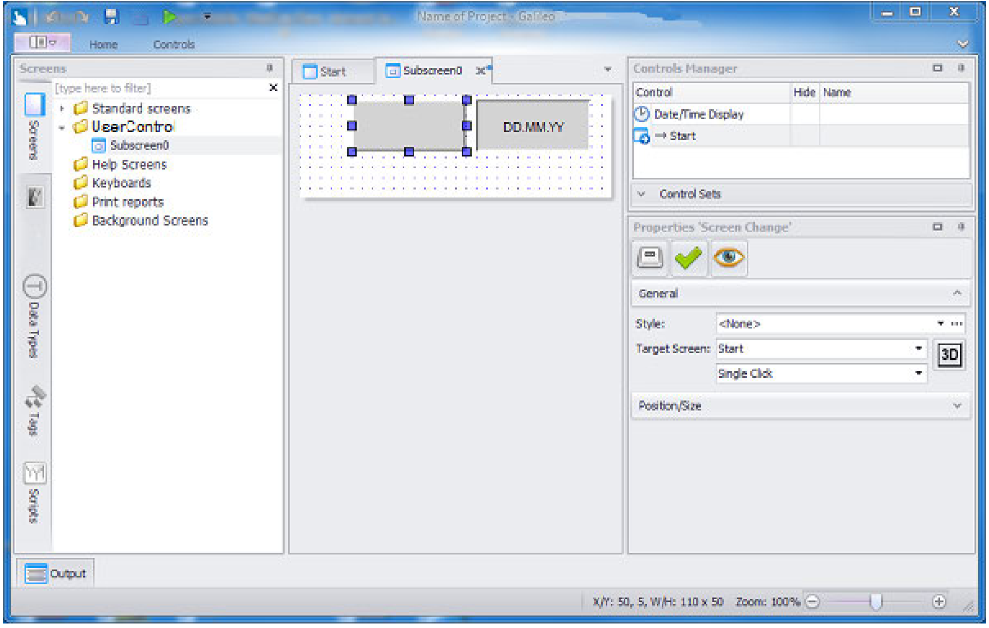
▶ Open the "from" screen (e.g., <Subscreen0> from which you want to reach a different screen (e.g., the Start screen).

▶ Click on *Controls\Screen Change*.

▶ Position the object
Move the cursor to the screen and draw the area for the visualization object.

- Make sure that the visualization object is selected on the screen.
- In the Properties 'Screen Change' pane, select the "to" screen in the Target Screen field by clicking on the drop-down menu arrow (the "to" screen is the screen that should be reached, e.g., Start).

Screen change properties in user control



Control	Hide	Name
Date/Time Display		
→ Start		

Style: <None>
Target Screen: Start
Single Click

Figure 134: Screen change added to user control

You can configure texts, images, and backgrounds for the screen change the same way you have already done so for other objects in the previous steps in this chapter.

- Double-click on the screen change in order to access its extended properties.

Using an icon for the screen change

In order to ensure that device users will know what the screen change button does, you can add an icon to it.

- Make sure that you have already imported the icon you want into the project tree.
- Double-click on the screen change and open the States tab.
- Click on the Off state.
- Instead of Text as foreground, select the Use an icon as foreground option.
- Select the icon you want from the Project images folder.
- Click on OK.

Icon instead of label

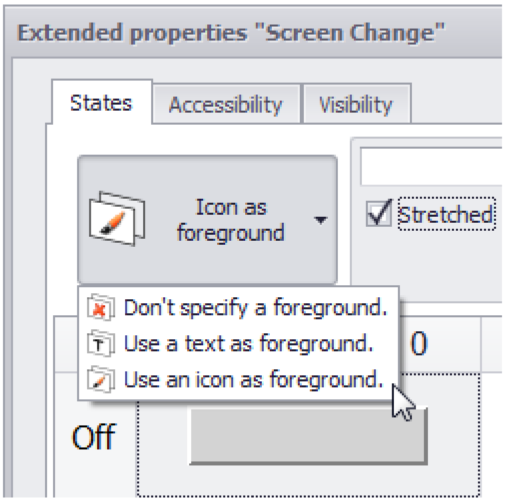

Extended properties "Screen Change"
States
Accessibility
Visibility
Icon as foreground
☑ Stretched
Don't specify a foreground.
Use a text as foreground.
Use an icon as foreground.
0
Off

Figure 135: Screen Change with Icon

Continue to the following step with your first project:

→ "Step 9: Testing the project with the simulator", page 191

1.6.4.12 Step 9: Testing the project with the simulator

To test your project, there are two substeps that you need to carry out: building and deploying it.

Build – shortcut key: F5

You can compile your project at any time (i.e., generate the data for the GRS runtime system) while working on the various screens and objects.

During compilation, the objects will be validated and, if applicable, the Output pane will show any relevant messages and/or warnings.

→ If there are any errors or warnings, you will need to look into them and make the necessary corrections in your project before the program can successfully generate the data for the GRS runtime system.

Build and Deploy – shortcut key: F6

This substep includes both generating the data for the GRS runtime system and transferring the data to the configured and selected destination.

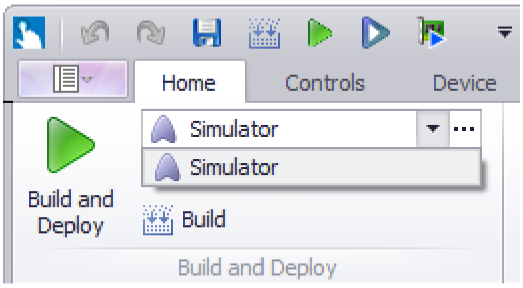
For your project, you will be using the Inspector Galileo Simulator.

The Galileo Simulator can be used to simulate the project:

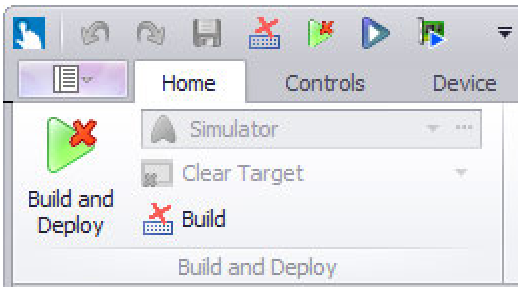
- On the computer on which GALILEO is installed
- Without the need for a touch panel
- Without a data connection having to be configured

To do this, set the Galileo Simulator as the destination first. Once you do so, your project will be shown in the Galileo Simulator on your computer after being compiled.

- Open the Home tab and select the Simulator option in the drop-down menu for Build and Deploy.



- Run the test by clicking on Build and Deploy



GALILEO will be locked as long as the process is running.

Once the project is successfully compiled, the Galileo Simulator Inspector will start automatically and display the project the way it will look on the touch panel.

This completes your first project.

1.6.5 Notes on creating a Project

This section is intended to serve as a reminder of the things that you should keep in mind in order to ensure that your work with the Automatization Software GALILEO program will be as efficient and effective as possible.

For information on migrating existing GALILEO 8.1 projects to GALILEO, as well as on supported protocols, please refer to the Galileo Migration Guide PDF document that was stored in the selected folder on your computer when GALILEO was installed.

Galileo Migration Guide.pdf
C:\Program Files\Eaton\Galileo 10\doc (default installation path)

The Galileo Migration Guide provides help when porting projects from GALILEO 8.1 to GALILEO. GALILEO will automatically perform and configure many of the required conversions and applications during the import process. However, you may need to make certain changes by hand in certain cases. The Galileo Migration Guide is designed to provide assistance in these cases.

1.6.5.1 Things you should keep in mind

Follow the rules and recommendations described below to ensure optimum and speedy project design.

Preparation

Before starting to create a project, you should plan how it will be set up and what kind of screen structure you want to use for it.

Multiple languages
Create the project languages you need before creating the screens; please refer to → "Window Text Editor dialog box - Text list excerpt", page 643

Make a rough draft of the screens and controls you will be using. Check your plans for errors, inconsistencies, and unnecessary duplicates.
Prepare all the media you will be needing for the project (images, graphs, pictures, and/or videos). On one hand, by creating the ones that need to be created with a graphics editor and, on the other, by importing the media into the project tree's Media tab pane. For an example showing how to do this, please refer to: → section "Substep 1: Adding an image (medium) to the project tree", page 174.

All the objects used in the Automatization Software GALILEO program should be organized in the project tree before or while working the project and then used on the various screens.

Because of this, you should consider the following questions before creating a project:

- How should the project be named? Where should it be stored?
  - Name and location.-
- Do you want the project to have multiple languages?
  - Add project languages -
- Which types of screens will be needed? How should they be named?
  - Standard screens, help screens, user controls, ...-
- How will the screens be connected to each other?
  - Which screens are accessed from the start screen? -
- What should be displayed on the screens?
  - Buttons, images, values, indicators, ...-
- Which values should be shown? How should they be named?
  - Button labels, indicators, variables-
- Which PLC will be connected? With which communication method?
- The locations for the data in the PLC
- Will you be needing the user logger?

When configuring tags

Data blocks can also be sent between the HMI and PLC. The advantage of this is that communication will be more efficient than when transmitting each byte and bit individually.

In this case, make sure that the data packed into structures is of the same type or belongs together and that the structures are assigned addresses directly.

Limitations due to Windows CE

Devices with Windows CE only support filenames with ASCII characters.
This applies to all functions for which a file needs to be saved locally.

Color palettes

Make sure to use the same color palette for the entire project. This will ensure that the selected color settings will not change when you load a new image.

Background Image

In order to achieve maximum performance when switching between screens and when refreshing a screen, the following need to be observed when using a background image:

- The background image should not have any visibility attributes
- Do not use tags to define the background image's position or size

Size and position of dynamic objects

Displaying dynamic objects takes a lot of processing power. In certain cases, performance may suffer when switching between screens or when objects are being drawn.

Please note:

- As far as possible, make sure not to superimpose dynamic objects on top of each other, and avoid superimposing other objects on top of dynamic objects as well.
- Make sure not to superimpose dynamic objects on list objects (alarm lists, parameter lists, recipes) or camera objects and vice versa.

Layout

Use a consistent, recurring layout for your projects. Make sure to use this layout for your first screen, as this screen will be used as a template when you create new standard screens.

For company logos, etc., design an object of your own in User Controls that can be placed as often as necessary on (standard) screens. Using containers can also come in handy.

Time and effort required in order to create and maintain content if changes are made to screen contents

Reusing and nesting individual user controls will dramatically reduce the time and effort required in order to create and maintain content when changes are made to screen contents, as the change to the user control source will be automatically applied everywhere where the user control is used.

1.6.5.2 Creating a project – A finished project in only a few steps

GALILEO is designed to enable you to create visualization projects with little effort and without the need for any special previous experience.
It comes with a large variety of ready-to-use functions and basic objects that make it easy to configure and test devices according to your specific needs.

Following are the general steps required in order to create a new project.
Of course, the individual steps can be repeated as many times as necessary – the list is simply meant as a reference.
→ These steps need to be completed for every project.

1. Create the project and select a panel model

The New Project... Wizard will guide you through the process.

2. Select one or more communication methods

You can use up to eight communication methods simultaneously. Data can be transmitted from PLC to PLC using the panel.

Configure the communication methods you select and review the settings for your project.

3. Create screens

You can set up full screens, subscreens, diagnostic screens, and custom input screens.
If you installed the sample projects when installing the software, you can refer to them in order to learn more about the various GALILEO features. Simply click on Recent.

4. Create tags or import them from CODESYS

An input dialog box specific to the communication method chosen will be available for defining tags. Tags and data structures from XSOFT-CODESYS or other CODESYS-based controllers can be conveniently imported and synchronized when PLC project changes are made.

5. Place an object on the screen

All visualization objects can be dragged and dropped onto the corresponding screens, after which they can be customized as necessary for the specific application by configuring their comprehensive properties.

6. Simulate the project on the computer used to create it

The project can be compiled and simulated directly on your development computer at any time. When you do this, detailed error messages and warnings will let you know of any problems with the project.

7. Load the project onto the panel

Once the project is successfully compiled, it can be transferred to the panel in real time during ongoing operation with a simple click.

1.6.5.3 Project Configuration

Window Project Settings

This dialog box can be used to configure the settings that apply to the entire project. To open this dialog box, click onHome/Project Configuration/Settings.

The settings in this dialog box are project-specific and need to be configured as necessary for every single new project.
If you copy a project, i.e., if you create duplicates of it as a basis for other projects, the settings from the original project will automatically carry over.

The Project Settings dialog box is subdivided into various tabs.

- → "Runtime", page 196
- → "User Logger", page 201
- System Keyboards
- → "System Messages", page 209
- → "Screen View", page 212
- → "Frame", page 216
- → "Advanced Settings", page 218

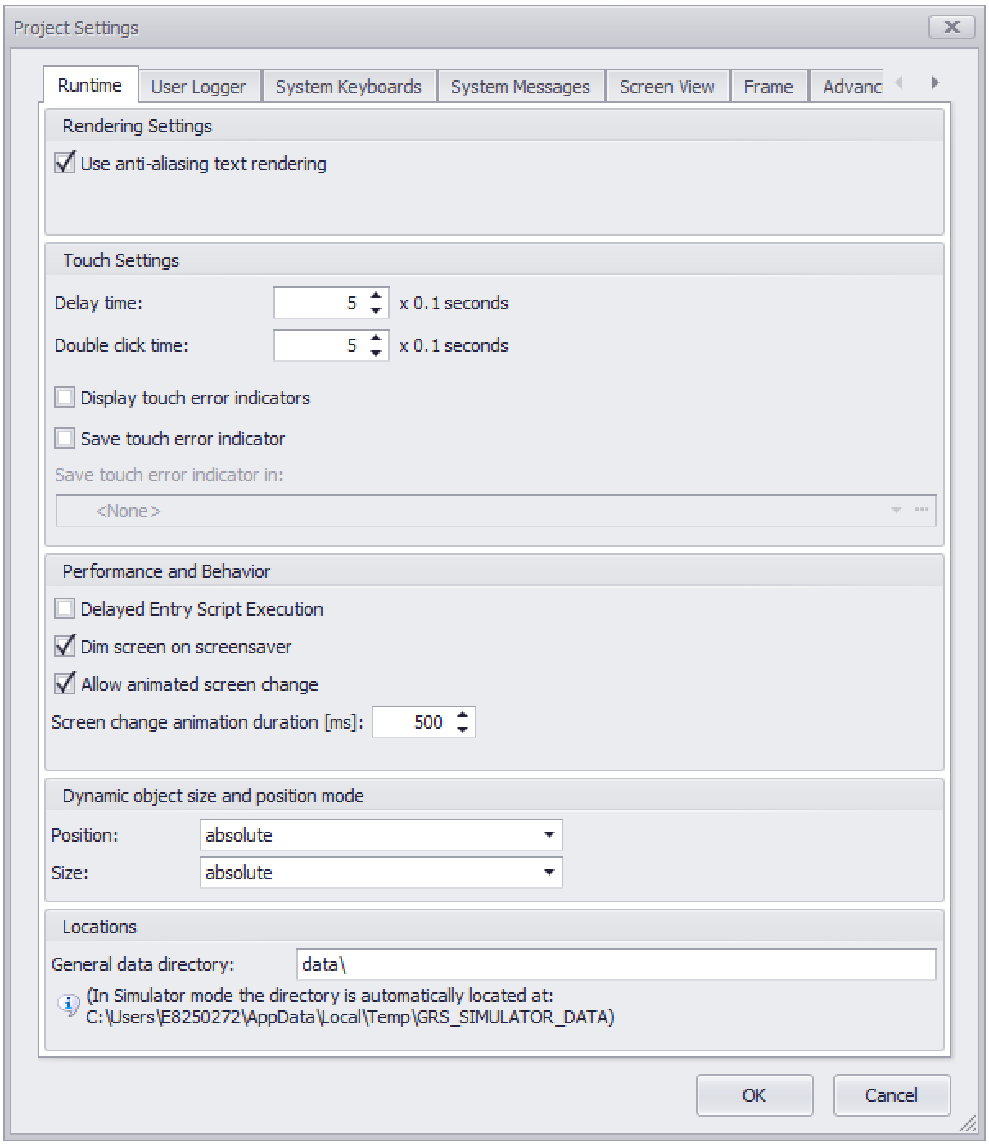
→ Further, general settings for visualization and communication can be found on the home tab Device
described in:
→ section " Web visualization", page 1070
→ section "Select Communication", page 581
→ section "Data communication principles", page 580

Runtime

This tab can be used to configure basic settings for the project if any changes are necessary.

Some of the settings in this tab will have already been configured when creating the project with the New Project ... Wizard.

Home\Project Configuration\Settings tab



Rendering Settings

☑ Use anti-aliasing text rendering

Touch Settings

Delay time: 5 x 0.1 seconds

Double click time: 5 x 0.1 seconds

□ Display touch error indicators

□ Save touch error indicator

Save touch error indicator in:

<None>

Performance and Behavior

□ Delayed Entry Script Execution

☑ Dim screen on screensaver

☑ Allow animated screen change

Screen change animation duration [ms]: 500

Dynamic object size and position mode

Position: absolute

Size: absolute

Locations

General data directory: data\

(In Simulator mode the directory is automatically located at:
C:\Users\E8250272\AppData\Local\Temp\GRS_SIMULATOR_DATA)

Figure 136: *Project Settings/Runtime*

Text Style (option)

This property can only be enabled for devices belonging to the XV-1xx and XV-4xx product groups.

As soon as the checkbox is ☑ enabled, anti-aliasing will be disabled for text by default. If the GALILEO User Help uses thin fonts or fonts that contrast with the background, this option may improve the way the text looks on the device screen.

Touch Settings

Settings for the touch zone on the panel

Delay time: If a touch zone uses the "Delay" attribute, the touch zone will not become active until after it is continuously touched for a time equal to the factor specified here times x seconds.

Double click time: If a touch zone uses the "Double click" attribute, a function will not be executed until after the corresponding object is touched again within a time longer than the factor x entered here in seconds.

Display touch error indicators: If a touch is not interpreted correctly and this checkbox is enabled (☑), the touch error indicator icon will be displayed on the panel's top left and right corners.
Touch errors can be due to a dirty IR touch frame, an operating error, or a technical fault.
After approx. two seconds, the icon will be hidden again.

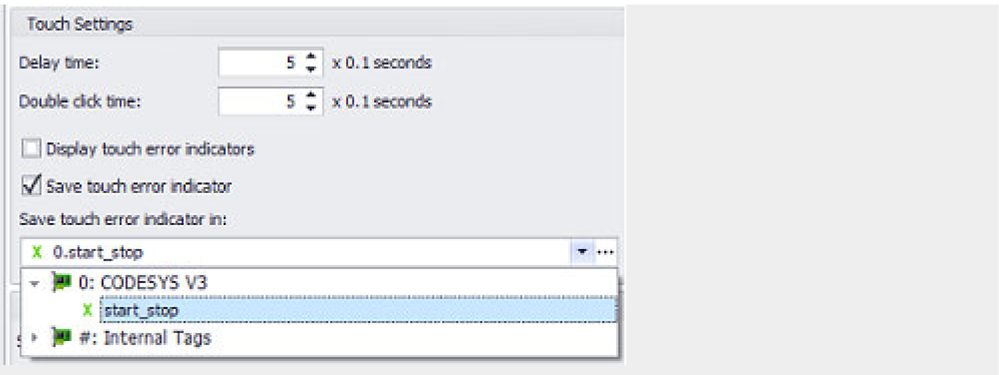
Save touch error indicator: If this checkbox is enabled (☑), the error state will be saved in a bit tag.
As soon as the touch error indicator becomes active, the selected bit tag will be set to TRUE (a value of 1). When the error is eliminated, the bit tag will be set to FALSE (value of 0).
If you do not want a tag to be used for this purpose, make sure to disable the Save touch error indicator checkbox.

Save touch error indicator in:

As soon as this checkbox is enabled (☑), you can select the tag that should be used to report touch error events, i.e., to save them.

The communication methods you have added to the project will be available together with the corresponding tags.

Example showing how to select a tag



Touch Settings

Delay time: 5 x 0.1 seconds

Double click time: 5 x 0.1 seconds

□ Display touch error indicators

☑ Save touch error indicator

Save touch error indicator in:

0.start_stop

Application script

The application entry script selected in the drop-down menu will be run exactly once when the application is started. More specifically, the script will be run before the start screen is displayed.
Without selection this function is skipped.

→ Certain function calls will be ignored, or may not even be allowed at all, in the application entry script.
The information on whether a function can be called in the application start script can be found as a table entry in the description of the respective function.

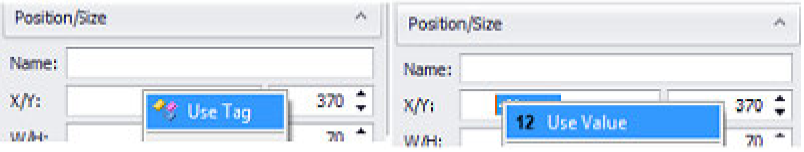
Performance and Behavior

☑ Delayed Entry Script Execution	Scripts configured as "entry scripts" on screens will normally be run before the corresponding screen is graphically rendered. When you use very large scripts, this can give the impression that a screen change is taking a long time. If you enable this option, the screen will be displayed first and the script will then be run. Make sure to only enable this option if the script does not modify any data that is relevant to how the current screen will be displayed.
☑Dim screen on screensaver	If a screen saver is set, it is hereby adopted for the project
☑Allow animated screen change	Enable/Disable animated screen changes in general. If not activated all the Screen Change controls and Menu Button controls will act non-animated.
Screen change animation duration [ms] (option)	Defines, how fast the animation will be visualized on animated screen changes (e.g. times span for a new screen on slide in). Min = 100ms, Max = 3s. This property is not available for devices of product groups XV-1xx and XV-4xx.

Dynamic object size and position mode

For certain types of objects, the position and size can be defined either as a fixed value or as a dynamic value.
The settings for interpreting dynamic object positions and sizes when tags are used can be set here.

'...'\Position/Size\X/Y properties

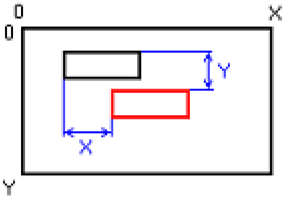


The context menu of the right mouse button allows you to switch absolute value 12 Use Value to Use Tag.

Position **Used to specify how tag values will be interpreted:**

relative To the configured object position

The tag value will represent the object's displacement in pixels relative to the object position defined in GALILEO.



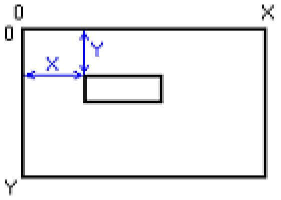
Example showing how an object would be moved

A tag value of 1 will move the object one pixel down or to the right. A negative tag value for the X-axis position will move the object to the left, while a negative tag value for the Y-axis position will move the object up.

absolute To the screen

The variable value represents the position of the top left object corner in pixels and uses the coordinates of the mask as the reference.

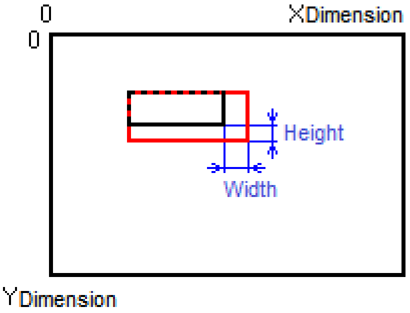
The top left corner of the mask is the zero point.



Size **Options: relative and absolute**

relative To the configured object position

The tag value will represent the degree to which the object should be enlarged or shrunk in pixels relative to the object size defined in GALILEO.

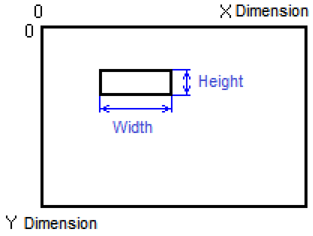


Example showing how an object would be enlarged

A tag value of 1 will increase the object's height/width by one pixel. If the tag value is negative, the object will be shrunk accordingly.

absolute to the screen

The variable value represents the height/width in pixels.



Objects must always be fully located within the screen.
If an object extends beyond a screen's edge, it will be drawn at the edge instead.

Locations

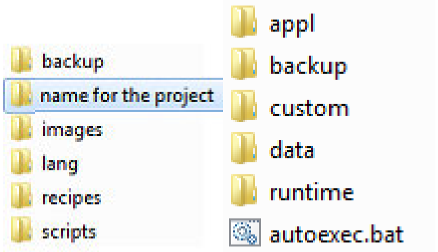
The data directory your enter here will be used as the destination for the panel when the project is built and deployed.

After a project is compiled (generated), the data structure for the touch panel will be generated.
This data structure will be stored in a folder called `<Project Name>` (Deploy).

General data directory: This field can be used to enter a data directory in which project-relevant data files should be stored, e.g., user management data.
By default, "data\" will be entered in this field.

A folder called <custom> will also be created and offered as the default storage location for all project-relevant files. This folder will be transferred to the panel together with the project,
meaning that consistently using this folder for storage and as an access location is an efficient way to solve the problem of relative and absolute file paths when opening or integrating third-party files.

Location of created project files for deployment to the panel



backup
name for the project
images
lang
recipes
scripts

appl
backup
custom
data
runtime
autoexec.bat

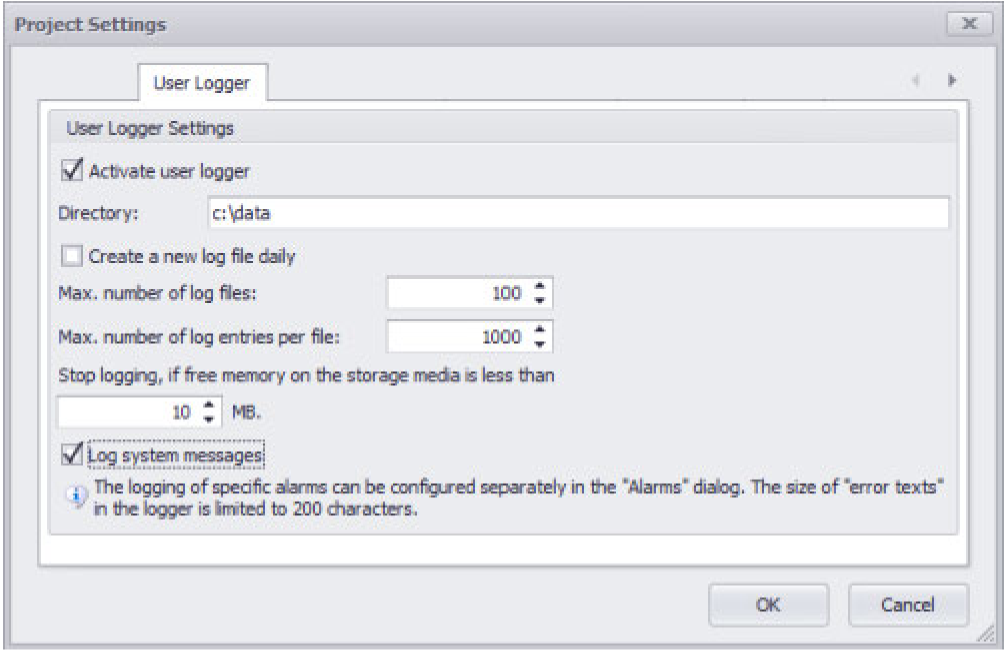
Figure 137: View in the computer's file browser

User Logger

If the user logger is activated, changes made by users to values and parameters will be logged in a log file

In addition, the user logger will log every instance of a function key being used.

Home\Project Configuration\Settings tab



User Logger Settings

☑ Activate user logger

Directory: c:\data

□ Create a new log file daily

Max. number of log files: 100

Max. number of log entries per file: 1000

Stop logging, if free memory on the storage media is less than 10 MB.

☑ Log system messages

The logging of specific alarms can be configured separately in the "Alarms" dialog. The size of "error texts" in the logger is limited to 200 characters.

Figure 138: *Project Settings:User Logger window*

User Logger Settings

□ The functions will be disabled

☑ Activate user logger

If you enable the user logger, you will have to enter a storage location.
Activates the user logger for the current project. This means that logging will being after startup.
You can use the Logger_ON-OFF function to start and stop the user logger at runtime.
Please note that starting the user logger at runtime will only be possible if the user logger has been activated previously by enabling the corresponding checkbox while setting up the project.

Directory

Used to enter the storage location where the log files will be stored.
This directory must not be located on the same data storage device as the operating system on which GRS is running.

→ Configuring the directory incorrectly will result in a system error message at startup and will prevent the user logger from being used,

☑ Create a new log file daily

Prevents log entries from being overwritten
This option ensures that a log file will not contain entries belonging to different days. When the date changes and a new log entry is generated, a new log file will be automatically generated even if the maximum number of log entries per file has not been reached yet.

Max. number of log entries per file

The maximum number of entries per log file. If the maximum number of log entries is reached, a new log file will be automatically generated and the filename index will be incremented by 1.
If the date has changed since the last time a log file was generated, the index will start again from 0.

Max. number of log files	The maximum number of log files that will be generated. If this number is reached, the oldest log file will be deleted before a new one is generated.
Stop logging, if free memory on the storage media is less than	The minimum amount of free space that there must be on the data storage device. The defined amount of memory must be available in order for a log entry to be generated. If the amount of free space available is less than the value specified here, a system error message will be shown every time an attempt is made to write to the log file. In order for the error message to stop appearing, enough free space must be made available or the user logger must be disabled.
☑ Log system messages	If this checkbox is enabled, GALILEO will also log system messages in the log file.

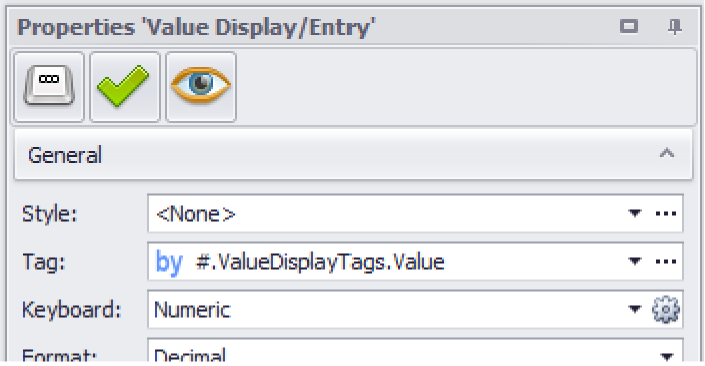
A detailed description can be found in: → section "User Logger", page 942

For a detailed description of how to create users, put together user groups, and specify access permissions, i.e., of how the user management feature works, please refer to the → "User Management", page 891 section

System Keyboards

GALILEO provides various system keyboards that enable device operators to enter numeric and alphanumeric values and texts on the device at runtime (e.g., numeric keyboard). These system keyboards will be used automatically without any special configuration if a keyboard is selected for an object (e.g., value display):

Object tab\1.2



Properties 'Value Display/Entry'

General

Style: <None>

Tag: #.ValueDisplayTags.Value

Keyboard: Numeric

Format: Decimal

Figure 139: *Properties pane as an example*

The display settings for all system keyboards can be configured in the System Keyboards tab in Project Configuration. All settings, with the except of positioning settings, will also apply to custom keyboards.

Home\Project Configuration\ Settings tab

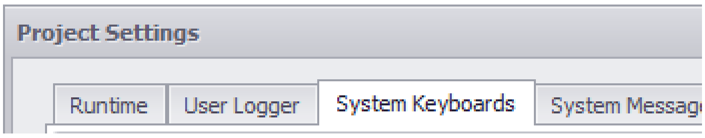


Figure 140: *Project Settings/System Keyboards window*

There are two different types of keyboard designs:

Classical

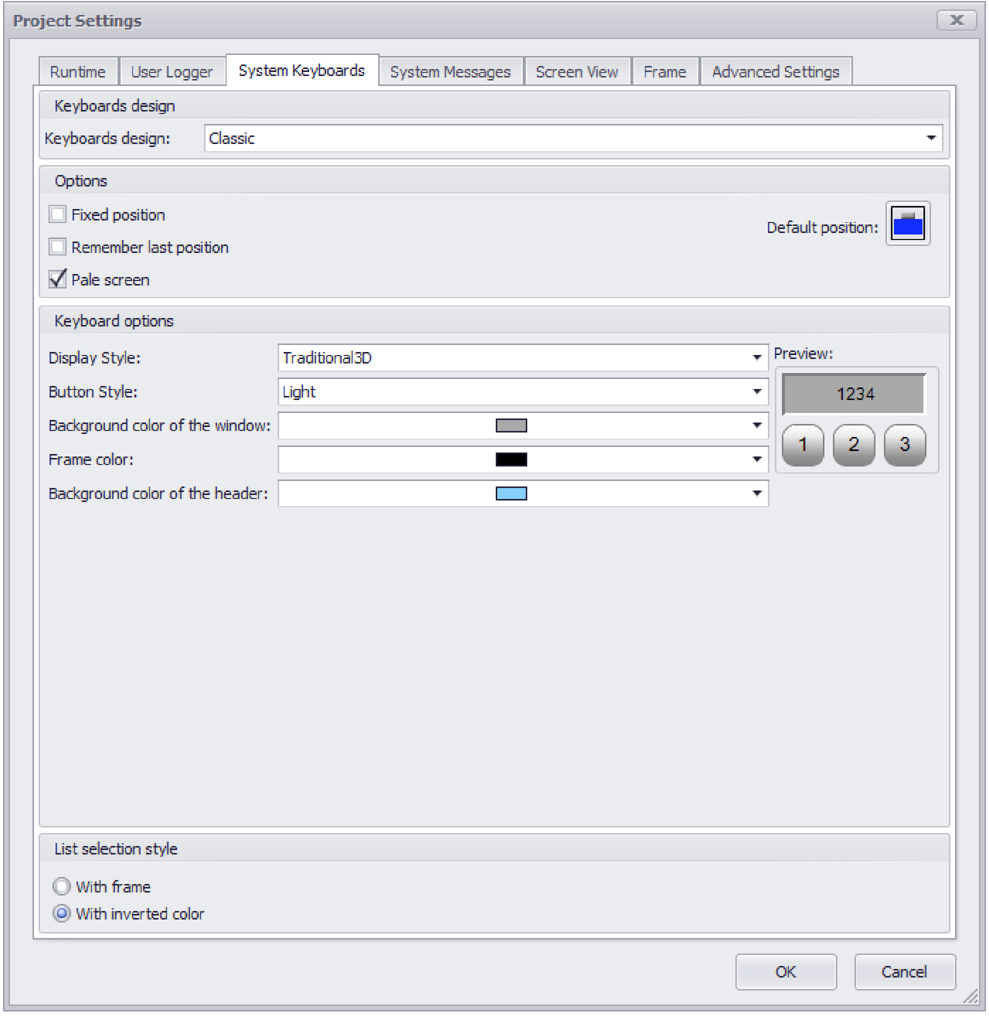
The classic keyboard layout and functionality
that has been used for all keyboards up to GALILEO Version 10.5

Extended

All options in the keyboard design drop-down menu with the exception of "Classic"
Available on GALILEO Version 10.5 and higher
Various designs and expanded functionality,
e.g., cursor positioning for entering values,
cursor function keys when using custom keyboards

Keyboard design Classic

Home\Project Configuration\ Settings tab



Project Settings

Tabs: Runtime | User Logger | System Keyboards | System Messages | Screen View | Frame | Advanced Settings

Keyboards design

Keyboards design: Classic

Options

□ Fixed position

□ Remember last position

☑ Pale screen

Default position:

Keyboard options

Display Style: Traditional3D

Button Style: Light

Background color of the window:

Frame color:

Background color of the header:

Preview:

List selection style

○ With frame

● With inverted color

Figure 141: *Project Settings/System Keyboards window*

Options

Fixed Position

If this option is enabled, an active keyboard cannot be moved around

Remember last position

If this option is enabled, the keyboard's position will be saved when closed. The next time the same keyboard is opened, it will be automatically moved to this saved position.

Gray out screen

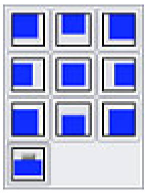
If this option is enabled, the view behind the keyboard will be dimmed.

Default position:

Used to specify the keyboard's default position relative to the object calling it. The alignments available for selection include "top left," "center center," "bottom center," etc. plus "docked."
If you select the "docked" option, the Galileo runtime system will decide where the keyboard can best be displayed. If the "Remember last position" option is enabled, this setting will only be applied the very first time the keyboard is opened.

▣ Default position

The keyboard will be positioned based on the option selected here. The default position on the screen can be defined by clicking inside the selection box as follows:



Top left ▣ , top center ▣ or top right ▣ Center left ▣ , center center ▣ or center right ▣ Bottom left ▣ , bottom center ▣ or bottom right ▣ or ▣ Docked on the button

Keyboard options

Display style: Used to select a style from the project for showing "display" objects

Button style: Used to select a style from the project for showing "button" objects

Background color of the window: The background color for the keyboard window on the screen.

Frame color: The color for the keyboard's frame

Background color of the header: A title can be assigned to certain system keyboards.
In the case of numeric keyboards, this color will also be the background color for the limit value display, e.g., blue.

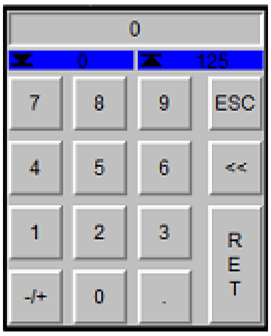


Figure 142: Numeric keyboard with display field <0>

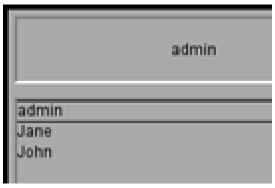
Preview:

Generates a preview for the currently configured settings.

List selection style

○ One of the two options needs to be selected

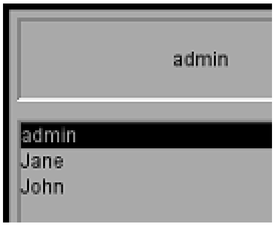
With frame The selected item will be surrounded with a box.



admin

admin
Jane
John

With inverted color The selected item will be highlighted by inverting the corresponding colors.



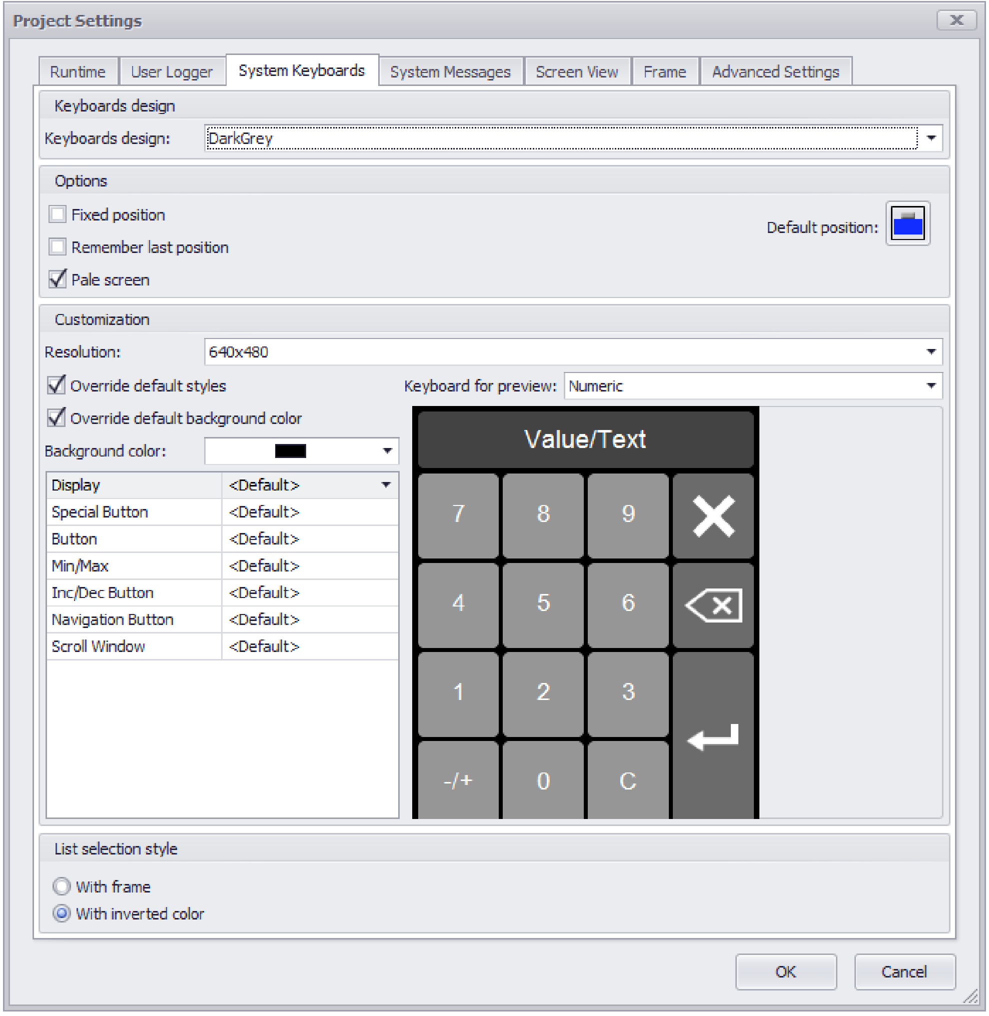
admin

admin
Jane
John

Keyboard design Exhanced

All options in the keyboard design drop-down menu with the exception of "Classic"

Home\Project Configuration\ Settings tab



Keyboards design

Keyboards design: DarkGrey

Options

□ Fixed position

□ Remember last position

☑ Pale screen

Default position:

Customization

Resolution: 640x480

☑ Override default styles

☑ Override default background color

Background color: ■

Keyboard for preview: Numeric

Display	<Default>
Special Button	<Default>
Button	<Default>
Min/Max	<Default>
Inc/Dec Button	<Default>
Navigation Button	<Default>
Scroll Window	<Default>

List selection style

○ With frame

● With inverted color

Figure 143: *Project Settings/System Keyboards window*

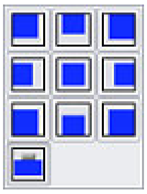
Options

Fixed Position	If this option is enabled, an active keyboard cannot be moved around
Remember last position	If this option is enabled, the keyboard's position will be saved when closed. The next time the same keyboard is opened, it will be automatically moved to this saved position.
Gray out screen	If this option is enabled, the view behind the keyboard will be dimmed.
Default position:	Used to specify the keyboard's default position relative to the object calling it. The alignments available for selection include "top left," "center center," "bottom center," etc. plus "docked." If you select the "docked" option, the Galileo runtime system will decide where the

keyboard can best be displayed. If the "Remember last position" option is enabled, this setting will only be applied the very first time the keyboard is opened.

Default position

The keyboard will be positioned based on the option selected here. The default position on the screen can be defined by clicking inside the selection box as follows:



Top left , top center or top right Center left , center center or center right Bottom left , bottom center or bottom right or Docked on the button

Customization

Resolution:	Can be used to select the resolution for the keyboard preview. (The settings do not refer to the resolution set up here)
Overwrite default style:	Each keyboard design defines its own default style. If you enable this option, it will be possible to overwrite styles for displays, buttons, etc.
Overwrite default background color:	Each keyboard design defines its own default background color. If you enable this option, it will be possible to overwrite the background color.
Back color:	Used to select the background color
Display/Default table	Configuration of elements
Keyboard for preview:	Used to select the keyboard type for the preview while taking the current configuration into account.

System Messages

Basic settings for project-related system messages.

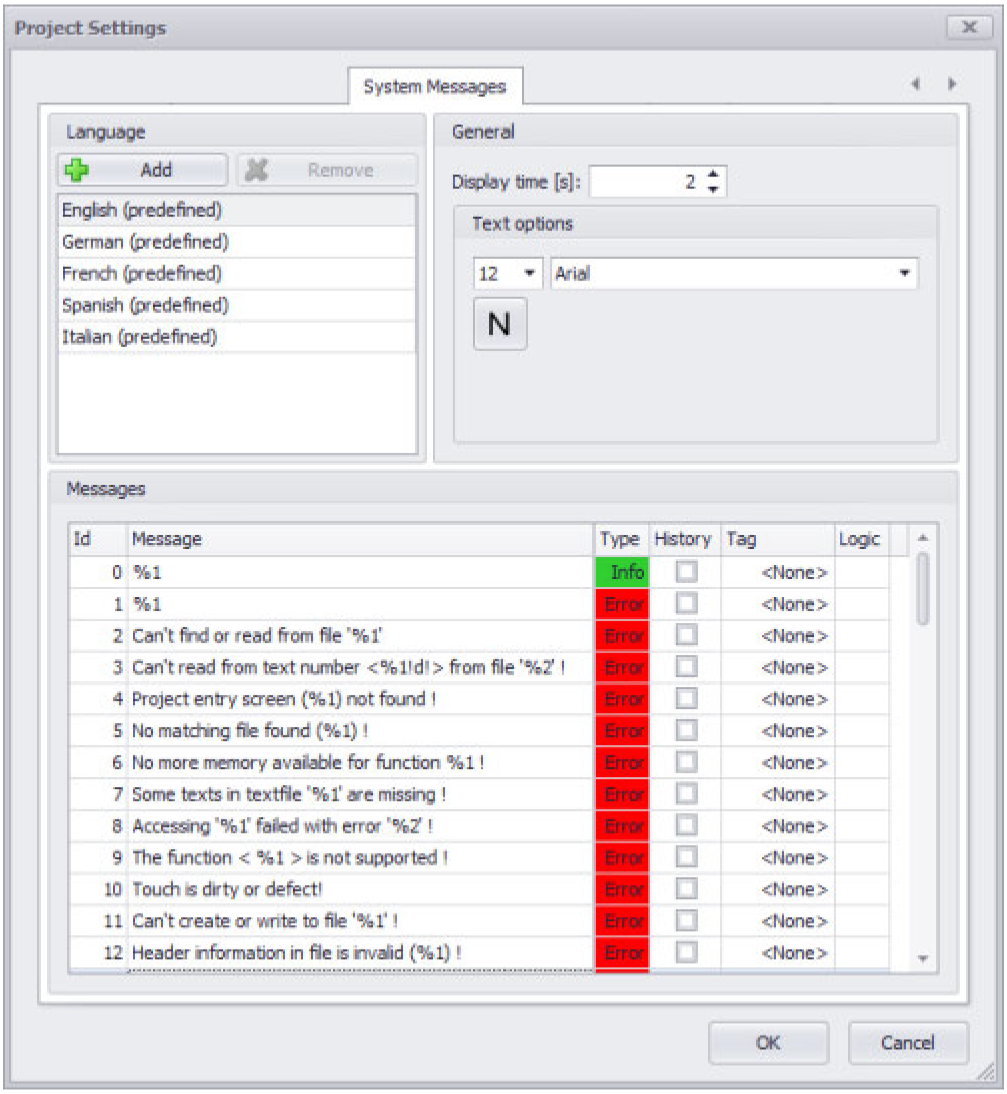
System messages are output by the GRS as required.

Messages are written on the screen in two different ways. Errors are shown in list form during the initialization phase.

In devices with Windows, the initialization stage can be interrupted with a long press gesture or by holding the left mouse button down. The first screen will be displayed only after letting go.

System messages will later be shown in the Output pane, which will only be shown for a certain amount of time.

Home\Project Configuration\Settings tab



English (predefined)
German (predefined)
French (predefined)
Spanish (predefined)
Italian (predefined)

Display time [s]: 2

Text options

12 Arial

N

Id	Message	Type	History	Tag	Logic
0	%1	Info	□	<None>	
1	%1	Error	□	<None>	
2	Can't find or read from file '%1'	Error	□	<None>	
3	Can't read from text number <%1!d!> from file '%2' !	Error	□	<None>	
4	Project entry screen (%1) not found !	Error	□	<None>	
5	No matching file found (%1) !	Error	□	<None>	
6	No more memory available for function %1 !	Error	□	<None>	
7	Some texts in textfile '%1' are missing !	Error	□	<None>	
8	Accessing '%1' failed with error '%2' !	Error	□	<None>	
9	The function < %1 > is not supported !	Error	□	<None>	
10	Touch is dirty or defect!	Error	□	<None>	
11	Can't create or write to file '%1' !	Error	□	<None>	
12	Header information in file is invalid (%1) !	Error	□	<None>	

Figure 144: *Window Project Settings/System Messages*

Language

If you want to add additional languages to your project on top of the pre-defined languages, you can use this section to create the corresponding language data records.

✚Add	Adds a new language to the project
xxx (predefined)	Data records automatically generated for every project. As of this writing: English, German,

	French, Spanish, and Italian The active language, i.e., the language in which the system messages are being shown in the Messages section, will be highlighted in color.
❌Remove	Removes the selected language from the project (can only be used with additional languages created by the user, i.e., this option will not be available for pre-defined languages)

General

Display time	The length of time that the system message will be displayed, in seconds
Text options	Settings used to configure the formatting for system messages

Messages

This section is used to define the message texts that will be generated automatically by the GALILEO Runtime System in the event of faults or operator intervention.

ID	Consecutive ascending number in the system
Message	Message text that will be generated
Catalog Number	Message type
About	These system messages are provided for information purposes only and will be displayed inside a green bar on the panel. An example is when the project language is changed on the panel.
Error	GALILEO Runtime System system errors. These messages will be displayed inside a red bar on the panel.
History	If this checkbox is enabled (☑), the corresponding message will be logged in the log file.

Tag If a tag is selected here, the bit tag will be set to 1 when the system message is triggered. If the tag already has a value of 1 when the system message is triggered, its value will remain the same. Please note that only bit tags with a write data direction will be available for selection here.

→ An option for resetting the bit tag to 0 must be included in the project, as the system will never reset the tag back to 0 by itself.

Logic Defines the logic how the tag is to be set.

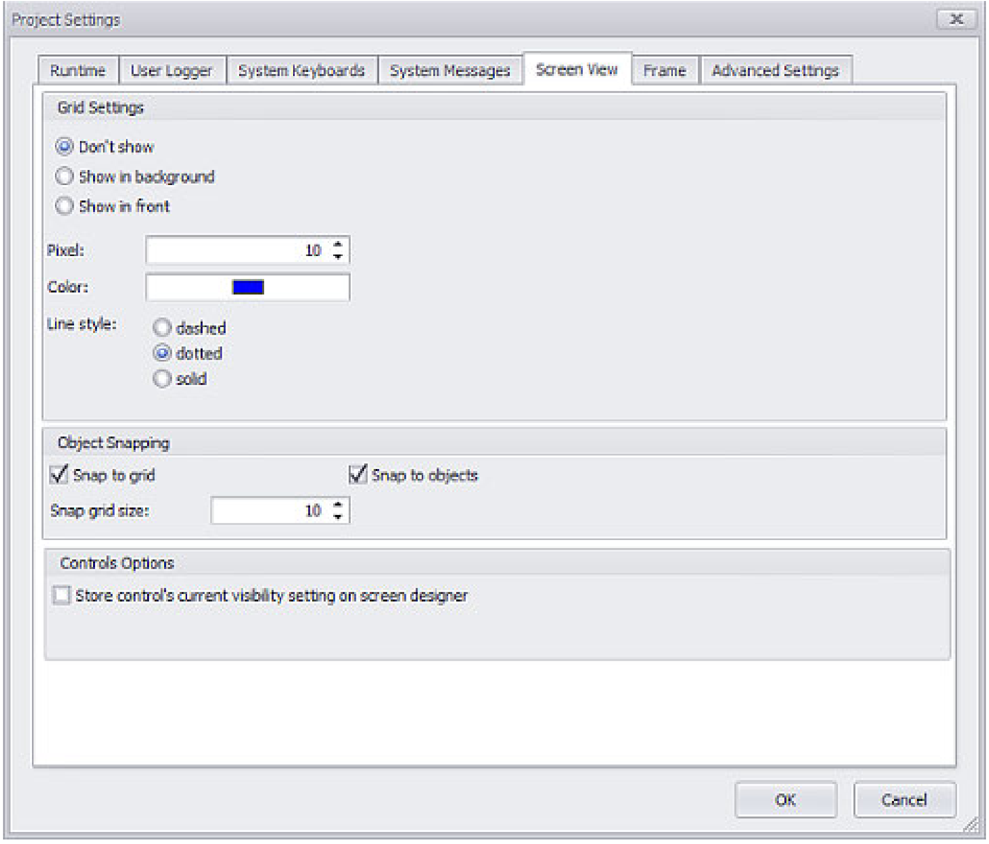
See also

→ section "System Messages", page 1147

Screen View

This tab can be used to configure the grid that helps visually design screens, as well as the settings for aligning the individual objects and for aligning objects relative to each other on a screen.

Home\Project Configuration\Settings tab



Grid Settings

● Don't show
○ Show in background
○ Show in front

Pixel: 10

Color: 🟦

Line style: ○ dashed ● dotted ○ solid

Object Snapping

☑ Snap to grid ☑ Snap to objects

Snap grid size: 10

Controls Options

□ Store control's current visibility setting on screen designer

Figure 145: *Project Settings/Screen View window*

Grid Settings

The options for configuring whether and how the grid will be shown on the screen are found in this section.

◉ One of the two options needs to be selected

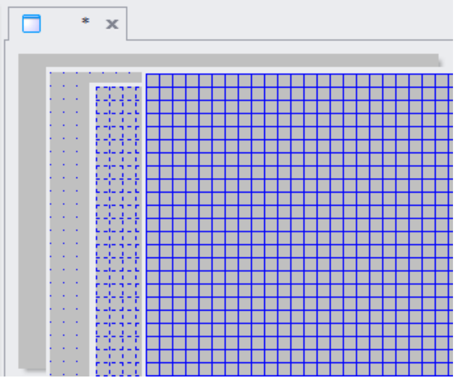


Figure 146: Grid as a positioning assistance tool

Don't show: The grid will not be shown

Show in background The grid will be shown on the bottommost screen layer, meaning all inserted objects will be shown on top of the grid.

Show in front The grid will always be on top of the various objects

Pixel: Used to define the spacing between grid markings (dots, dashes, or lines).

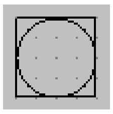
Color: Used to select the color that will be used to display the grid markings.
For detailed information on the available colors, please refer to the following section: → section "Color choice - Object's Color Palette", page 234

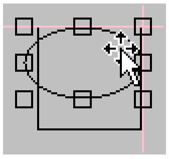
Line style: ◉ One of the two options needs to be selected
Used to specify the type of line that will be shown, i.e., dashed, dotted, or solid.

Object Snapping

Whenever a new object is drawn, or whenever an object is moved, enlarged, or shrunk, it will do so by the increment configured here.
Please note that distortion caused when an object is enlarged or shrunk as a result of the configured snap grid size cannot be corrected without changing the grid size as required.

Snap to grid If this option is enabled ☑,
it will be possible to achieve more exact results when free-drawing and sizing objects, as objects will be positioned and sized using this uniform grid increment.



Snap to objects	If this option is enabled ☑, colored lines aligned with the closest object on the screen will be shown when positioning an object in order to make it easier to align the objects.  Not available when multiple objects have been selected.
Snap grid size	Used to define the size of the grid increments when positioning objects, in pixels

→ The snap grid does not have to have the same grid settings as the screen.

Use case: Aligning several objects of the same size on a screen

In this example: Objects with a fixed size of 50 x 50 pixels will be positioned at a distance of 10 x 10 pixels from each other on the screen

Prerequisites

- The objects must have already been created on the screen
- The following Object Snapping settings must be configured in *Project Configuration\Settings\* Screen View tab:

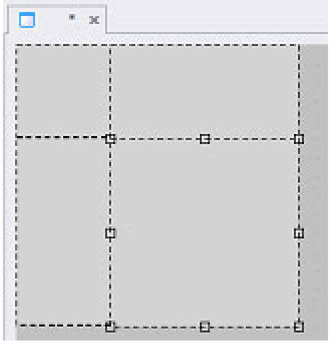
☑ Snap to grid
☑ Snap to objects
Snap grid size: 50

The Grid Settings are not relevant, as they only define how the grid will be displayed.
● Show in background
Pixel: 10
Color: A color different from the screen's color
Line style: ● dotted

▶ Select one or more objects on the screen

▶ Draw an object shape.
The object will snap to a multiple of the snap grid size

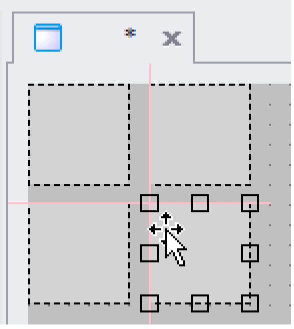
Size W/H of 50 x 50, 50 x 100, 100 x 50, 100 x 100 pixels, etc.



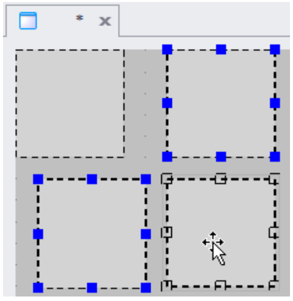
The size will be shown in the Properties 'Button' pane.

The object size can be adjusted in the Properties 'Button' pane independently of the grid.

- Adjust all objects so that they have the same size.
- Go to *Project Configuration\Settings* and decrease the snap grid size from 50 to 10
- Individually align the objects relative to each other at a uniform distance of 10 pixels by moving them.



- You can also align multiple objects together.



Controls Options

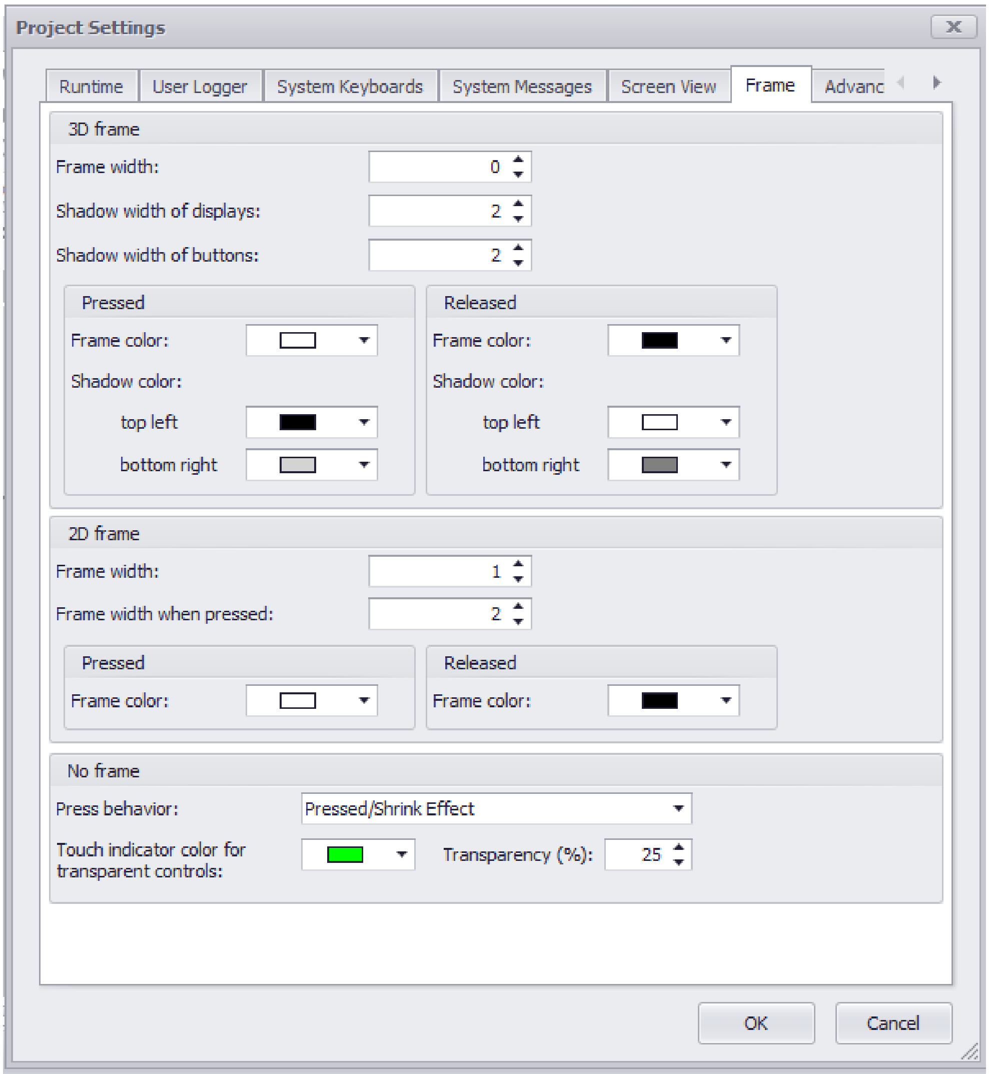
Store control's current visibility settings on screen designer	If this option is enabled ☑ , the current visibility setting of the controls on the screen designer (set via the Controls Manager) will be stored together with the project and be

applied on the next opening of the project. Because this could be very irritating for others users when opening the project, there will be a warning messsage appearing when opening the project.

Frame

This tab contains the settings for the way in which all visualization objects on the screens will be displayed.
The settings configured here will be applied when a style is selected as a property for a visualization object.

Home\Project Configuration\Settings
tab



3D frame

Frame width: 0

Shadow width of displays: 2

Shadow width of buttons: 2

Pressed

Frame color: □

Shadow color:

top left ■

bottom right □

Released

Frame color: ■

Shadow color:

top left □

bottom right ▨

2D frame

Frame width: 1

Frame width when pressed: 2

Pressed

Frame color: □

Released

Frame color: ■

No frame

Press behavior: Pressed/Shrink Effect

Touch indicator color for transparent controls: 🟩

Transparency (%): 25

Figure 147: *Project Settings/Frame window*

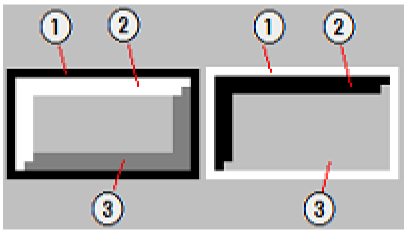
3D Settings

For objects on the screen that have a frame and are shadowed so that they will look like 3D objects

Outline Width — Defines the width of the object's outer borders.

Shadow width of displays: — Display setting for active and inactive objects on the screen

Shadow width of buttons: — Setting used to define how strong the 3D effect should be



① Outline Width

② Display shadow width

③ Button shadow width

The display of the buttons can be set for the respective state Pressed or Released separately.

Frame color: — Used to select the color for the frame

Shadow color: — Used to select the color for the shadowing used for the 3D effect. There are two separate settings in this case:
top left
bottom right

2D Settings

For objects on the screen that have a frame and look like 2D objects

Frame color for a button's or display object's borders

Outline Width — Defines the width of the object's outer borders.

Selected frame width: — Defines the width of the frame shown when the button is being pressed
or when the cursor is moved over it

The display of the buttons can be set for the respective state Pressed or Released separately.

Frame color: — Used to select the color for the frame

No Frame

For objects on the screen that do not have a frame and that only get a frame when activated, e.g., when the cursor is moved over a button

Press behavior:

Semi Transparent or

Pressed/Shrink Effect is available for selecting the display.

Touch indicator color for transparent controls: Selection of the color in which the touch panel-application gives visual feedback in case the background color: Transparent is set and no foreground is defined.

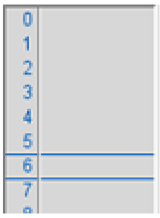
For detailed information on the available colors, please refer to the following section:
→ section "Color choice - Object's Color Palette", page 234

List selection style

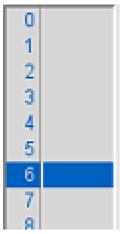
Used to specify ◉ how list elements will be shown when selected.

◉ One of the two options needs to be selected

With frame A frame will be shown around the selected item.



With inverted color The selected item will be highlighted in color.



Advanced Settings

These settings can be used to configure settings for special parameters or internal statements.
For more information, please refer to the Performance Optimization Guide, which you can access as a PDF document (available only in English) by clicking on ▤ \ ? Help and Tools.

- ▶ Click on the pane and enter the text you want.
- ▶ Click on OK to save the text in the project.
- ▶ Click on Cancel to exit the editor without saving your text.

Home\Project Configuration\Settings tab

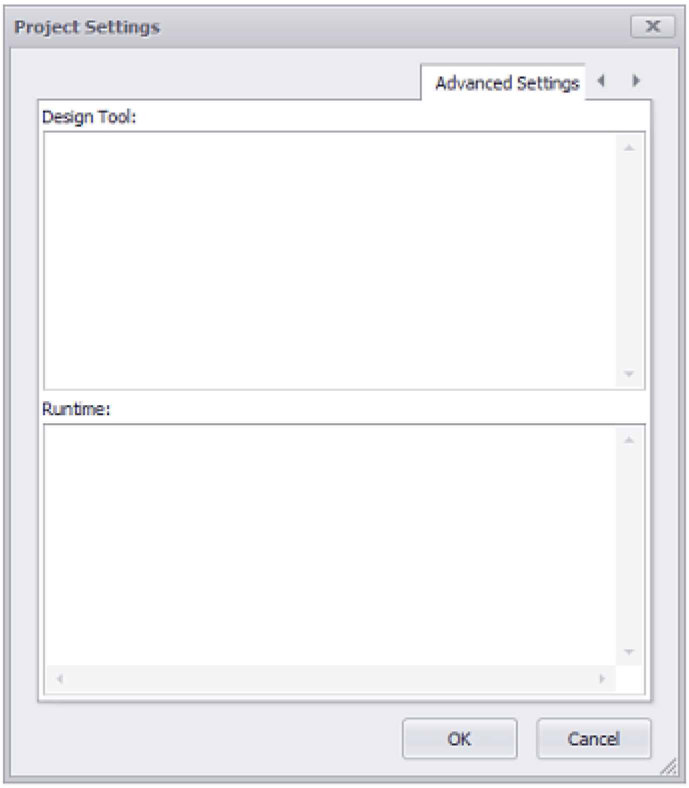


Figure 148: *Project Settings:Advanced Settings window*

1.6.5.4 Optimizing project

The project optimization function can be used to prune project objects that were created in the project tree or while working on the project but that in the end were not used.

To do this, the function will search for objects of a specific type that are not being used, after which these objects can be selected and deleted.

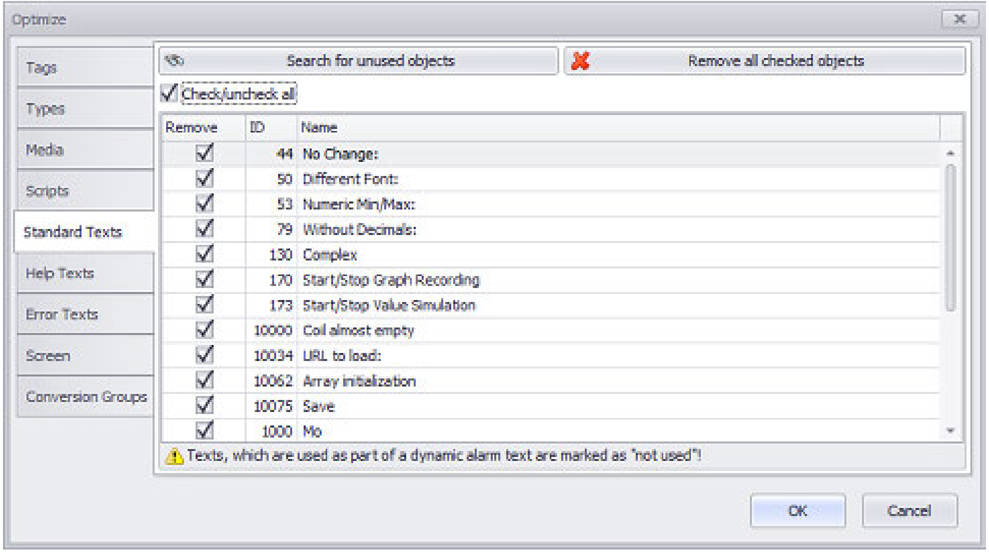
Pruning will shorten the time it takes to open/save/close and reduce the size of the project file.
Optimization instructions for the GALILEO Runtime System are described separately in the "Performance Optimization Guide."

The following object types can be pruned using this function:

- Tags
- Types
- Media
- Scripts
- Standard Texts
- Help Texts
- Error Texts

- Screen
- Conversion Groups

Home\Project Configuration\Optimize tab



Optimize

Tags
Types
Media
Scripts
Standard Texts
Help Texts
Error Texts
Screen
Conversion Groups

Search for unused objects   Remove all checked objects

☑ Check/uncheck all

Remove	ID	Name
☑	44	No Change:
☑	50	Different Font:
☑	53	Numeric Min/Max:
☑	79	Without Decimals:
☑	130	Complex
☑	170	Start/Stop Graph Recording
☑	173	Start/Stop Value Simulation
☑	10000	Coil almost empty
☑	10034	URL to load:
☑	10062	Array initialization
☑	10075	Save
☑	1000	Mo

⚠ Texts, which are used as part of a dynamic alarm text are marked as "not used"!

OK   Cancel

Figure 149: Window Optimize

NOTICE
Objects that are only activated via PLC by using the screen ID will be listed as not used!

Search for unused objects

Searches for objects of the selected type that exist but that are not being used in the open project.

☑ The structures for which the checkbox is enabled will be searched through

Remove all checked objects

Completely deletes the objects from the structure.
As long as the project has not been saved, you will still be able to use the undo function to undo this action.

☑ Remove

The selected structures will be deleted with OK.

☑ Check/uncheck all

refers to the found objects that are unused

See also

→ section "Confirmation", page 137

1.6.5.5 Settings for communicating with devices

GALILEO is Automatization Software that makes it possible to design and create application-specific man-machine interfaces. These interfaces are provided to device users so that they can view the relevant machine's data in order to be able to make decisions. This, in turn, enables device users to enter input and control the machine accordingly.

To this end, the application from the Automatization Software communicates with the machine's PLC.

The interface is created when a project is built using the visualization software. This project is loaded onto the device or, more specifically, the touch panel. The program is then started on the touch panel and will provide the device user with the PLC's data in the form of values or characters and vice versa.
Values and characters can also be entered on the device and sent to the PLC.

→ " Select panel", page 221	→ "Settings Visualization", page 1085	→ "Settings for connection", page 1089	→ "The certificate manager helps securely manage the certificates in the GALILEO project.", page 1091	→ "PLC Selection & Configuration", page 585	→ " CE Configuration", page 228

Select panel

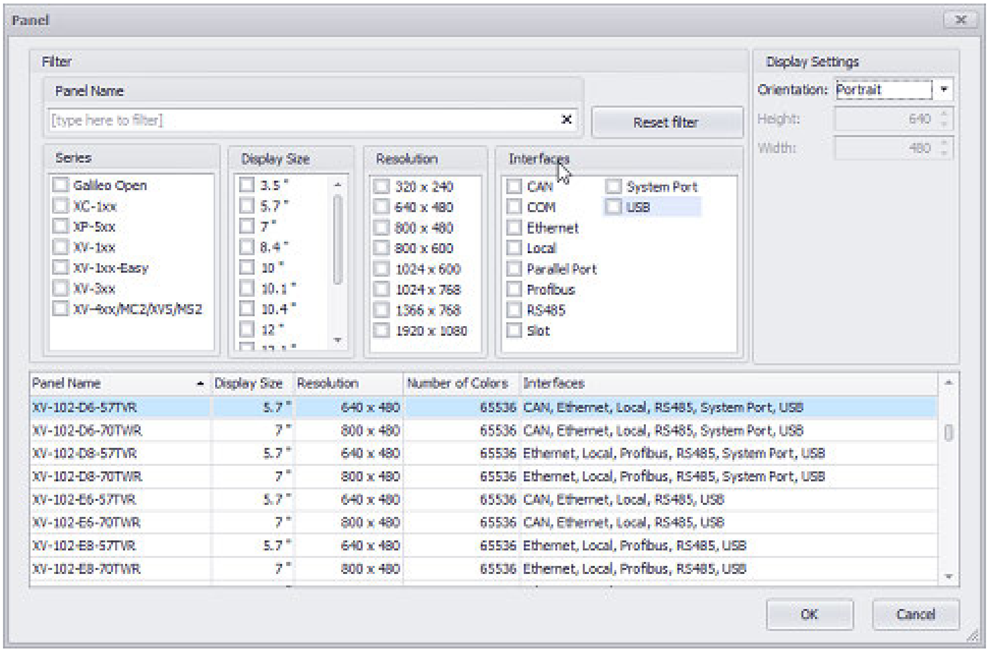
This dialog box can be used to change panels when you want a project that you have already created to be deployed as a user interface on a different panel. The dialog box features a device filter function designed to make it easier to find the panel model you want.

A panel is always selected when first creating a project.

If the display size or the screen settings change, all the screens in the project will need to be modified accordingly.

Only one panel can be selected per project.

Device tab\ Panel type



Panel

Filter

Panel Name

[type here to filter]

Reset filter

Series
- □ Galileo Open
- □ XC-1xx
- □ XP-5xx
- □ XV-1xx
- □ XV-1xx-Easy
- □ XV-3xx
- □ XV-4xx/MC2/XVS/MS2

Display Size
- □ 3.5 "
- □ 5.7 "
- □ 7 "
- □ 8.4 "
- □ 10 "
- □ 10.1 "
- □ 10.4 "
- □ 12 "

Resolution
- □ 320 x 240
- □ 640 x 480
- □ 800 x 480
- □ 800 x 600
- □ 1024 x 600
- □ 1024 x 768
- □ 1366 x 768
- □ 1920 x 1080

Interfaces
- □ CAN
- □ COM
- □ Ethernet
- □ Local
- □ Parallel Port
- □ Profibus
- □ RS485
- □ Slot
- □ System Port
- □ USB

Display Settings

Orientation: Portrait

Height: 640

Width: 480

Panel Name	Display Size	Resolution	Number of Colors	Interfaces
XV-102-D6-57TVR	5.7 "	640 x 480	65536	CAN, Ethernet, Local, RS485, System Port, USB
XV-102-D6-70TWR	7 "	800 x 480	65536	CAN, Ethernet, Local, RS485, System Port, USB
XV-102-D8-57TVR	5.7 "	640 x 480	65536	Ethernet, Local, Profibus, RS485, System Port, USB
XV-102-D8-70TWR	7 "	800 x 480	65536	Ethernet, Local, Profibus, RS485, System Port, USB
XV-102-E6-57TVR	5.7 "	640 x 480	65536	CAN, Ethernet, Local, RS485, USB
XV-102-E6-70TWR	7 "	800 x 480	65536	CAN, Ethernet, Local, RS485, USB
XV-102-E8-57TVR	5.7 "	640 x 480	65536	Ethernet, Local, Profibus, RS485, USB
XV-102-E8-70TWR	7 "	800 x 480	65536	Ethernet, Local, Profibus, RS485, USB

Figure 150: Window Select Panel

The panel currently selected for the project will be highlighted and will be shown at the top of the list pane when the dialog box first opens.

There are various options for changing the panel you are using.

Reset Filter	Clears all filter selections

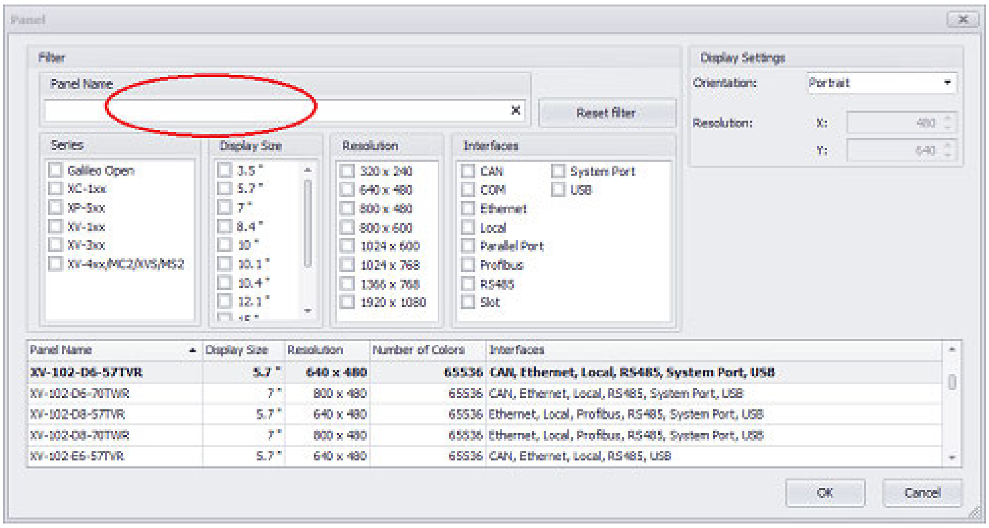
Input field – identifying the panel using the nameplate

→ You can get the panel name from the nameplate on the back of the panel.

The name will contain the series to which the panel belongs, as well as the relevant display size, resolution, and available ports and interfaces in coded format.

▶ Enter the panel name you want into the input field **<...>**

Device tab\ Panel type



Orientation: Portrait

Resolution: X: 480 Y: 640

Panel Name	Display Size	Resolution	Number of Colors	Interfaces
XV-102-D6-57TVR	**5.7 "**	**640 x 480**	**65536**	**CAN, Ethernet, Local, RS485, System Port, USB**
XV-102-D6-70TWR	7 "	800 x 480	65536	CAN, Ethernet, Local, RS485, System Port, USB
XV-102-D8-57TVR	5.7 "	640 x 480	65536	Ethernet, Local, Profibus, RS485, System Port, USB
XV-102-D8-70TWR	7 "	800 x 480	65536	Ethernet, Local, Profibus, RS485, System Port, USB
XV-102-E6-57TVR	5.7 "	640 x 480	65536	CAN, Ethernet, Local, RS485, USB

Figure 151: Panel dialog box

The options available for selection in the lower pane will change automatically to reflect the options you select above and any text you enter into the Panel Name field.

Clicking on a panel will apply the corresponding parameters internally. However, the various options on the dialog box will not be changed.

Double-clicking on a panel will select the panel model and cause the dialog box to go to the next project creation step.

▶ Click on OK.

▶ To select the touch panel you want, click on it in the list.

The item you click on will be highlighted in blue.

Identifying the panel using individual parameters

You can use the Series filter criterion to pre-filter the panel models being shown.

You can also individually select the panel's Display Size and Resolution, as well as the Interfaces that will be used for communication, as additional filter criteria.

You can also select multiple options if necessary. The project will adjust the list of panels available for selection in real time when you select an option.

Simply enable the checkbox (☑) to the left of the options you want by clicking on it.

Only one panel can be selected per project.

When you enable one or more parameters, some of the options for the other criteria may be grayed out as a result of not being available for the parameters you selected.

If the information you enter does not match any of the available panel configurations, a message to this effect will be shown.

→ If the panel size is not selected correctly or is changed later on, you will need to check the position of the various objects on all of the corresponding project's screens.
→ "Notes on creating a Project", page 192
→ section "Display Settings", page 225

▶ Enable the panel parameters that you know and that apply.
You can use the Series:, Display Size:, Resolution:, and Interfaces: criteria to filter the list of panels being shown.

The options available for selection in the lower pane will change automatically to reflect the options you select above and any text you enter into the Panel Name field.

Clicking on a panel will apply the corresponding parameters internally. However, the various options on the dialog box will not be changed.

Double-clicking on a panel will select the panel model and cause the dialog box to go to the next project creation step.

▶ To select the touch panel you want, click on it in the list.

The item you click on will be highlighted in blue.

▶ Click on OK.

Selecting a panel model from the list

You can also simply select a panel directly from the list.

▶ Scroll to the panel you want in the list.

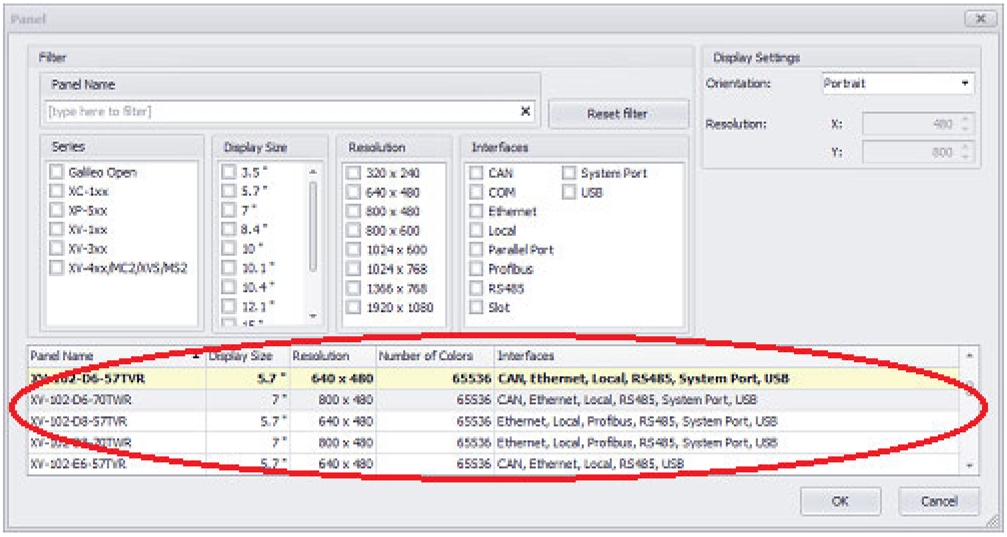
▶ To select the touch panel you want, click on it in the list.

The item you click on will be highlighted in blue.

▶ Click on OK.

Example showing how to select a panel from the list

Device tab\ Panel type



Panel

Filter

Panel Name

[type here to filter]

Reset filter

Series	Display Size	Resolution	Interfaces	
□ Galileo Open	□ 3.5 "	□ 320 x 240	□ CAN	□ System Port
□ XC-1xx	□ 5.7 "	□ 640 x 480	□ COM	□ USB
□ XP-5xx	□ 7 "	□ 800 x 480	□ Ethernet	
□ XV-1xx	□ 8.4 "	□ 800 x 600	□ Local	
□ XV-3xx	□ 10 "	□ 1024 x 600	□ Parallel Port	
□ XV-4xx/MC2/XVS/MS2	□ 10.1 "	□ 1024 x 768	□ Profibus	
	□ 10.4 "	□ 1366 x 768	□ RS485	
	□ 12.1 "	□ 1920 x 1080	□ Slot	
	□ 15 "			

Display Settings

Orientation: Portrait

Resolution: X: 480 Y: 800

Panel Name	Display Size	Resolution	Number of Colors	Interfaces
XV-102-D6-57TVR	**5.7 "**	**640 x 480**	**65536**	**CAN, Ethernet, Local, RS485, System Port, USB**
XV-102-D6-70TWR	7 "	800 x 480	65536	CAN, Ethernet, Local, RS485, System Port, USB
XV-102-D8-57TVR	5.7 "	640 x 480	65536	Ethernet, Local, Profibus, RS485, System Port, USB
XV-102-D8-70TWR	7 "	800 x 480	65536	Ethernet, Local, Profibus, RS485, System Port, USB
XV-102-E6-57TVR	5.7 "	640 x 480	65536	CAN, Ethernet, Local, RS485, USB

Figure 152: Selecting a panel from the list

Display Settings

The screen settings are determined by the orientation and resolution being used.

All XV devices can be installed in both landscape and portrait modes.

Table 3: Screen settings: orientation and resolution

Portrait	Landscape
320 x 240	320 x 240
640 x 480	640 x 480
800 x 480	800 x 480
800 x 600	800 x 600
1024 x 600	1024 x 600
1024 x 768	1024 x 768
1366 x 768	1366 x 768
1920 x 1080	1920 x 1080

Depending on the touch panel you select, the Resize options dialog box may allow you to adjust the orientation and resolution being used, e.g., to switch from portrait mode to landscape mode or vice versa. The dialog box can also be used to specify whether the objects should be resized or keep their size in these cases.

Orientation

Used to select the orientation for the screens based on how the panel will be used
Certain panels support both modes, in which case one will need to be selected. The screen resolution will be automatically adjusted accordingly. Please note that the mode selected here will need to

be selected on the device as well.

Resolution

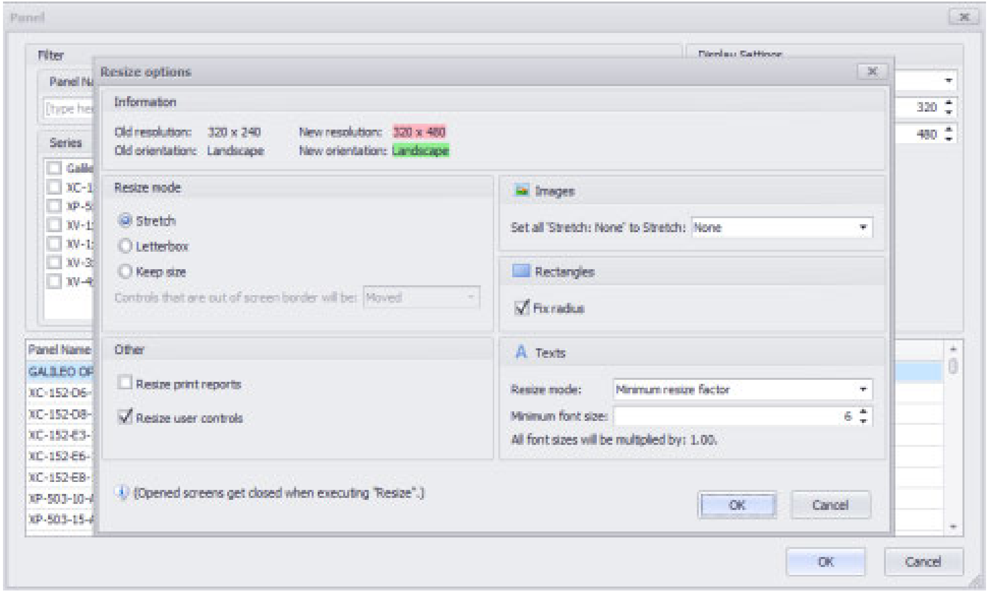
These two fields specify the resolution in the horizontal (X) and vertical (Y) directions, in pixels.
The only case in which these two fields will be enabled will be when using a panel that supports adjusting the resolution.

In order for the screens in the project to be displayed correctly, the same orientation must be selected in the project and on the actual device itself.
The orientation setting only applies to devices running a Windows operating system (Desktop and CE).

Making changes to the display settings

Changing the screen setting values will open the Resize options dialog box. This dialog box will show the resolution and/or orientation changes highlighted in color and provide additional options.

Device\Panel Type tab



Resize options

Information

Old resolution: 320 x 240 New resolution: 320 x 480
Old orientation: Landscape New orientation: Landscape

Resize mode

● Stretch
○ Letterbox
○ Keep size

Controls that are out of screen border will be: Moved

Images

Set all 'Stretch: None' to Stretch: None

Rectangles

☑ Fix radius

Other

□ Resize print reports
☑ Resize user controls

Texts

Resize mode: Minimum resize factor
Minimum font size: 6
All font sizes will be multiplied by: 1.00.

(Opened screens get closed when executing "Resize".)

Figure 153: Resize options dialog box shown after changing the display settings

Information

The parameters that have changed will be highlighted in pink
The parameters remaining the same will be highlighted in green
Orientation and/or resolution settings

Resize mode

Used to define what should be done with the objects on the screens if the orientation or size has changed.
● One of the two options needs to be selected

Stretch

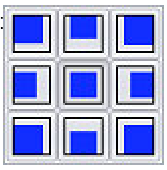
Objects will be stretched or shrunk as appropriate for the new resolution.

Letterbox

All objects will be positioned relative to each other in such a way as to keep the same basic proportions. Depending on the orientation being used, clicking on the ⊡ icon will provide the following position options:
Top (▀), centered (■), or bottom (▄)
Or left (▌), centered (■), or right (▐)

Keep size

If you decide to keep the size or select a larger screen, all objects will be positioned relative to each other using the same proportions.
Clicking on the ■ icon will provide the following position options:



Objects that are out of screen border will be:

Enabled only if the Keep size option is enabled
If the resolution is lower or the new orientation makes the width or height smaller, you will need to decide
What should be done with the objects that now fall partially or entirely outside the new screens.
The options in this case are "Move" and "Delete."

Images and Rectangles

These settings are used to specify whether all images should be stretched (☑) and whether the radii for rounding rectangles should be adjusted (☑)

Other

Print reports are not tied to the screen size. This means that reports can be configured in such a way as to be printed in DIN A4 size, for example.
☑ If this option is enabled, the print reports will be stretched/shrunk in the same proportion as the screens.

Texts A

Used to adjust text boxes

Resize mode

The resize mode option you select will determine what will be done with text boxes when resizing.
The factor indicates the multiplier that will be used to enlarge or shrink the font size
You can also set a minimum font size.
Example: font size of 10 -- >10 x 0.8 = 8 --> the font size will be changed to 8.

None

Factor of 1.0 - The font size will remain the same

Depends on the area change

Shows the factor that is yielded by the selected options for

	calculating the font size.
Minimum resize factor	Factor of 0.x - Text will be shrunk
Combine minimum and area	Font resize factor for objects that use word wrap. If there is no word wrap in the text, the second factor will apply.
Minimum font size	Used to specify the smallest font size that will be shown

Click on Resize to apply the changes or Cancel to close the dialog box.

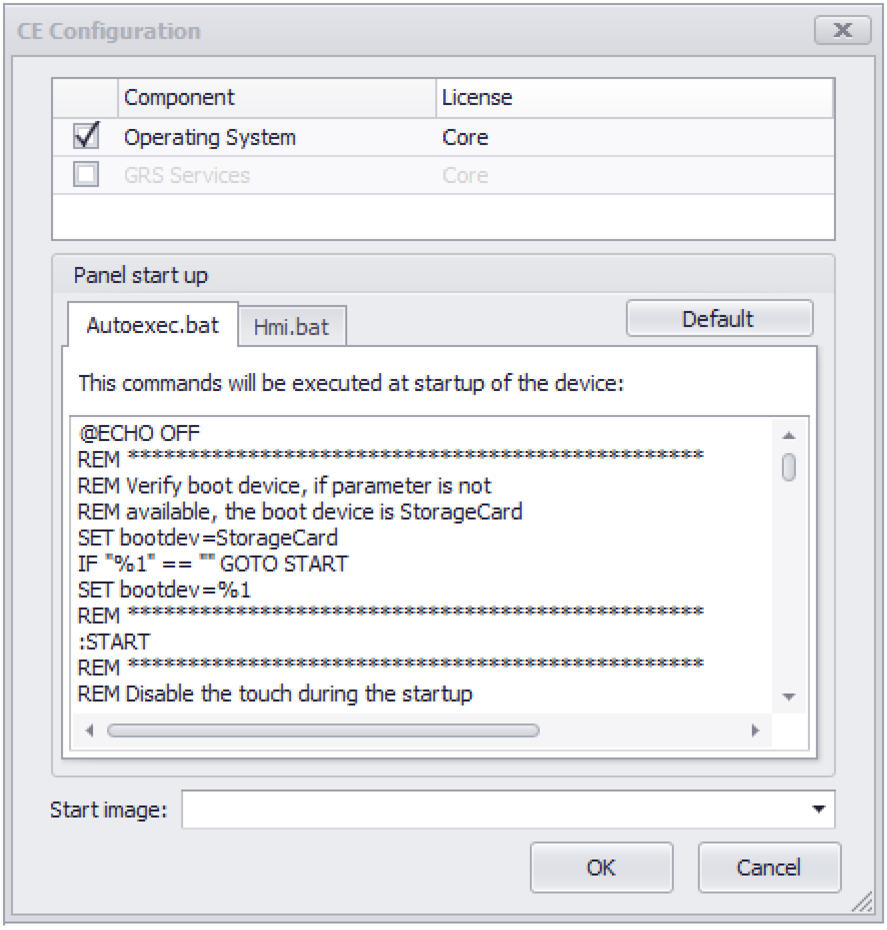
CE Configuration

This dialog box can be used to configure settings for the touch panel's startup behavior.

The autoexec.bat and hmi.bat batch files will be automatically loaded onto the device together with the GALILEO application unless the Don't overwrite autoexec.bat option has been enabled in the Download options. Please refer to → section "Download options", page 1061 as well.

When the device is started, the autoexec.bat batch file will be run automatically. This file will call the hmi.bat batch file by default. Both batch files can be edited as necessary.

Device tab\ CE Configuration



CE Configuration

	Component	License
☑	Operating System	Core
□	GRS Services	Core

Panel start up

This commands will be executed at startup of the device:

```
@ECHO OFF
REM ****************************************************
REM Verify boot device, if parameter is not
REM available, the boot device is StorageCard
SET bootdev=StorageCard
IF "%1" == "" GOTO START
SET bootdev=%1
REM ****************************************************
:START
REM ****************************************************
REM Disable the touch during the startup
```

Start image:

Figure 154: CE Configuration dialog box

The .bat files shown in the figure as an example will be automatically loaded onto the device during the download.

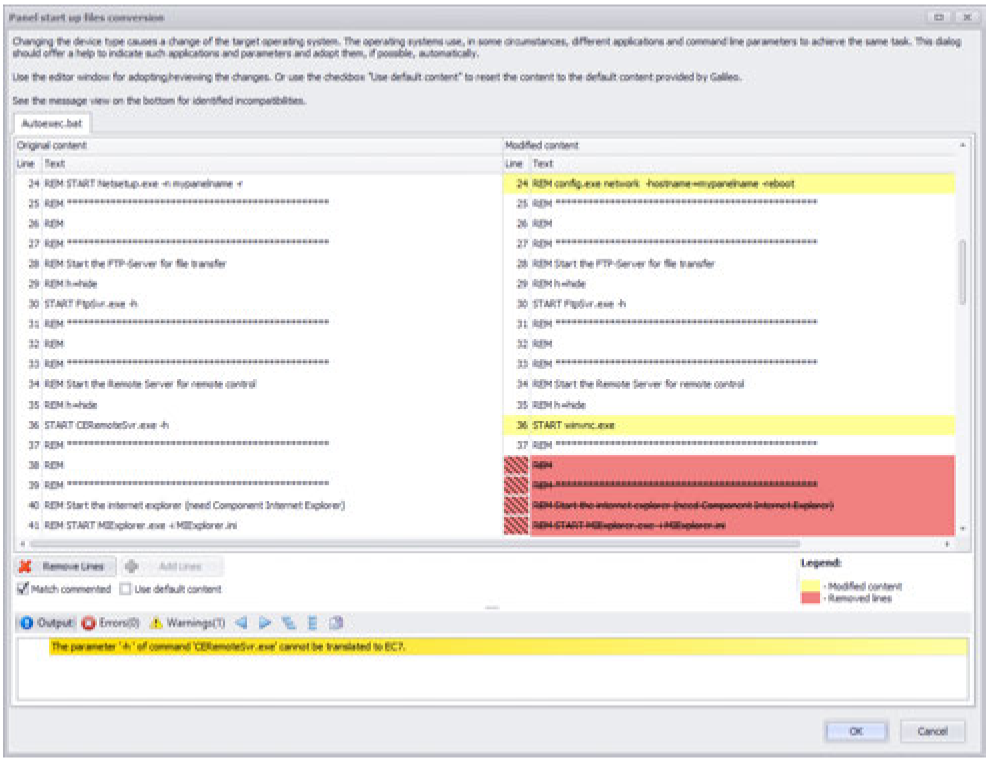
→ "Compile and transfer", page 1035

1.6.5.6 Changing the panel model

Changing the panel model in an existing project can result in certain incompatibilities.
For example, there are certain system calls (e.g., for setting the IP address) that are different between XV300 and XV100 panel models.

GALILEO will systematically check the project in order to identify any incompatibilities and will change the calls for the new panel model in such a way that they will continue to work the same way.
Accordingly, the following dialog box may appear after changing the panel model:

Device tab\ ▬ Panel type



Panel start up files conversion

Changing the device type causes a change of the target operating system. The operating systems use, in some circumstances, different applications and command line parameters to achieve the same task. This dialog should offer a help to indicate such applications and parameters and adopt them, if possible, automatically.

Use the editor window for adopting/reviewing the changes. Or use the checkbox "Use default content" to reset the content to the default content provided by Galileo.

See the message view on the bottom for identified incompatibilities.

Autoexec.bat

Original content		Modified content	
Line	Text	Line	Text
24	REM START Netsetup.exe -n mypanelname -r	24	REM config.exe network -hostname=mypanelname -reboot
25	REM **	25	REM **
26	REM	26	REM
27	REM **	27	REM **
28	REM Start the FTP-Server for file transfer	28	REM Start the FTP-Server for file transfer
29	REM h=hide	29	REM h=hide
30	START FtpSvr.exe -h	30	START FtpSvr.exe -h
31	REM **	31	REM **
32	REM	32	REM
33	REM **	33	REM **
34	REM Start the Remote Server for remote control	34	REM Start the Remote Server for remote control
35	REM h=hide	35	REM h=hide
36	START CERemoteSvr.exe -h	36	START winvnc.exe
37	REM **	37	REM **
38	REM	▨	~~REM~~
39	REM **	▨	~~REM **~~
40	REM Start the internet explorer (need Component Internet Explorer)	▨	~~REM Start the internet explorer (need Component Internet Explorer)~~
41	REM START MIExplorer.exe -i MIExplorer.ini	▨	~~REM START MIExplorer.exe -i MIExplorer.ini~~

☑ Match commented ☐ Use default content

Legend:
🟨 - Modified content
🟥 - Removed lines

The parameter '-h' of command 'CERemoteSvr.exe' cannot be translated to EC7.

Figure 155: File conversion for the panel's startup behavior

This dialog box will show the current contents of the project's "autoexec.bat" file in the left column.

Meanwhile, the right column will show suggested corrections.
The lines highlighted in yellow consist of automatically modified content.
In contrast, the lines highlighted in red consist of calls that are not supported at all anymore and that accordingly will be deleted.

You can use the "Remove Lines" and "Add Lines" buttons to make manual changes.

You can use the ☑ Match commented option to define whether lines that have been commented out should be checked as well.

Additionally, the Use default content option can be used to apply the default GALILEO settings for the new panel model.

→ If the ☑ Use default content option is enabled, any manual changes to the project will be lost, and you will need to make them again afterwards.

1.6.5.7 Colors in GALILEO

GALILEO can be used to select the color depth for any touch panel (typically 256 or 65536 colors). Images and objects for which a color can be selected will always be saved with a maximum of 256 colors. Of course, when images and objects with colors are displayed, the question will be which colors will be used to display them on the touch panel.

Colors in objects and images

Color properties of objects for which colors can be selected

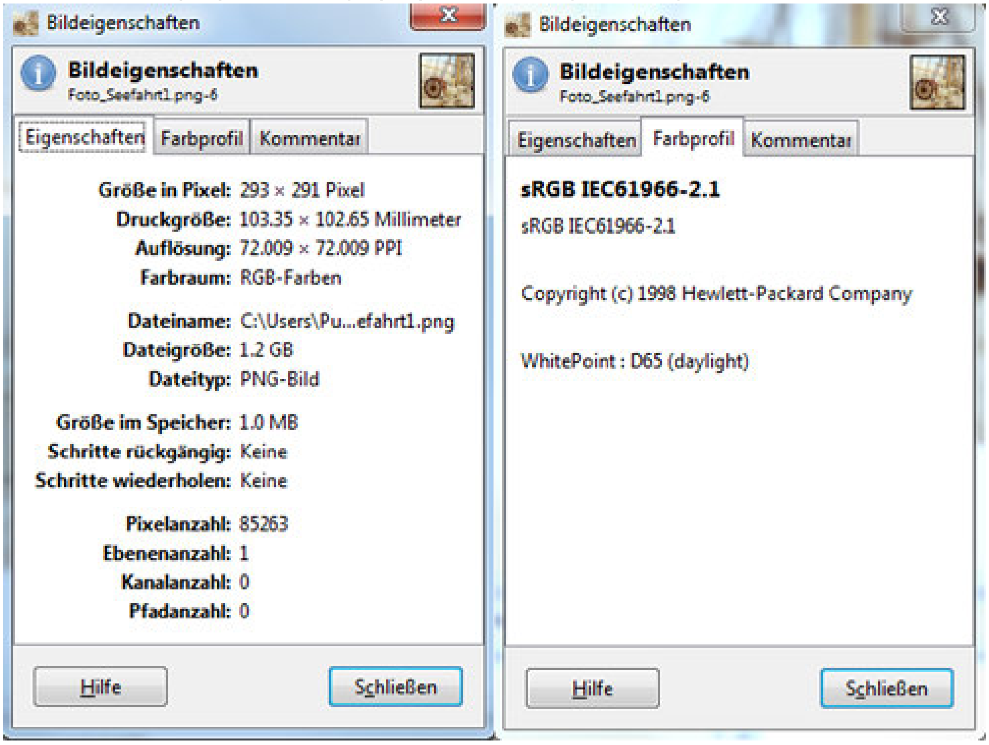
In the case of drawn objects, as well as when selecting colors for objects, a color can be selected from the color palette as a property. This color palette can be customized by replacing colors not being used with new colors or by simply creating a completely new color palette. In fact, this can be a good idea when you need colors not found in the existing palette. However, please note that only one color palette can be used per project, and that it will always be limited to 256 colors.

Objects will be displayed on the panel exactly as defined in GALILEO, as the color palette will also be transferred to the panel.

Image color properties

When imported into a project, images will be converted to PNG format. During the import process, a color palette with a maximum of 256 colors will be generated for each individual image. If an image has a color depth with more than 256 colors, a color palette with 256 colors optimized for the image based on its color space will be generated when the image is imported and converted to PNG format. The fact that an individual color palette is generated for each image ensures that the color characteristics of every single image will be preserved.

For illustration purposes: image properties of original image with 60641 colors



Bildeigenschaften

Bildeigenschaften
Foto_Seefahrt1.png-6

Größe in Pixel: 293 × 291 Pixel
Druckgröße: 103.35 × 102.65 Millimeter
Auflösung: 72.009 × 72.009 PPI
Farbraum: RGB-Farben

Dateiname: C:\Users\Pu...efahrt1.png
Dateigröße: 1.2 GB
Dateityp: PNG-Bild

Größe im Speicher: 1.0 MB
Schritte rückgängig: Keine
Schritte wiederholen: Keine

Pixelanzahl: 85263
Ebenenanzahl: 1
Kanalanzahl: 0
Pfadanzahl: 0

Bildeigenschaften

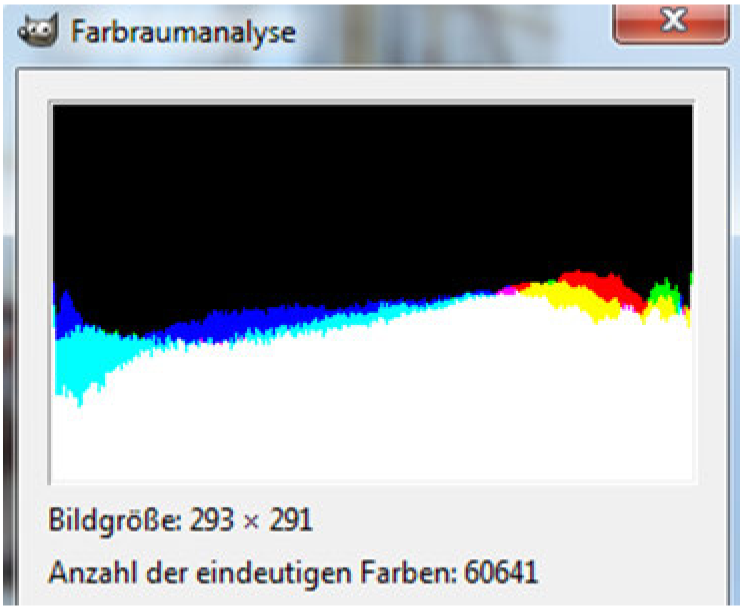
Bildeigenschaften
Foto_Seefahrt1.png-6

sRGB IEC61966-2.1
sRGB IEC61966-2.1

Copyright (c) 1998 Hewlett-Packard Company

WhitePoint : D65 (daylight)

Figure 156: Image properties of original image

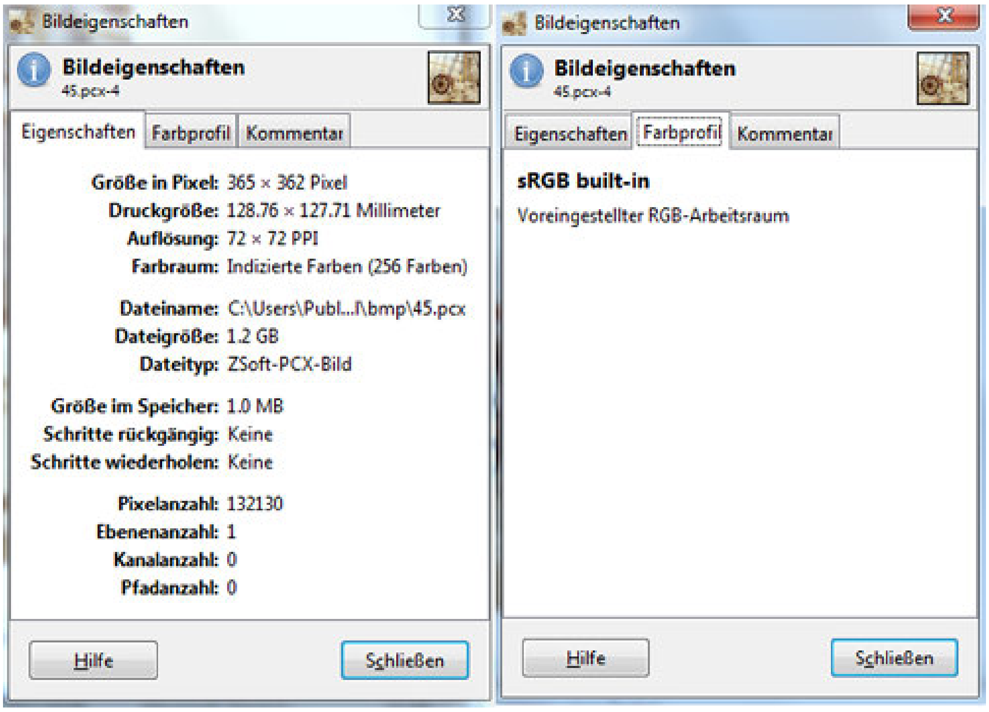


Bildgröße: 293 × 291

Anzahl der eindeutigen Farben: 60641

Figure 157: The original image's color space

Image properties after importing the image – optimized color palette with 256 colors in PNG format



Größe in Pixel:	365 × 362 Pixel
Druckgröße:	128.76 × 127.71 Millimeter
Auflösung:	72 × 72 PPI
Farbraum:	Indizierte Farben (256 Farben)
Dateiname:	C:\Users\Publ...l\bmp\45.pcx
Dateigröße:	1.2 GB
Dateityp:	ZSoft-PCX-Bild
Größe im Speicher:	1.0 MB
Schritte rückgängig:	Keine
Schritte wiederholen:	Keine
Pixelanzahl:	132130
Ebenenanzahl:	1
Kanalanzahl:	0
Pfadanzahl:	0

sRGB built-in

Voreingestellter RGB-Arbeitsraum

Figure 158: Image properties of imported image

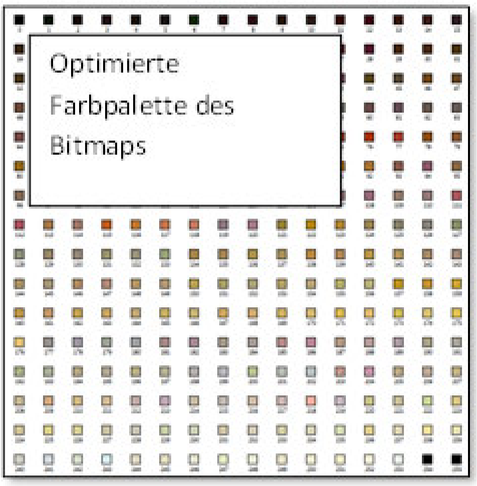


Figure 159: Color space analysis for original image

This generated color palette will be loaded onto the panel together with the bitmap.

Optimized color palettes for images

The following two images and their color palettes show how each of the two color palettes has been custom-generated for its respective image using the characteristic colors from that image's color space.

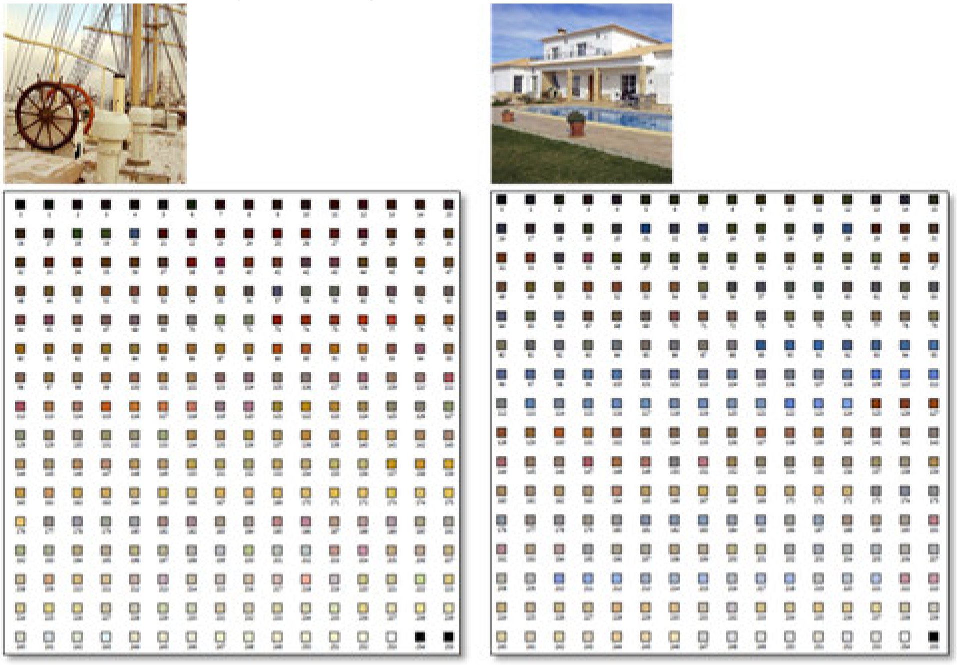


Figure 160: Optimized color palette

Limits

touch panels can display a maximum of 65536 colors. This means that, when put together, all optimized image color palettes and the color palette for objects for which colors can be selected must not have more than 65536 different colors total. For example, if all images were to have 256 different colors in their color palettes, you would be able to display a maximum of 255 images with their optimized colors (65536 / 256 – 1 × color palette for objects for which colors can be selected).

Summary

- There are 256 colors available for drawing objects and for selecting colors for objects. If necessary, you can customize this color palette or add new colors to it.
- For each image, a separate color palette with 256 colors optimized for the image will be created and transferred to the panel together with the corresponding bitmap. This ensures that the color characteristics for each image will be preserved (included in the image header chunk for the .png format).
- Even if you use multiple images on the same screen, each image will have its own optimized color palette

Color choice - Object's Color Palette

Colors can be selected in the Properties pane or in the Extended properties dialog box for the relevant object.

Properties\..color

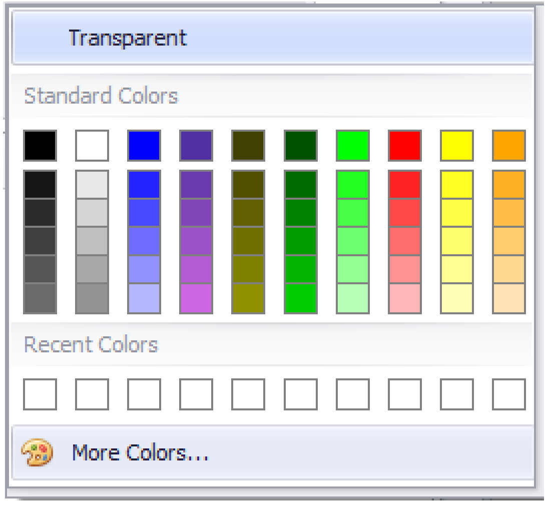


Figure 161: Color pane

Clicking on a color will automatically apply it to the selected object.
When you move the cursor over a color, the corresponding RGB value will be displayed.

RGB value being displayed when the cursor is moved over a color

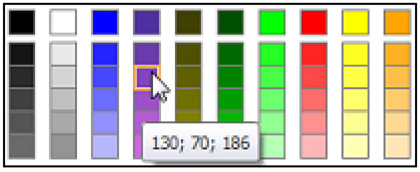
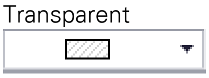
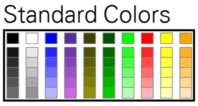


Figure 162: Properties\..color\Standard Colors

Opening the color pane will provide the following options:



No fill or line color
Transparency will be indicated with gray hatching in the background.



Groups together a series of basic colors with varying intensities that can be used to quickly choose a color.

Displays the last 10 colors that have been selected so that they can be chosen again for an object in the same project or in a different project. As long as GALILEO remains open, this quick selection palette will be available. Please note, however, that it will not be saved when the application is closed and that it cannot be saved as a separate palette.

Recent Colors

Opens the Select Color dialog box that can be used to select a color graphically or enter its hexadecimal value directly

Make sure to choose sufficiently contrasting colors when using color combinations, e.g., between screens and text. The apparent intensity of colors can be enhanced or softened with the use of contrasting colors.

More Colors...

There are two different device-dependent color models available when working with GALILEO.

Please note that not all device models will be able to exactly reproduce all the colors available.
If the number of colors is limited to 256 colors, i.e., in the case of graphics adapters with a color depth of 8 bits, you should use web-safe colors.
For more information on colors, color models, etc., you can consult the Internet or other resources.

In GALILEO, any color can be selected by entering its hexadecimal value into the # field.

Hexadecimal color value

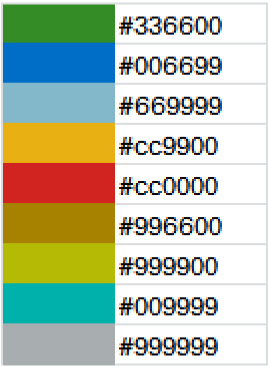
The hexadecimal value can be entered into the # field as RRGGBB,

where R, G, and B can have a value of 0, 3, 6, 9, C, F
and R stands for red, G for green, and B for blue.

Examples of hexadecimal values for various colors

The code for pure blue is 0000FF
The code for pure yellow is FFFF00
The code for medium gray is 999999

The code for bright gray is 333333

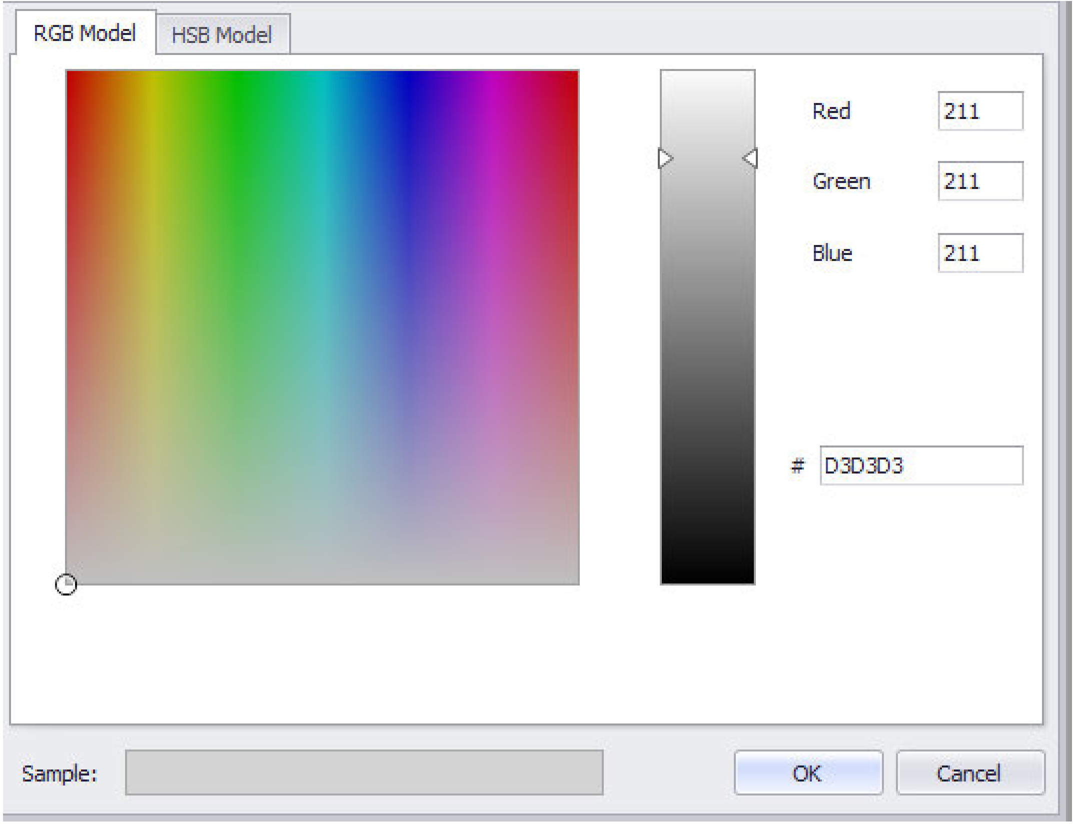


🟩	#336600
🟦	#006699
🟦	#669999
🟨	#cc9900
🟥	#cc0000
🟫	#996600
🟨	#999900
🟦	#009999
⬜	#999999

RGB model

The RGB model is an additive color mixing system.
It is usually used in all standard applications on a computer.

RGB Model tab in Select Color dialog box



Red 211
Green 211
Blue 211
D3D3D3
Sample:

Figure 163: Properties\...color\More Colors...

The three numeric values for the RGB model can be entered directly as numbers.
Valid input range: 0 to 255

You can also click on the color field to select a color.
The bar next to the color field can then be used to adjust the color's intensity, i.e., the higher the arrows on the bar, the brighter the color will be.
The color values will change based on the selection in the display. The RGB model

will directly convert the additive color mixing system color to the appropriate hexadecimal color value.

R
Red
The color's red component
Valid input range: 0 to 255

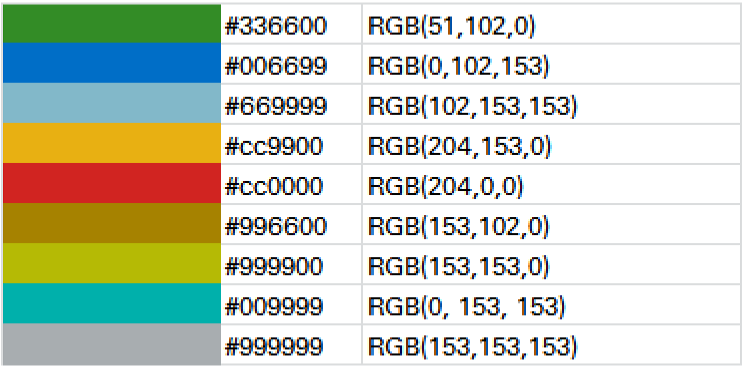
G
Green
The color's green component
Valid input range: 0 to 255

B
Blue
The color's blue component
Valid input range: 0 to 255

#
Shows the hexadecimal color value, and can be used to enter it, with the following notation: #RedGreenBlue

Example: Preview of selected colors

RGB color examples



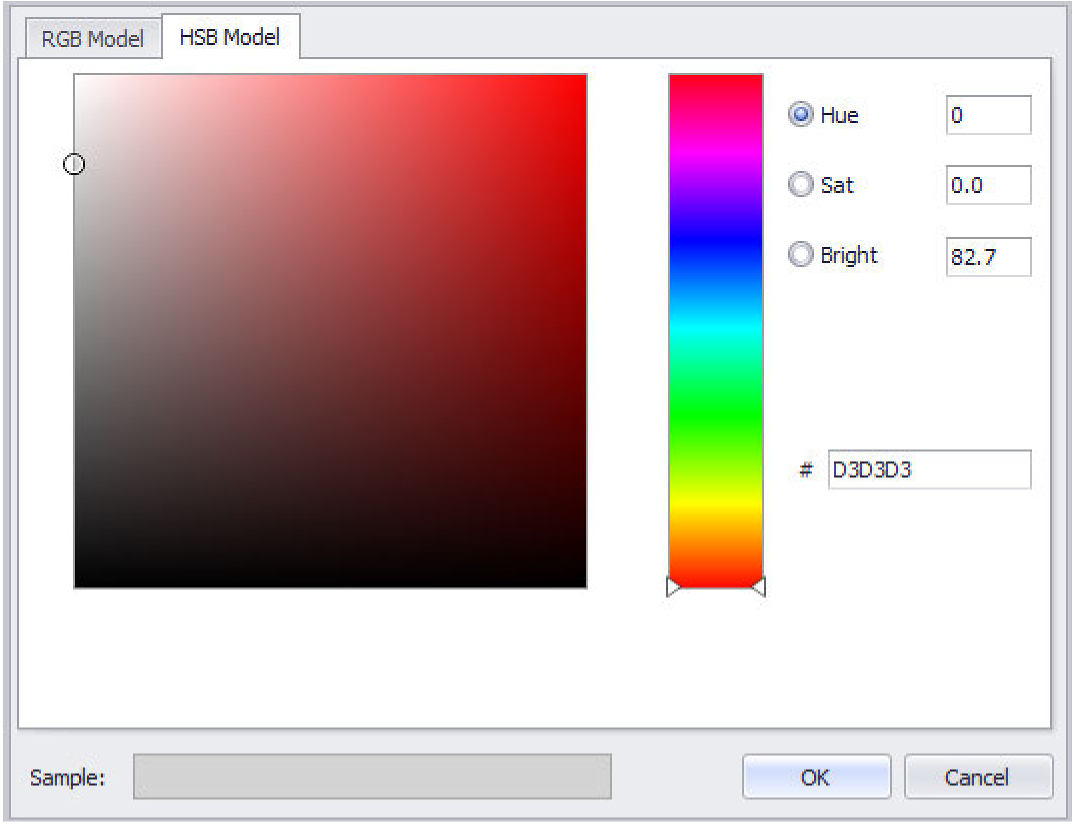
🟩	#336600	RGB(51,102,0)
🟦	#006699	RGB(0,102,153)
🟦	#669999	RGB(102,153,153)
🟨	#cc9900	RGB(204,153,0)
🟥	#cc0000	RGB(204,0,0)
🟫	#996600	RGB(153,102,0)
🟨	#999900	RGB(153,153,0)
🟦	#009999	RGB(0, 153, 153)
⬜	#999999	RGB(153,153,153)

HSB Model

The HSB model is based on RGB color spaces. In the HSB model, colors are described by means of three color properties: hue, saturation, and brightness.

→ Please note that the HSB model will not be available for images.

HSB Model tab in Select Color dialog box



● Hue 0

○ Sat 0.0

○ Bright 82.7

D3D3D3

Figure 164: Properties\...color\More Colors...

Any shade of color can be entered directly by using the numeric values for the hue, saturation, and brightness properties.

In addition, you can click on the bar to select a hue.
Meanwhile, the color field can be used to select the color's intensity, i.e., the various colors shown will be saturation and brightness variations of the same hue. More specifically, saturation increases from right to left and brightness decreases from top to bottom in the field.
The color values will change based on the selection in the display. The HSB model will directly convert the additive color mixing system color to the appropriate hexadecimal color value.

How brightness and saturation affect a color:
The more saturated and bright a color is, the more intense it will be

H
Hue

Hue is the color reflected or absorbed by an object. This hue is specified as an angle between 0 and 360 on a standard color circle.
Valid input range: 0 to 359

S
Saturation

Saturation is a color's intensity or purity. It describes a color's gray component relative to its hue, and is measured as a percentage between 0% (gray) and 100% (saturated).
Input value range as a %: 0 – 100

B
Brightness

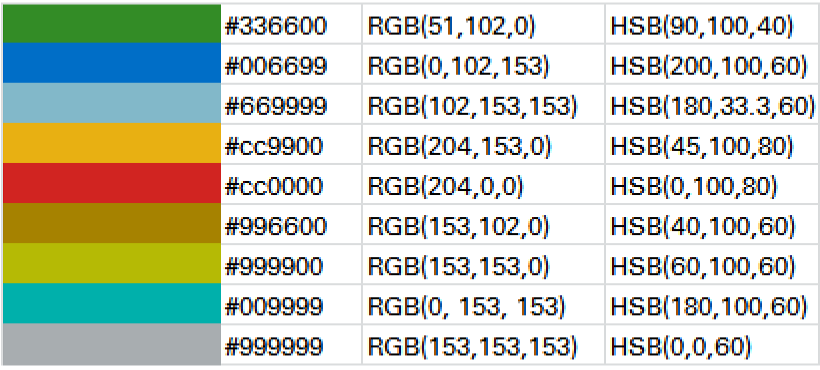
Brightness is a color's relative brightness and is measured as a percentage between 0% (black) and 100% (white).
Input value range as a %: 0 – 100

#

Shows the hexadecimal color value, and can be used to enter it, with the following notation: #RedGreenBlue

Example: Preview of selected colors

HSB color examples with the corresponding RGB values



🟩	#336600	RGB(51,102,0)	HSB(90,100,40)
🟦	#006699	RGB(0,102,153)	HSB(200,100,60)
🟦	#669999	RGB(102,153,153)	HSB(180,33.3,60)
🟨	#cc9900	RGB(204,153,0)	HSB(45,100,80)
🟥	#cc0000	RGB(204,0,0)	HSB(0,100,80)
🟫	#996600	RGB(153,102,0)	HSB(40,100,60)
🟨	#999900	RGB(153,153,0)	HSB(60,100,60)
🟦	#009999	RGB(0, 153, 153)	HSB(180,100,60)
⬜	#999999	RGB(153,153,153)	HSB(0,0,60)

1.6.5.8 Basic control properties

Special properties can be assigned to any visualization object available in GALILEO. This means that each object can be configured in a detailed custom manner for the application at hand.

The basic properties and extended properties that can be adjusted for each visualization object are described below.

Please note that not all properties will be available for every visualization object.

For further details regarding special properties, please refer to the description of the properties for the relevant visualization object.

- → section "Position/Size", page 242
- → section "Defining text series", page 251
- ✓ → section "Accessibility", page 258
- → section "Visibility", page 259

The color selection options have already been described within the context of the visualization on the device → section "Colors in GALILEO", page 230.

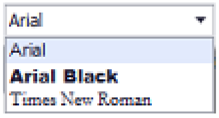
Labels in GALILEO

In GALILEO, a label is added to certain visualization objects.
In order to make it possible to get a consistent appearance for your application, GALILEO will assist you by offering the same selection options listed here on the visualization object.

Font attributes

Font

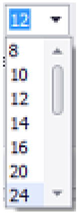
With the exception of the <Arial> default font, the fonts available will be those that have been added in *Home\Languages\Project Languages*.



Arial
Arial
Arial Black
Times New Roman

Font Size

The font size can be left with its default value or can be manually changed to a value within a range of 1 to 999 pt. The line height will be automatically adjusted. In addition, the text in the inserted text box will automatically be broken as required for the object's size if the automatic word wrap option has been enabled.



12
8
10
12
14
16
20
24

Text color

You can change the color if you want. The following options are available for this purpose:

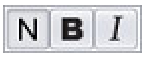
- Transparent
- Standard Colors
- More Colors...

The most recently used colors will be shown (user-specific).

For detailed information on the available colors, please refer to the following section: → section "Color choice - Object's Color Palette", page 234

Font style N

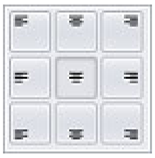
The following formatting options are available:



- Default (**N**ormal)
- **Bold** (**B**old)
- *Italic (**I**talic)*

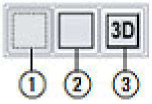
Alignment ≡

The buttons shown below can be used to configure the text to be left-aligned, centered, or right-aligned vertically and horizontally.



Frame 3D

The visualization object can be displayed with any one of three different designs.



① no frame

② with frame

③ 3D frame

Background

You can change the color if you want. The following options are available for this purpose:

- Transparent
- Standard Colors
- More Colors...

The most recently used colors will be shown (user-specific).

For detailed information on the available colors, please refer to the following section: → section "Color choice - Object's Color Palette", page 234

Word Wrap

If this checkbox is enabled, the text will automatically be broken as required for the visualization object's size. Otherwise, the text will remain on a single line, and only the portion that fits in the visualization object's width will be displayed.

Blinking Text

If this checkbox is enabled, the visualization object's text will flash. This flashing effect will be achieved by alternately hiding and showing the text in front of the selected background.

Flashing

If this checkbox is enabled, the visualization object's background will flash. This flashing effect will be achieved either by switching between the configured background color and black or by inverting the selected icon's colors.

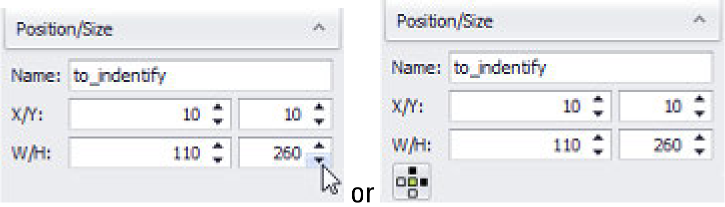
This attribute will not be available for objects of type "parameter list" or "recipe."

Position/Size

In GALILEO, every object's size and position on the screen can be changed.

The only exceptions in terms of entering absolute values are the → " Line", page 471 and → " Polygon", page 476 static objects used to design screens.

The position and size can be used to position the visualization object on the device screen and define the size it should have when displayed.



Position/Size

Name: to_indentify

X/Y: 10 10

W/H: 110 260

or

Position/Size

Name: to_indentify

X/Y: 10 10

W/H: 110 260

Figure 165: Position/Size properties pane

Name

To be able to distinguish between multiple objects of the same type on a single screen in the Controls Manager or when used in scripts, you should assign names to the objects. The names you enter will appear and be editable both here and in the Controls Manager.

X/Y

The units are pixels. Enter the X/Y axis position either as a value or as a tag. If you enter a value, the value will be interpreted as being absolute.

The context menu of the right mouse button allows you to switch absolute value 12 Use Value to Use Tag.

Absolute

Within this context, "absolute" means that the value defines the distance between the screen's origin in the upper left corner and the visualization object's origin in the object's upper left corner.

However, the X/Y axis position can also be defined using a tag so that it can change at runtime. The following section explains how to do this in detail. When using tags, you will be able to choose whether the position should be interpreted as being absolute or relative. The default is "absolute."

Relative

Within this context, "relative" means that the tag's current value defines the distance between the upper left corner of the visualization object in its starting position and in its current position. The starting position will be the tag's default value.

To select whether the position should be interpreted as being absolute or relative, go to *Home\Settings\Runtime tab\Dynamic object size and position mode*. The setting you select will apply to all visualization objects in the project.

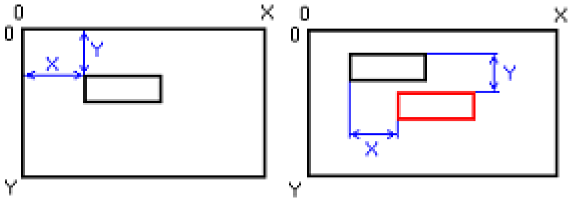


Figure 166: Absolute and relative positions – for the relative position, the visualization object's starting position is shown in black and the relative position in red

W/H

The units are pixels. Enter the width/height either as a value or as a tag. Please note that the width/height can only be entered as a value for the following visualization objects:

- Parameter list
- Error Window
- Recipe

If you enter a value, the value will be interpreted as being absolute.

Absolute

Within this context, "absolute" means that the value defines the width or height of the visualization object.

However, the width/height can also be defined using a tag so that it can change at runtime. The following section explains how to do this in detail. When using tags, you will be able to choose whether the width/height should be interpreted as being absolute or relative. The default is "absolute."

Relative

Within this context, "relative" means that the tag's current value defines the change in width or height relative to the starting size. The starting size will be the tag's default value.

To select whether the position should be interpreted as being absolute or relative, go to *Home\Settings\Runtime tab\Dynamic object size and position mode*. The setting you select will apply to all visualization objects in the project.

Size modifications

There are two ways to define the size change for an object:

- Anchor points on the visualization object itself
- Resize mode for user controls

Anchor points can be defined as a property under Position/Size for the object.

Meanwhile, resize mode for user controls can be configured in the General properties for the screen where the user control is being used. Please refer to → section "Resize mode", page 269.

Anchor points

Anchor points define which points/edges in a visualization object should remain at a fixed, relative position if the size of the user control in which the object is being used is changed as a whole.

▶ Enable the individual top, bottom, right, and left anchors by clicking on them. Enabled anchors will be filled out in black.

▶ Clicking on an anchor again will disable it.

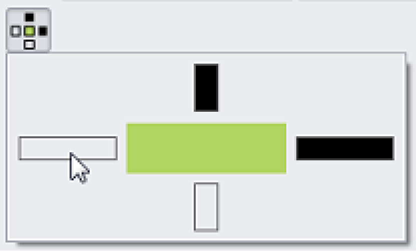


Figure 167: Top and right anchors enabled for the visualization object

The → " Line", page 471 and → " Polygon", page 476 static objects used to design screens cannot be anchored.
Moreover, groups of objects cannot be anchored either.

You can use any combination of the top, bottom, right, and left anchors.

If you enable the left anchor,
the distance between the left edge of the user control and the individual visualization object will not change.
This applies analogously to the top, bottom, and right anchors.

If opposite anchors such as left / right and top / bottom are enabled,
the object in the user control will be stretched or shrunk in the corresponding direction.

If no anchor points are enabled,

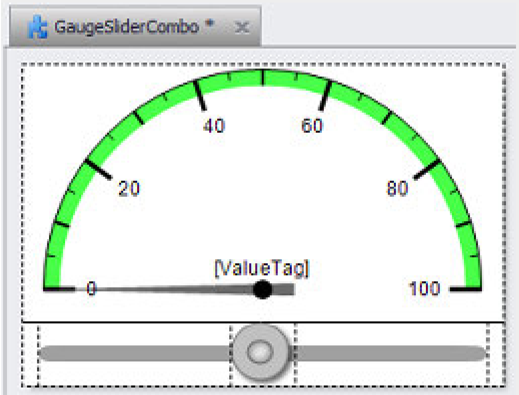
the size of the object will not change within the user control that has been resized.

The ✛ icon in the properties for the individual visualization object directly shows the anchors that are currently enabled.

When a visualization object is copied, the enabled anchors will be copied as well.

Anchor point example

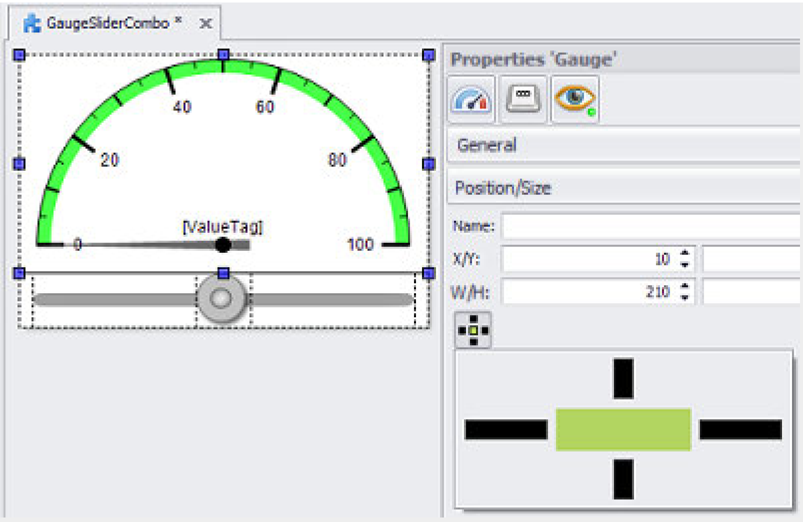
Say that a user control called "GaugeSliderCombo" has been created. This user control contains both a gauge and a slider, with the combination filling out all the space in the user control.



Now say that you want the gauge's height and width to be adjusted in the event that the user control is resized, but that you also want for the slider to be adjusted only in terms of width. In other words, say that you want the slider's height to remain unchanged regardless of the user control's size.

In order to set this up, you would configure the anchor points as follows:

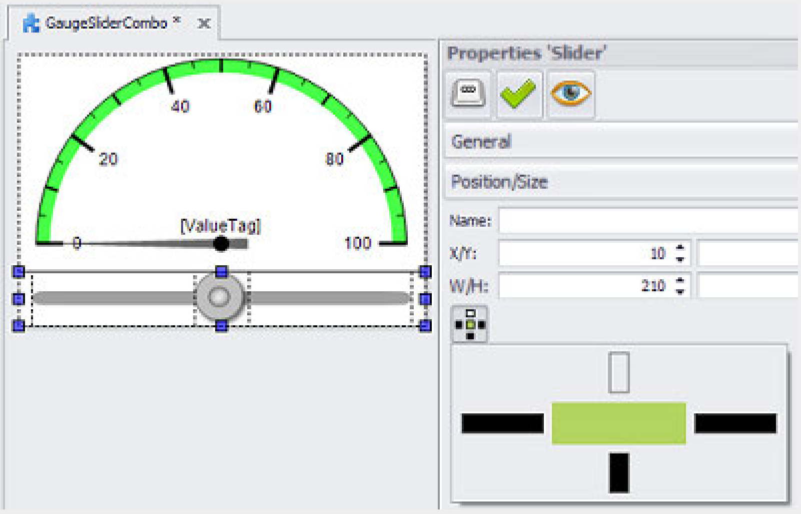
For the "gauge" visualization object:
You would enable all the anchor points for the "gauge" object, meaning that the distance on all four sides (top, bottom, left, right) from the gauge in the user control would remain the same after the user control's size changes. In this case, this distance would be 0 pixels, since the entire user control would remain filled out.



Name:
X/Y: 10
W/H: 210

Figure 168: Anchors for "gauge" visualization object

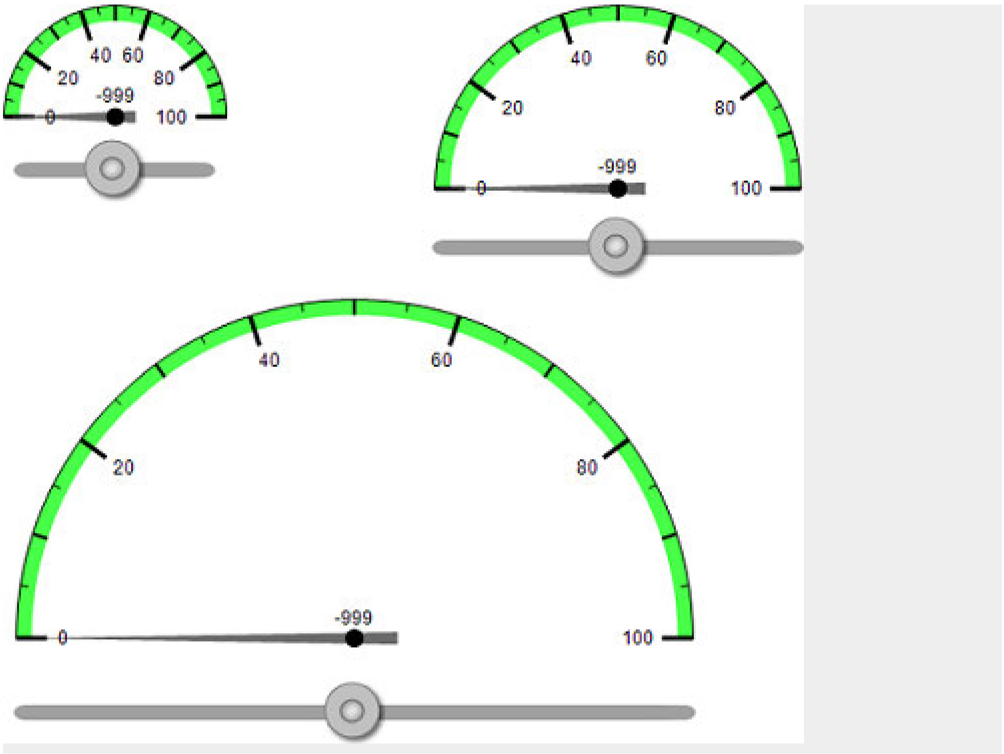
For the "slider" visualization object:
You would enable the left, right, and bottom anchor points, meaning that the distance on the left, right, and bottom from the gauge in the user control would remain the same after the user control's size changes. In this case, the distance from the top in the user control would be adjusted relative to the user control's size change.



Name:
X/Y: 10
W/H: 210

Figure 169: Anchors for "slider" visualization object

If you were to then use the "GaugeSliderCombo" user control, you would be able to resize it, and the gauge and slider contents would change as per the anchors you enabled previously:


0
20
40
60
80
100
-999
0
20
40
60
80
100
-999
0
20
40
60
80
100
-999

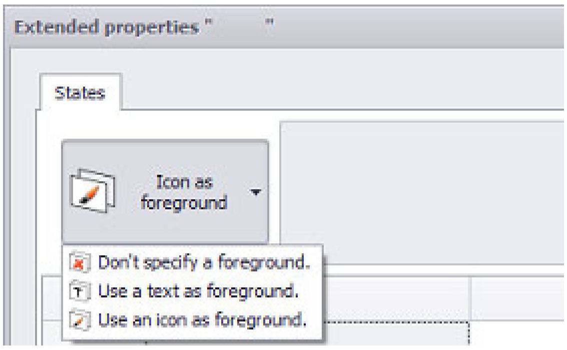
States

The States tab in the Extended properties dialog box can be used to define what the visualization object should look like in its various states.

Each visualization object will have its own specific states and corresponding options as applicable, but the one thing all objects share in common is that a foreground and a background can be designed for each available state.

Depending on the specific visualization object, not all parameters may be available. For further details, please refer to the description for the relevant visualization object.

→ Remember to select the state or states before defining the properties for it / them.


Extended properties "
States
Icon as foreground
Don't specify a foreground.
Use a text as foreground.
Use an icon as foreground.

Don't specify a foreground

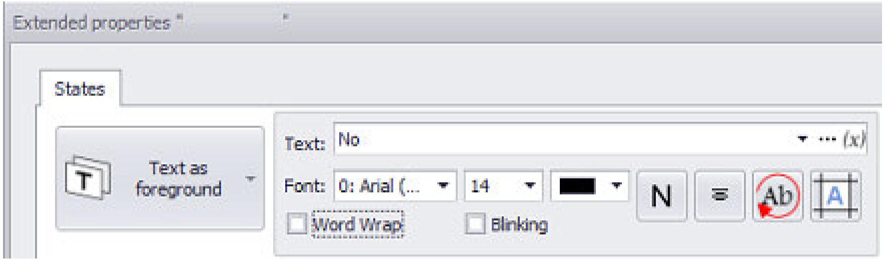
Can be used to optimize the project and reduce the amount of memory required

If a special text or icon design is not required for the object's foreground, such as when an image will be shown as the background, this setting ensures that no unnecessary data will be saved.

→ Please note that this attribute is not available for all objects.

Text as foreground

Controls\<Visualization object>\Properties\



Text: No

Font: 0: Arial (... 14

□ Word Wrap □ Blinking

Text: Used to enter the text that should be shown in the corresponding visualization object state.
Clicking on the drop-down menu arrow will show the texts available for selection (all the existing standard texts in the text list).

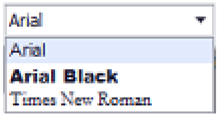
This Text setting is not found in "'Value Display/Entry" visualization objects, since the text in this case is equal to the tag value.

Alternatively, you can simply enter the text you want directly.
When you do this, GALILEO will use its advanced text management functionalities to provide text suggestions.
The new text will be added to the text list and will become available for future selection.

For more information on text management, please refer to the → "Languages and texts", page 630 section.

Font:

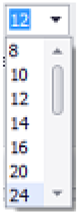
With the exception of the <Arial> default font, the fonts available will be those that have been added in *Home\Languages\Project Languages*.



Arial
Arial
Arial Black
Times New Roman

Font Size

The font size can be left with its default value or can be manually changed to a value within a range of 1 to 999 pt.
The line height will be automatically adjusted.
In addition, the text in the inserted text box will automatically be broken as required for the object's size if the word wrap option has been enabled ☑ .



12
8
10
12
14
16
20
24

Text color

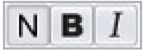
You can change the color if you want. The following options are available for this purpose:

- Transparent
- Standard Colors
- More Colors...

The most recently used colors will be shown (user-specific).
For detailed information on the available colors, please refer to the following section:
→ section "Color choice - Object's Color Palette", page 234

N
Font Style

The following formatting options are available:

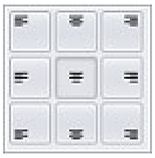


N B *I*

- Default (**N**ormal)
- **Bold** (**B**old)
- *Italic (**I**talic)*

Alignment

The buttons shown below can be used to configure the text to be left-aligned, centered, or right-aligned vertically and horizontally.



Text Rotation

The following alignments are available:

- Standard

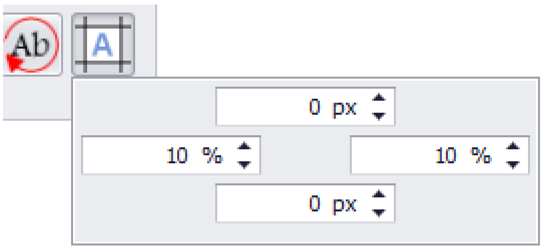
and turned clockwise

- 90°
- 180°
- 270°

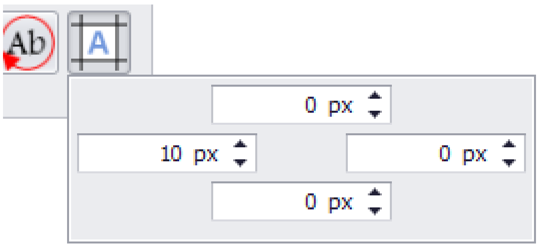
Distance from edge

Can be used to specify the distance from the visualization object edges that should be maintained.

The unit can either be pixels or a percentage (relative to the size of the object). To select the latter, click on the unit. px The mode will change to "%".

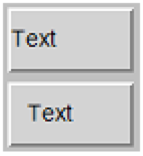


0 px
10 % 10 %
0 px



0 px
10 px 0 px
0 px

You can use these settings, for example, to define a left distance of 10 px for a left-aligned text so that the text will be indented 10 px on the object.



Text

Text

Word Wrap

If this checkbox is enabled ☑, the text will automatically be broken as required for the visualization object's size.

Otherwise □, the text will remain on a single line, and only the portion that fits in the visualization object's width will be displayed.

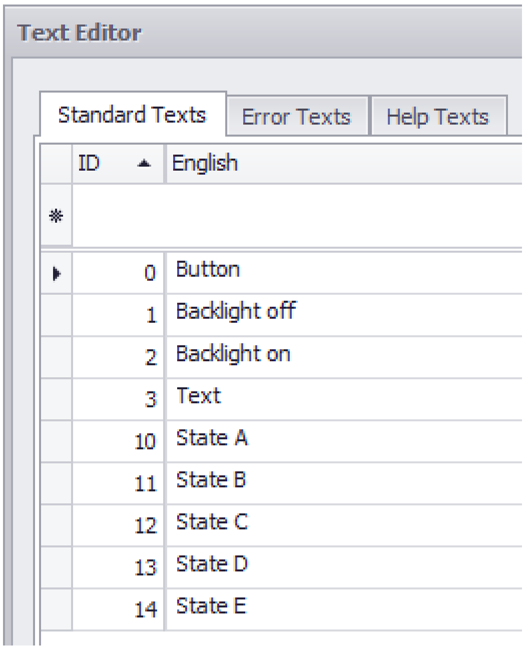
Blinking

If this checkbox is enabled ☑, the visualization object's text will flash.

This flashing effect will be achieved by alternately hiding and showing the text in front of the selected background.

1.6.5.9 Defining text series

Texts in GALILEO are identified with unique numbers. This numbering can also be used to group similar texts that belong together, as the following example illustrates:



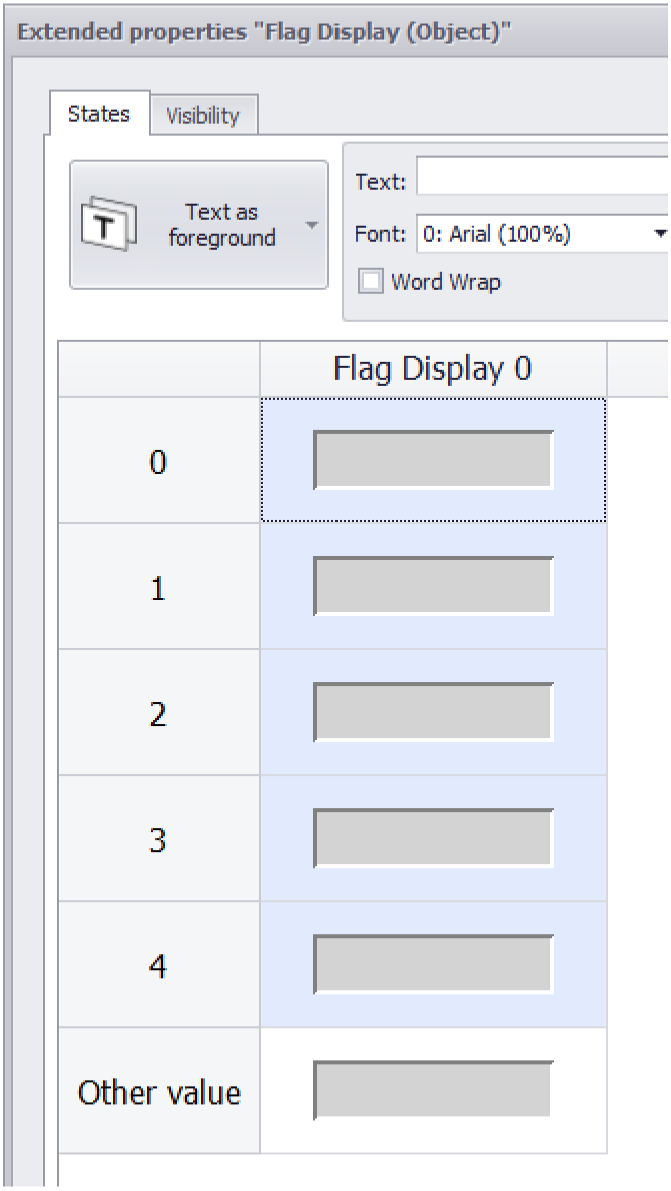
ID	English
0	Button
1	Backlight off
2	Backlight on
3	Text
10	State A
11	State B
12	State C
13	State D
14	State E

Texts State A through State E have consecutive numbers (10 – 14).

When configuring states, this can be used in order to assign text to multiple states more quickly.

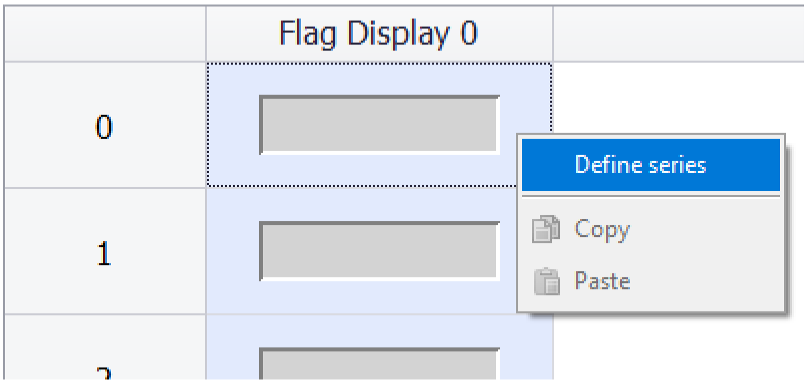
Follow the steps below:

1. Select the states you want with the mouse

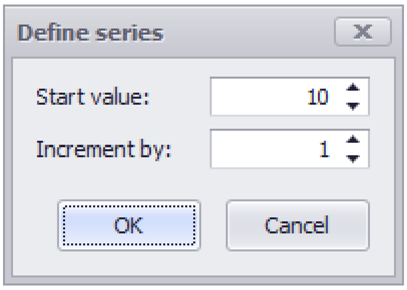


Extended properties "Flag Display (Object)"

2. Open the context menu and click on the Define series option:



3. Enter the text number of the first text to be used and the numeric value that should be incremented.



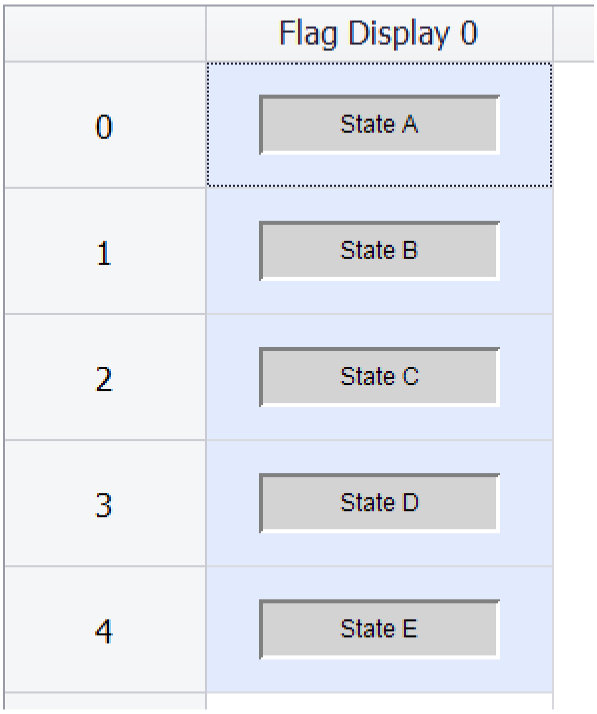
Define series

Start value: 10

Increment by: 1

In this example, the first text is the one with the number 10, then the one with the number 11, etc.

The texts will be assigned automatically:



	Flag Display 0
0	State A
1	State B
2	State C
3	State D
4	State E

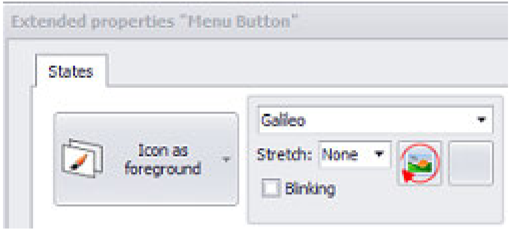
Icon as foreground

An icon can be added to most visualization objects – especially if it is the icon that will be displayed as a button on the device.

This significantly expands the visual possibilities available when designing a visualization object.

When an icon is added, you can choose to show the icon in the foreground instead of a text.

Controls\<visualization object>\General properties



Extended properties "Menu Button"

States

Icon as foreground

Galileo

Stretch: None

□ Blinking

<...> If you select Icon as foreground from the drop-down menu, you will need to select the icon you want from the drop-down menu to the right. Before this, make sure that the image is found in the project tree, in *Images\Project images*. If it is not there, it will not appear in the drop-down menu.

For detailed information on adding an image to the project tree, please refer to the "Your first project" section → section "Step 5: Adding an image", page 174.

Stretch:

- No
  The icon will be displayed with its original size and will not be resized to fit the button.

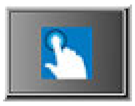


Figure 170: Visualization object with icon as foreground, <None> selected in the Stretch drop-down menu

- Uniform
  The icon will be centered and resized to fit the button. The aspect ratio will be maintained. If the image's aspect ratio does not match the button's aspect ratio, the selected background color will be shown in the remaining area.

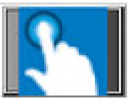


Figure 171: Visualization object with icon as foreground, <Uniform> selected in the Stretch drop-down menu

- Fill
  The icon will be resized to fit the button exactly. This means that it will be stretched if the button is larger than it and shrunk if the button is smaller than it. The aspect ratio will not be maintained.

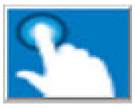


Figure 172: Visualization object with icon as foreground, <Fill> selected in the Stretch drop-down menu

Blinking If this checkbox is enabled, the visualization object icon will flash. This flashing effect will be achieved by inverting the selected icon's colors.

→ This property is not available for the parameter list, graph, alarm list, and recipe visualization objects.

Image The buttons shown above can be used to rotate the image 90°, 180°, or 270° as needed.

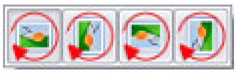
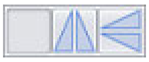


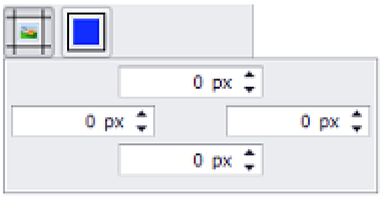
Image Flipping:

The buttons shown above can be used to flip the image.



Distance from edge

Can be used to specify the distance from the visualization object edges that should be maintained. This can be used, for example, to use an image with Uniform stretching without it going all the way to the edge of the visualization object.



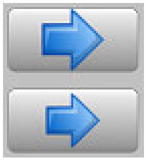
0 px

0 px

0 px

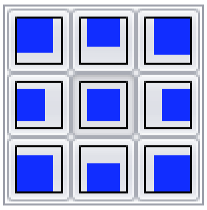
0 px

Example showing a button with an icon with Uniform stretching. The button on top does not have a distance from the edges, while the button on the bottom has a distance of 5 px from the top and bottom edges:

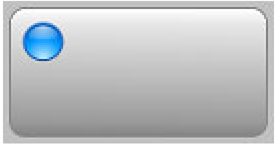


Alignment

Used to set the icon's alignment within the visualization object.



Example showing a button in which the icon is top-left-aligned and has a distance of 5 px from the edge:



Color as Background

You can change the color if you want. The following options are available for this purpose:

- Transparent
- Standard Colors
- More Colors...

The most recently used colors will be shown (user-specific).

For detailed information on the available colors, please refer to the following section:
→ section "Color choice - Object's Color Palette", page 234

Blinking

→ Please note that this attribute is not available for all objects.

If this checkbox is enabled, the visualization object's background will flash. This flashing effect will be achieved either by switching between the configured background color and black or by inverting the selected icon's colors.

Image as background

An image can be added as a background for the visualization object – especially if the visualization object will be displayed as a button on the device.
In this case, the image will be used instead of a color as a background.

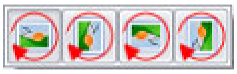
If you select Use an image as background from the drop-down menu, you will need to select the image you want from the drop-down menu to the right. Click on ▾ to select the background image you want. Before this, make sure that the image is found in the project tree, in *Media\Project images*. If it is not there, it will not appear in the drop-down menu.
For detailed information on adding an image to the project tree, please refer to the "Your first project" section → section "Step 5: Adding an image", page 174.

Blinking

→ Please note that this attribute is not available for all objects.

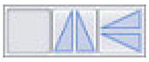
If this checkbox is enabled, the image in the visualization object's background will flash. This flashing effect will be achieved by inverting the selected icon's colors.

Rotate



The buttons shown above can be used to rotate the image 90°, 180°, or 270° as needed.

Flip



The buttons shown above can be used to flip the image.

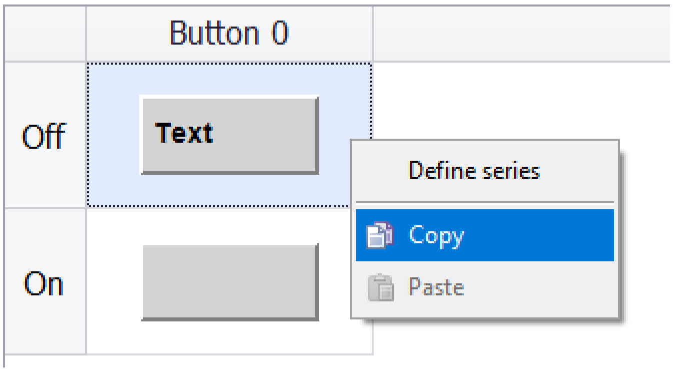
1.6.5.10 Copying/pasting states

You can copy individual configured states and paste them together with all settings to another state.

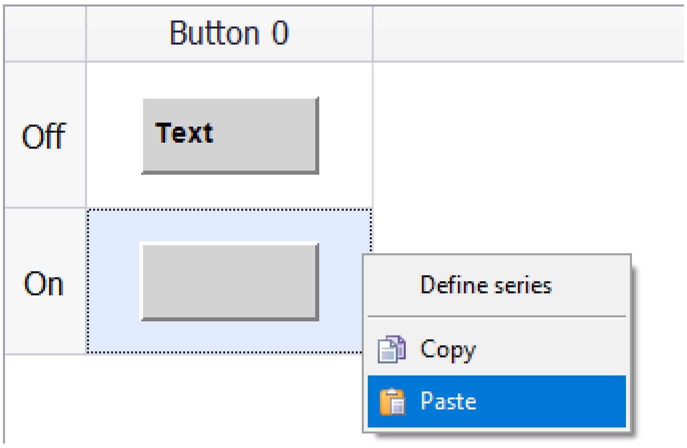
Follow the steps below:

1. Select the state you want to copy in the table and either click on the "Copy" option in the context menu

or press Ctrl + C



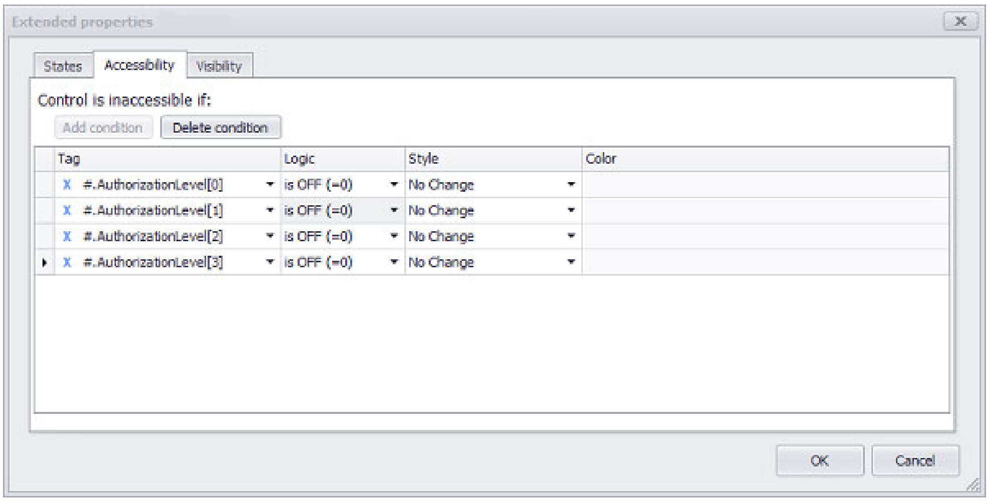
2. Select one or more states in the table and either click on the "Paste" option in the context menu or press Ctrl + V



Accessibility

The ✓ Accessibility tab can be used to define the conditions under which the visualization object will not be accessible. A maximum of four conditions can be entered.
These conditions will be OR'd, meaning that the visualization object will STOP being usable as soon as any of them are TRUE.
The conditions are linked to tags of data type BIT.
Within this context, "not usable" means that it will not be possible to move or modify the object even if it is visible.

Controls\<Dynamic Controls>\Properties\✓



Control is inaccessible if:

Tag	Logic	Style	Color
X #.AuthorizationLevel[0]	is OFF (=0)	No Change	
X #.AuthorizationLevel[1]	is OFF (=0)	No Change	
X #.AuthorizationLevel[2]	is OFF (=0)	No Change	
X #.AuthorizationLevel[3]	is OFF (=0)	No Change	

Figure 173: Example in which four bit tag states are used to define whether a visualization object will be accessible

Visualization object accessibility example

In the example above, the visualization object will be usable if:

AuthorizationLevel[0] =1; AuthorizationLevel[1] =1;
AuthorizationLevel[2] =1; AuthorizationLevel[3] =1;

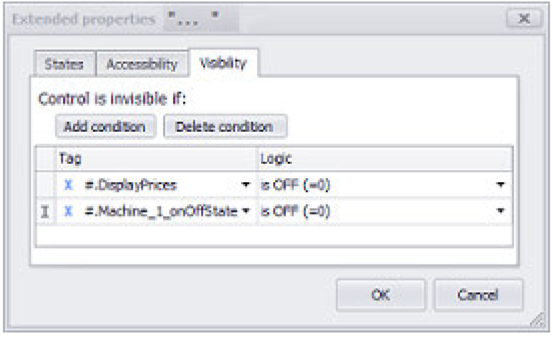
The object will not be usable if:

AuthorizationLevel[0] =0; AuthorizationLevel[1] =1;
AuthorizationLevel[2] =1; AuthorizationLevel[3] =1;

Visibility

The Visibility tab can be used to define the conditions under which the visualization object should be **invisible**.
A maximum of four conditions can be entered.
These conditions will be OR'd, meaning that the object will be **invisible** as soon as any of them are TRUE.
The conditions are linked to tags of data type BIT.

Controls\Controls\Properties\



Control is invisible if:

Tag	Logic
#.DisplayPrices	is OFF (=0)
#.Machine_1_onOffState	is OFF (=0)

Figure 174: Example in which three bit tag states are used to define whether a visualization object will be visible

Visualization object visibility example

In the example above, the visualization object will be displayed if:
PreiseAnzeigen = 1;
Machine_1_onOffState = 1;
The visualization object will not be displayed if:
PreiseAnzeigen = 1;
Machine_1_onOffState = 0;
or if:
PreiseAnzeigen = 0;
Machine_1_onOffState = 1;

If the visualization object is in a user control, you will also be able to use input parameters.

2. Screens and User Controls

2.1 Screens: The application's building blocksGALILEO

GALILEO has the following screen types, which differ from each other in terms of their functionality.

The Screen Graph dialog box shows how the individual screens are laid out in the project, as well as potential navigation paths, in an easily understandable manner.

Standard screens

Standard screens are the screens used as the basis for the application. They are used for user input and for entering and outputting PLC data with visualization objects. These screens can contain any visualization object, as well as User Controls.

A background screen can be assigned to every standard screen.

User controls

UserControls are ideal for grouping visualization objects that are constantly reused throughout a project at several locations or on several screens, e.g., headers, footers, and recurring object groups.
If necessary, UserControls can be used multiple times in all types of screens.

Dialog screens - special user controls

Dialog screens are used for prompts requiring user feedback. This type of screen is placed over the application, and interacting with the application may not be possible until the user closes the dialog box.
Dialog screens need to be created as a user control and then assigned to the dialog button visualization object.

Help screens

Help screens are special screens that provide device operators with helpful information in certain situations. These screens are used to set up a help management system. Like with keyboard screens, only certain types of objects can be used on these screens.

Keyboards

In order for a device operator to be able to enter information on the device, a keyboard screen needs to be displayed. Keyboard screens are special screens, and are the screens used to create custom keyboards. This also means that only certain types of objects are allowed on keyboard screens, as their purpose is to make it possible to implement a keyboard. This means that keyboard screens only allow for the use of static visualization objects, User Controls, and functions from the

Characters and Stream Characters function groups. There are also ready-to-use keyboard screens, which are referred to as "system keyboards," that are included with the system, e.g., Numeric, Numeric Min/Max, etc. These keyboards are found in the Keyboards group and can be accessed with a function.

Print reports

Print reports are screens that are only used for printouts. Print reports are compatible with all visualization objects except recipes, parameter lists, alarm lists, and help buttons. In addition, they will only show graph visualization objects correctly if recording and plotting are stopped for the corresponding graph block. The width and height of these screens can be freely adjusted. Printouts are printed in Graphic mode in the background, meaning that these screens are not visible on the device.

Background Screens

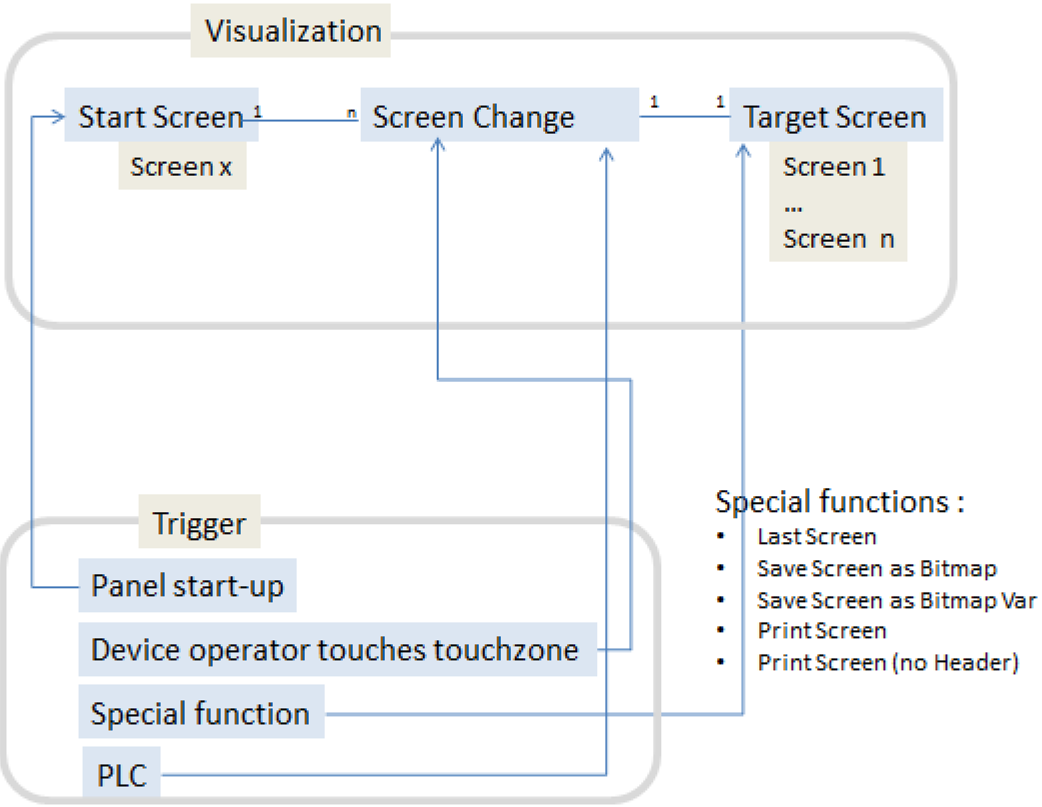
These screens can be used to create a standardized background for all screens or for several screens.

Screen Change

In order to make it possible to switch between standard screens, a screen change visualization object needs to be added to them. When a device operator taps on the corresponding button on the device screen at runtime, the system will switch between screens (screen changes can also be triggered directly by the corresponding PLC). The → section "Step 8: Adding screen changes", page 189 section describes a screen change example.

Setting a screen as the start screen

The start screen, also referred to as the startup screen, is the screen ▶ that will first appear on the device after the application is started. Any standard screen can be set to be the start screen by using the corresponding context menu option. However, please note that each project can only have one start screen.



Screens have a variety of properties that can be configured, with these properties depending on the specific type of screen.

2.2 Screen types

The project tree contains the following five types of screens. Each type has different properties. For an overview of the various screen types, please refer to → section "Screens and User Controls", page 260.

GALILEO features the following types of screens:

- Standard screens
- User Controls
  - Dialog screens
- Help screens
- Keyboards
- Printreport
- Background screens

The basic settings for configuring screens, such as grid settings and object snapping settings, can be configured in *Home\Settings\Screen View tab*, please refer to → section "Screen View", page 212.

To open a screen, double-click on it in the *Project Tree\Screens* tab. To select the screen, click on an unoccupied area within the screen. When you do this, the screen's properties will be shown.

The Properties pane will show the available configuration options for every screen and every visualization object placed on the screen, provided they are selected first.

Dialog screens are special user controls.

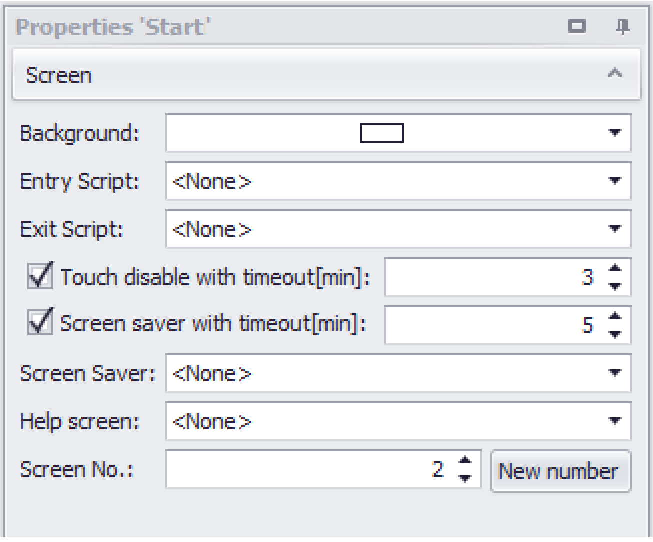
Prerequisites

To be able to follow the descriptions below on the program, click on a screen in *Project Tree\Screens* in order to open the Properties pane.

2.2.1 Standard screen

are used to display PLC data. You can place any type of visualization object on a standard screen. Moreover, standard screens can contain User Controls. The properties for a standard screen will remain the same regardless of which visualization objects are placed on it.

Project Tree\Screens folder\Standard screens\"Start"



Properties 'Start'

Screen

Background: ▭

Entry Script: <None>

Exit Script: <None>

☑ Touch disable with timeout[min]: 3

☑ Screen saver with timeout[min]: 5

Screen Saver: <None>

Help screen: <None>

Screen No.: 2 New number

Figure 175: Properties pane for the "Start" standard screen

Background color

You can change the color if you want. The following options are available for this purpose:

- Standard Colors
- More Colors...

The most recently used colors will be shown (user-specific).

For detailed information on the available colors, please refer to the following section: → section "Color choice - Object's Color Palette", page 234

Entry Script / Exit Script

Entry Script

This function is used to run an event script when the screen is opened (before the screen is displayed).
The event script specified in the Entry Script field will be run before the screen is opened.

Exit Script

This function is used to run an event script when the screen is exited.
The event script specified in the Exit Script field will be run before the screen is closed.

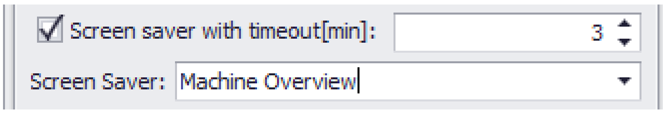
For a description of event scripts, please refer to the → section " Scripts", page 952 section.

Touch disable with Timeout [min]

The touch zone can be disabled. This field can be used to enter the time after which it will be disabled.

You can configure the number of minutes without any touch zone input after which the touch zones on the device screen will switch to standby mode. While these touch zones are on standby mode, a small image will recurringly appear and disappear briefly at random points on the device screen. The touch function can be reactivated by touching the screen twice simultaneously with two fingers in a diagonal position (two coordinate pairs).

Screen saver with Timeout [min]



☑ Screen saver with timeout[min]: 3

Screen Saver: Machine Overview

This property is used to configure the number of minutes without any touch zone input after which the screen saver will be be turned on. This screen saver will reduce the backlight current and switch to the screen specified in the Screen Saver field. The reduction in display brightness will lengthen the backlight's lifespan significantly. It is recommended to enable the screen saver option for all screens. A function that can be called for this purpose is described in the → section "Backlight: Learn Dim Value", page 1109 section

The backlight brightness can also be reduced as a constant setting. There are various display setting functions available for this purpose, such as Brightness 40%, that are described in greater detail in a different section – please refer to → section "Internal functions (Internals)", page 1109

Screen saver

This drop-down menu will only be active if the Screen saver with timeout checkbox is enabled. The screen saver function will switch to the screen selected in the Screen Saver field.

Help screen

A help screen can be assigned to every screen. This help screen can in turn be accessed with a help button. For a description of this help button, please refer to the → Chapter "2 Help Button", page 428 section.

Mask number New number

When created in a project, every screen will automatically be assigned a screen number as a unique ID. A project cannot contain more than one screen with the same number. When assigning a screen number manually, you can assign any number of your choice.

These screen numbers become relevant when the PLC triggers a screen change or when the PLC needs to determine which screen is currently being shown by the device. The way in which the PLC can trigger a screen change is described in detail in a different section – please refer to → section "Controlling functions with the PLC", page 602.

If you click on New number, the next free screen number in the project will be automatically entered into the Screen No. field. Screen numbers freed up when a screen is deleted will also be taken into account when determining the next free screen number.

2.2.2 Background Screens

These screens are available in GALILEO Version 10.1 and higher. They can be used to create a standardized background for all or multiple standard screens, and in this way resemble a layout template.

The background screen will be shown when working with the standard screen. However, its visualization objects will not be selectable, making it easier to focus on the standard screen.

If you want a background screen to be automatically used for every new standard screen added, use the Background Screens context menu to set it as the default background screen.

Changes made to background screens will automatically affect the appearance of all the standard screens to which they are linked.

You can create multiple background screens if necessary.

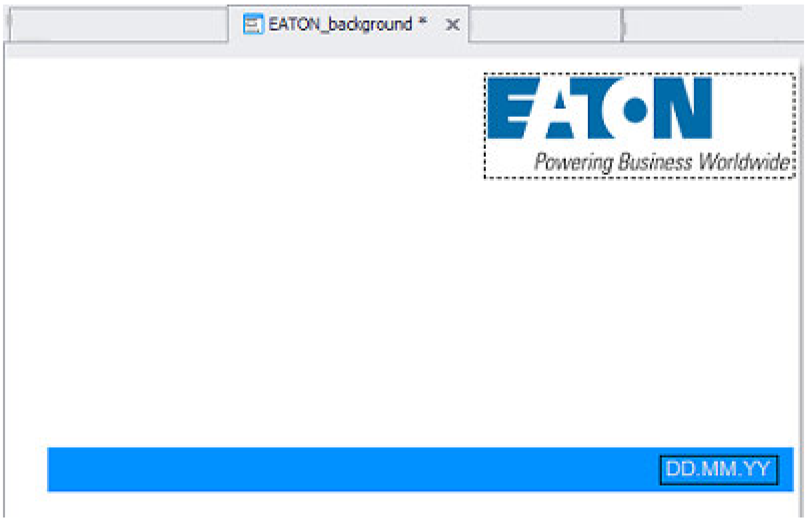
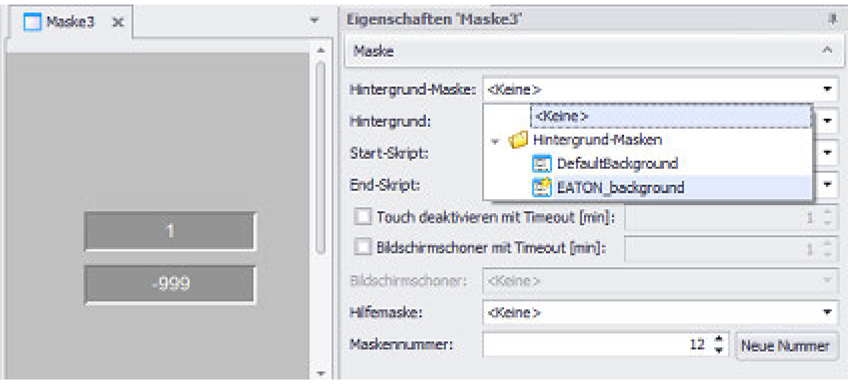


Figure 176: Background screen <EATON_background>



Eigenschaften 'Maske3'

Maske

Hintergrund-Maske: <Keine>

Hintergrund:

Start-Skript:

End-Skript:

<Keine>
Hintergrund-Masken
DefaultBackground
EATON_background

□ Touch deaktivieren mit Timeout [min]: 1

□ Bildschirmschoner mit Timeout [min]: 1

Bildschirmschoner: <Keine>

Hilfemaske: <Keine>

Maskennummer: 12 Neue Nummer

Figure 177: Standard screen <Screen3> is assigned <EATON_background> as its background screen

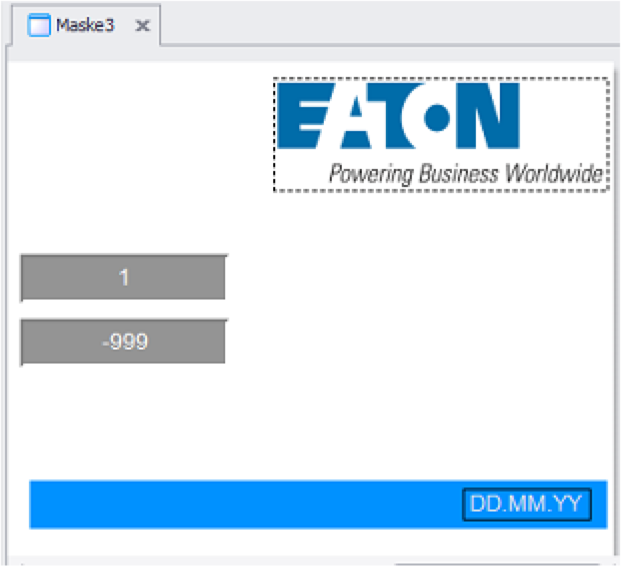


Figure 178: Standard screen <Sreen3> with background screen <EATON background>

2.2.3 UserControls

These screens are ideal for grouping objects that are constantly reused throughout a project at several locations, e.g., headers, footers, recurring visualization object groups, and dialog boxes.

- UserControls can contain any type of object.
- User controls can be used in all other screen types.
- In order to be able to use a UserControl in another screen, the UserControl first needs to be created in *Project Tree\Screens\UserControl*.
- To use a UserControl in another screen, the other screen needs to be opened, after which the UserControl needs to be dragged from *Project Tree\Screens\UserControl* into the screen that is open.
- As soon as a user control is selected and dragged onto the current screen, the cursor will change so that the contents of the user control being positioned will be shown.

A user control can be placed inside another user control in order to nest them; please refer to → section "Nesting user controls", page 288.

However, a container cannot be used inside a user control, since this will prevent certain object properties (e.g., touch) from being managed in the runtime environment.

A container can contain multiple user controls. For a description of this, please refer to the → "Using containers", page 294 section

The user control that was selected last will become the active element in the ribbon bar, and its name will be shown underneath this element. If you click on the user control icon without opening the drop-down menu, an instance of the user control that was selected last will be created.

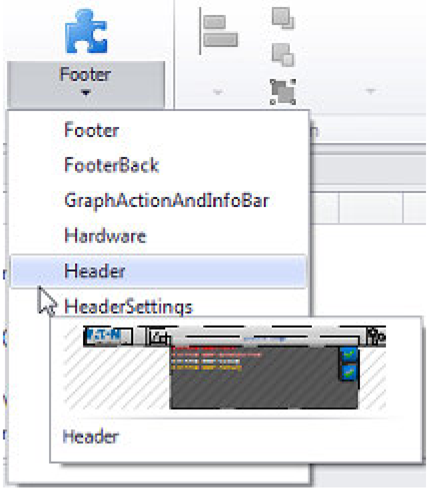
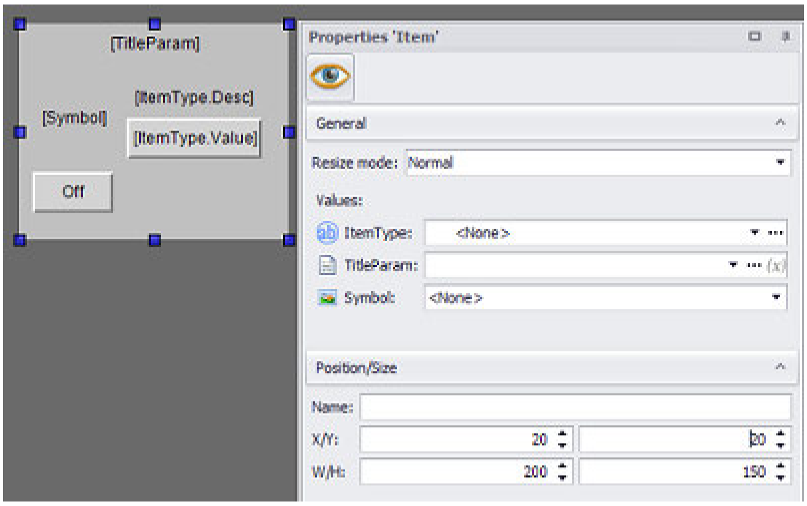


Figure 179: Example showing the drop-down menu in the Controls tab

Every single action related to user controls will be shown in Project Tree\Screens and in the ribbon bar for the Controls tab, on .
As soon as a new user control is added to the project tree, it will also be available in the drop-down menu (the behavior is the same for Delete and Rename).

If you have defined parameters for a user control, these parameters will be listed directly in the Properties pane so that you can assign them values directly.



Properties 'Item'

General

Resize mode: Normal

Values:

ItemType: <None>

TitleParam:

Symbol: <None>

Position/Size

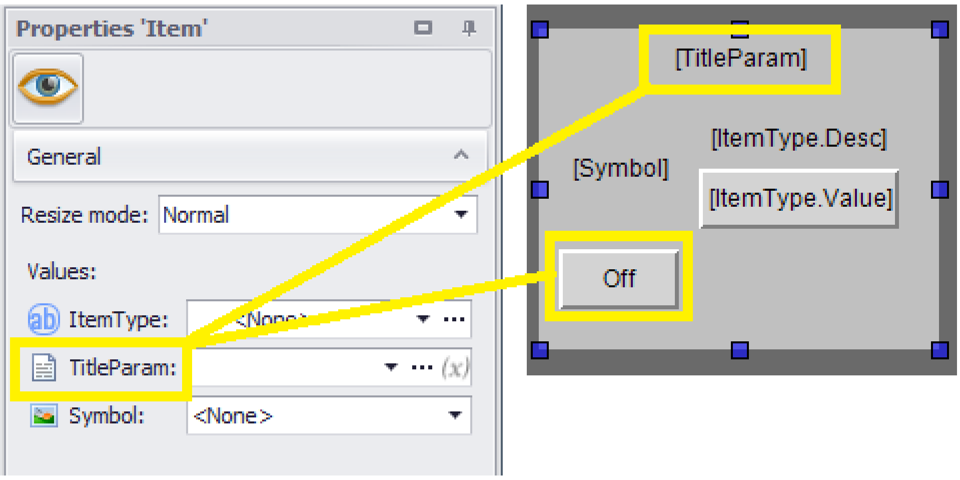
Name:

X/Y: 20 20

W/H: 200 150

Figure 180: Example showing a user control called <Item> that features parameters

Clicking on a parameter name in the Properties pane will highlight the references to that parameter in the selected user control.



Properties 'Item'

General

Resize mode: Normal

Values:

ItemType: <None>

TitleParam:

Symbol: <None>

Figure 181: Example showing a user control called <Item> that features parameters

For size changes, it is necessary to specify how the object should behave by configuring the corresponding settings for each object.

There are two ways to define the size change for an object:

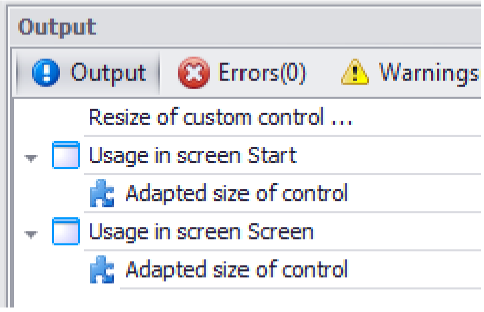
- Anchor points as a property for the object
- Resize mode as a property for the user control

Anchor points are set as a property in the Position/Size properties for an object; please refer to → section "Anchor points", page 244.

Meanwhile, resize mode can be configured as a property in the General properties for the user control on the screen.

This means that resize mode can be used to change the size of existing user controls without having to configure anchor points.

If the size of a user control itself changes in the project, all instances of that user control in the project will need to be adjusted. The user will be notified of this automatic size change to the instances of the user control in the Output pane.



Output

Output | Errors(0) | Warnings

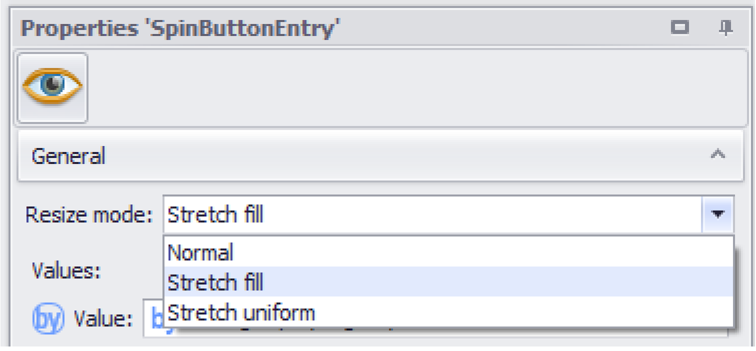
Resize of custom control ...
Usage in screen Start
Adapted size of control
Usage in screen Screen
Adapted size of control

Figure 182: Message indicating the size change in the Output pane

Resize mode

Only relevant to user controls.

The following options are available:



Properties 'SpinButtonEntry'

General

Resize mode: Stretch fill

- Normal
- Stretch fill
- Stretch uniform

Values:

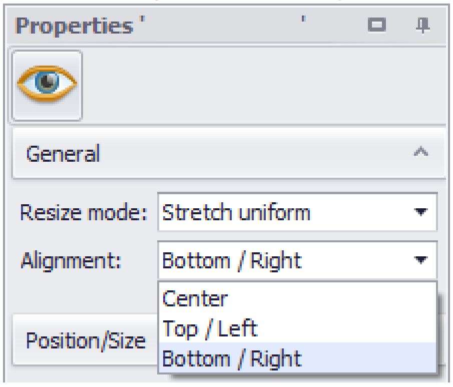
Value:

Figure 183: Selecting a resize mode

Regular — Use this mode if you have configured the anchor points and you want these anchor points to be applied when the size changes.

Fill — Resizes all the objects in the user control in such a way that the user control will be completely filled out.
Use this mode if you want to resize the user control space completely.

Uniform — Resizes all the objects in the user control in such a way that the sizes of the objects relative to each other will remain the same. Use this option if it is important for the contents of the user control to keep the same proportion relative to each other. Then select the objects' alignment in order to define the direction from which the size adjustment will be made (centered; top/left; bottom/right).



Properties '

General

Resize mode: Stretch uniform

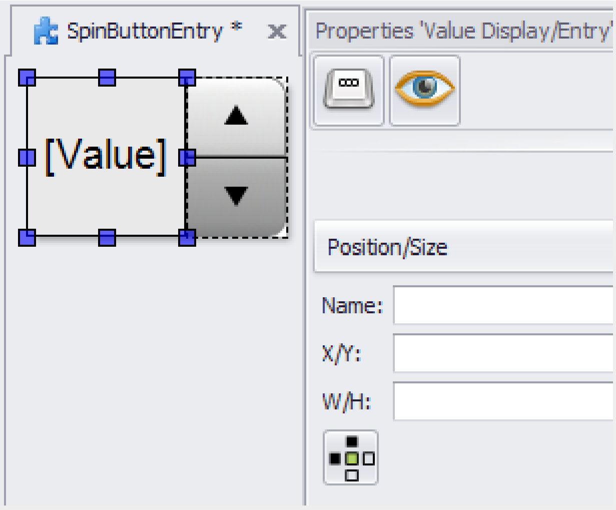
Alignment: Bottom / Right

- Center
- Top / Left
- Bottom / Right

Position/Size

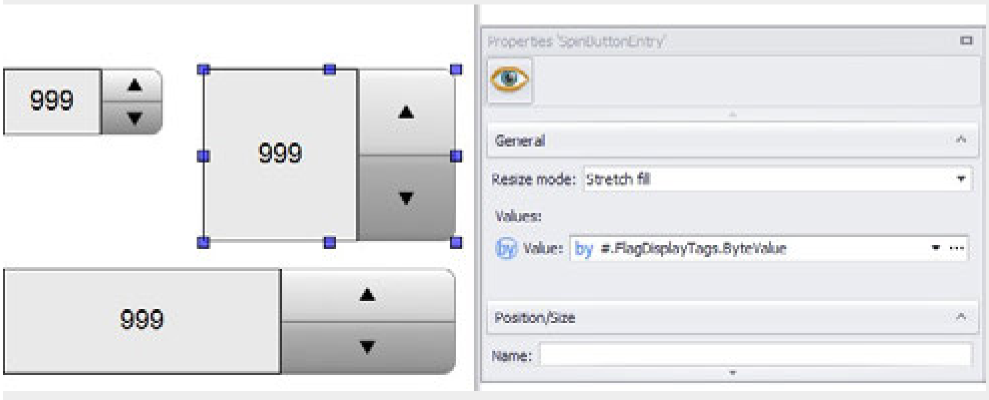
Resize mode example

Say that a user control called "SpinButtonEntry" has already been created in an older version of GALILEO.



The "left" and "top" anchors will be automatically enabled for all objects for imported user controls.

If you use this user control several times on a screen and always set the resize mode to "Stretch fill," the results will look as follows:



Resize mode: Stretch fill

Value: by #.FlagDisplayTags.ByteValue

Background color

You can change the color if you want. The following options are available for this purpose:

- Transparent
- Standard Colors
- More Colors...

The most recently used colors will be shown (user-specific).

For detailed information on the available colors, please refer to the following section: → section "Color choice - Object's Color Palette", page 234

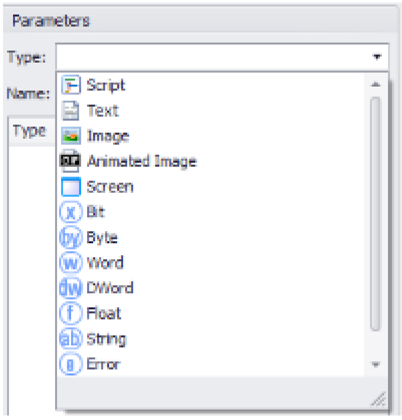
W/H

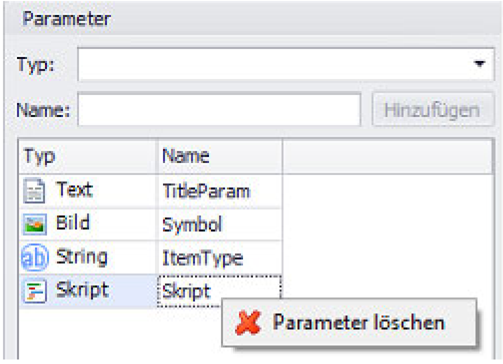
Screen width and height, in pixels.

Parameters

Input parameters consist of a data type and an identifying name.

The drop-down menu provides all available data types for selection. Once you have entered a name, you can click on the Add button to add the parameter to the table, after which it will be available for the user control.





Typ	Name
Text	TitleParam
Bild	Symbol
String	ItemType
Skript	Skript

Any parameters that are not needed anymore can be deleted from the table using the context menu.

For detailed information on how to use input parameters in user controls, please refer to → section "Using user controls to optimize work", page 279

2.2.3.1 Dragging and dropping user controls

You can select a user control and then add it as an element to any screen type in the project tree directly by dragging and dropping it. When doing so, please keep in mind the objects' functionality and the various limitations that apply to the various screen types in terms of the objects that can be used in them.

There are various possible options and behaviors, with the specifics depending on the target screen, the mouse button used (left/right), and the object affected by the cursor position when the object is dropped.

Dragging and dropping with the left mouse button

An object of type user control that represents the dragged user control screen will always be created. Within this context, it is irrelevant whether you drop the user control in an empty area of a screen or in an area where there is already an object.

Drag&Drop with the right mouse button

Shows the context menu with various options.

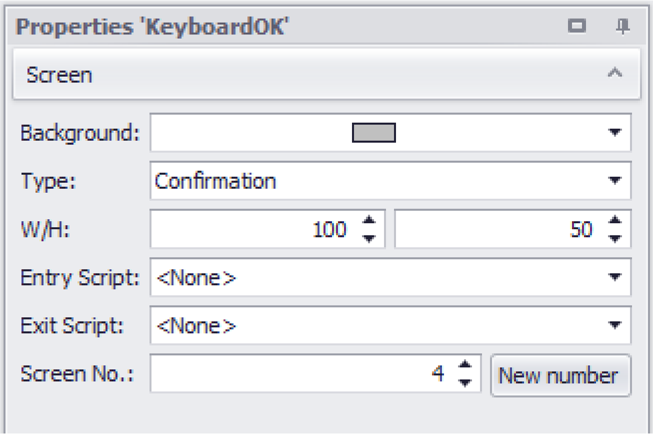
The following table shows the options that can be listed in the context menu, with the specific options depending on the target screen type and on whether the user control is placed on an object of type "container" or somewhere else.

Element from project tree Target screen	Used on container	Used elsewhere
Standard screen	Add to container UserControl Container Menu Button Dialog button	UserControl Container Menu Button Dialog button
User Controls	Not possible, since "container" controls are not allowed	User Control
Help Screens	Add to container UserControl Container	UserControl Container
Keyboards		
Printreport		
Background screen	Add to container UserControl Container Menu Button Dialog button	UserControl Container Menu Button Dialog button

2.2.4 Keyboard Properties

These screens can be used to create custom keyboards. For a description of the properties of keyboards, please refer to the → section "Keyboards", page 309 section. For a description of how to create a custom keyboard, please refer to the→ section "User Defined Keyboards", page 316 section.

Project Tree\Keyboard folder\"keyboardOK"



Properties 'KeyboardOK'

Screen

Background:

Type: Confirmation

W/H: 100 50

Entry Script: <None>

Exit Script: <None>

Screen No.: 4

Figure 184: Properties pane for "keyboardOK" keyboards

Background color

You can change the color if you want. The following options are available for this purpose:

- Transparent
- Standard Colors
- More Colors...

The most recently used colors will be shown (user-specific).

For detailed information on the available colors, please refer to the following section:
→ section "Color choice - Object's Color Palette", page 234

Transparent

The transparent background color is only available for user controls, help screens, and keyboards. The <transparent> background color is not available for standard screens.

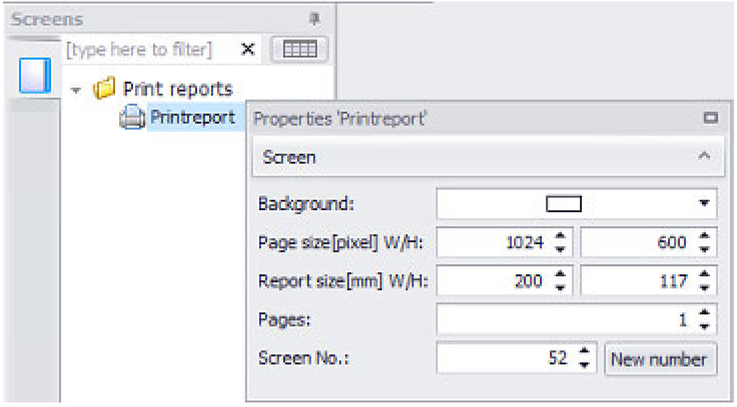
Activating this check box provides a screen with a transparent background so that a color setting cannot be made for it. On the device a screen change to a screen with a transparent background gives the appearance of overlaying the new screen on the previous one.

→ The zones in the previous screen will remain visible, but will no longer be functional.
Only the objects in the current screen will be usable and updated.

2.2.5 Print report properties

Print reports are screens that are used for printouts only.

Project Tree\Print reports folder\"Printreport0"



Background:	▭	
Page size[pixel] W/H:	1024	600
Report size[mm] W/H:	200	117
Pages:	1	
Screen No.:	52	New number

Figure 185: Properties pane for the "Printreport0" print report

For a description of the properties of print reports, please refer to the→ section "Print report", page 1029 section

2.3 Working with the screen table

To open the screen table, click on the ▦ button in the Screens project tree tab.

The table overview will show all screens and their settings in the project, as well as statistics concerning their use.

Project Tree\Screens

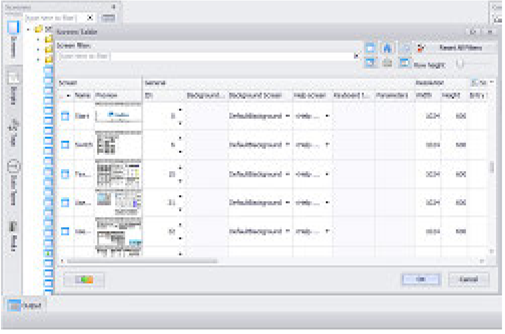


Figure 186: Screen table open

You can use the screen table to see the properties of all screens regardless of screen type. Properties that can be changed will be highlighted in white. Meanwhile, properties with a gray background cannot be edited in the screen table.

2.3.1 Screen filters in the screen table

There are various filters available at the top of the table dialog box. These filters can be combined.

Text

If you need to, you can simply filter screens by name.
All you need to to do is start typing in the Screen filter: field and the screens will be filtered as you type.

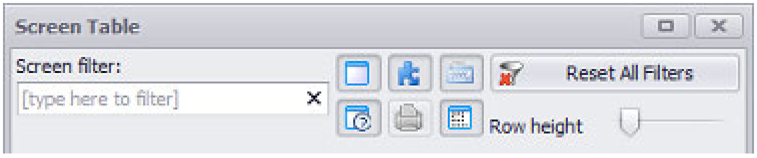


Figure 187: Screen table

Buttons for filtering by screen type

- Standard Masks
- UserControls
  - Dialog screens
- Keyboards
- Help Screens

- Printreport
- Background Screens

If a certain screen type is not yet in use in the project tree, the corresponding button will be grayed out.

Reset All Filters Cancels all applied filters and filter combinations in the screen table.

In order to find specific elements in the screen tables, use the column filter. For a description of this filter, please refer to → page 125.

Row height

The Row height slider can be used to adjust the height of the rows in the screen table between 50 px and 200 px.
You can use this setting to optimize the table view, adjusting it higher to make it easy to recognize individual screens or adjusting it lower if there is a large number of data sets in order to keep things manageable.

Customizing the screen table layout

You can change the order of columns and column blocks as necessary for your needs.
When you save the project, these changes will be saved as well.

Sorting

Most columns can be sorted by value.

▶ To sort by value, simply click on the column header

Please note that the Preview column cannot be sorted this way.

Copy table to clipboard

Clicking on the button will copy the screen table to the clipboard, after which it can be pasted into Microsoft Excel® or Microsoft Word® and worked on there.

- Excel - unformatted copy of cell contents
- Word - tab-separated cell contents

Column block Statistics

Controls, Tags, Scripts and Images will have corresponding statistics shown in the table.

Each of the statistics columns will show the number of objects that the corresponding screen uses. This number will include duplicates in nested groups and background screens, meaning that the number indicates the frequency of use.

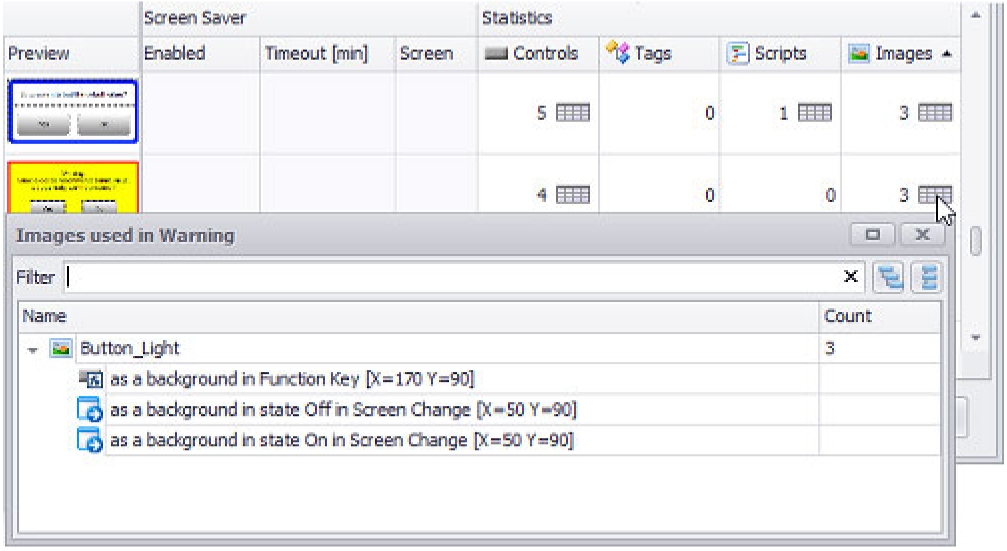
As soon as the number is not equal to zero, additional detailed information regarding the screen will become available.

▶ To access this information, click on the ▦ icon in the statistics column.

A dialog box with the following information will appear:

- Exactly which object (or tag, script, or image) is being used
- How often the object (or tag, script, or image) is being used in the selected screen
- The user control inside the screen in which the object (or tag, script, or image) is being used
- What the object (or tag, script, or image) is being used as

Objects used in <screen name>



	Screen Saver			Statistics			
Preview	Enabled	Timeout [min]	Screen	Controls	Tags	Scripts	Images
				5 ▦	0	1 ▦	3 ▦
				4 ▦	0	0	3 ▦

Images used in Warning

Filter

Name	Count
Button_Light	3
as a background in Function Key [X=170 Y=90]	
as a background in state Off in Screen Change [X=50 Y=90]	
as a background in state On in Screen Change [X=50 Y=90]	

Figure 188: Example showing the images used in the <Warning> screen

See also

→ section "Screens and User Controls", page 260

→ "Screen types", page 263

→ section " Screen change properties", page 449

→ section "Using user controls to optimize work", page 279

2.4 User controls

It is possible for the exact same arrangement of multiple visualization objects to be needed multiple times in a project. This specific combination of these individual objects can be thought of as a new project-specific "visualization object." In Galileo, this is called a user control.

User controls only need to be defined once in a project and can then be used multiple times in that same project.
Making a change to the original user control (e.g., changing the background color, adding another object) will apply the change to all other instances of the user control.

For a description of the properties of user controls and how to configure them, please refer to the → section "UserControls", page 267 section

Difference from "grouping objects"

You can also group multiple visualization objects into a single object by "grouping objects." This can be particularly useful when you do not want the specific object arrangement on a screen to be modifiable, i.e., when you want all objects to be moved proportionately when one of the objects is moved.

However, "object groups" are always independent entities. If you copy & paste an object group, the two groups will remain independent from each other, and making a change to the first group of objects will not affect the copy in any way.

See also

→ "Grouping objects", page 113

2.4.1 Using user controls to optimize work

Reusing existing content

2.4.1.1 Using input parameters

Certain applications use a large number of visualization objects of the same type to display values for a large number of system components of the same type that all have the same parameter set. For example, an application could use visualization objects to display three pieces of data for 10 different motors: the motor name, its horsepower, and its current speed.

In order to handle this sample application, you could create a gauge and assign the speed for Motor1 to it. You could then copy the gauge and assign the speed for Motor2 to it, etc. In order to avoid having to assign a large number of parameters in cases such as these, you can work with input parameters, which should be viewed as a sort of metatag. This type of approach can be used to assign tags to an entire parameter set with a single assignment definition. Another advantage is that the moment a change is made to on element, it will be propagated across all other elements.

Input parameters consist of a type and an identifying name. The use of an input parameter would make it possible, in our sample application, for all visualization objects to use the same structure element of type "MOTOR." The input parameter would not be replaced with actual data until the corresponding UserControl is used on another screen. The following parameter types will be available for selection:

- Tag data types
  (bit, byte, etc., as well as the element of a custom data type)
- Image
- Animated Image
- Screen
- Text
- Script

The Type name is used for later identification if the UserControl is used and thus if data is assigned.

Prerequisites

Visualization objects that use input parameters must be placed in UserControl.

Using the exact same name ("Main drive") will make both objects use the same instance of type "MOTOR" later on.
The "MOTOR" data type mentioned here could look the following way:
When user controls are used on a different screen, the configured input parameters are requested:
In this example, you would need to select a tag of type "Word" for "Ref Value" and an instance of type "MOTOR" for "Main drive."
For the specific examples mentioned, the "PLC_PRG.wValue1", "PLC_PRG.motor_x_1.currentRPM", and "PLC_PRG.motor_x_1.ID" project tags would be assigned to the objects in the user control.

Example showing how to use input parameters

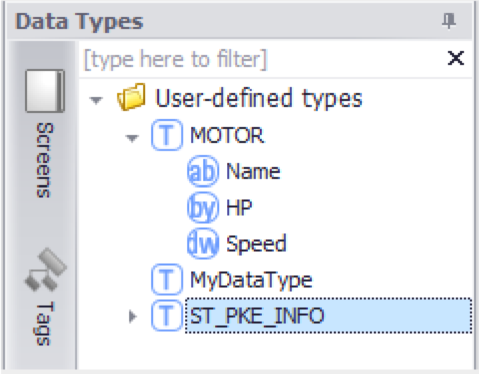
Say you want to show 4 motors with the corresponding motor name, horsepower, and current speed on a screen.

Define a Data type

Create a user-defined data type called MOTOR. MOTOR should have three structure elements

- <Name>, data type STRING; displayed with a PLC Text visualization object.
- <HP>, data type BYTE; displayed with a value display/entry visualization object.
- <Speed>, data type DWORD; displayed with a gauge visualization object.

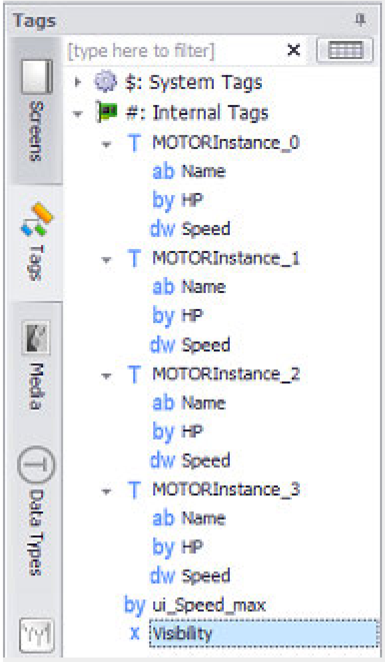
Project Organization\Folder Data Types



Instantiating tags

- ▶ Create four tags of user-defined data type MOTOR: MotorInstance_ 0,MotorInstance_1, MotorInstance_2,MotorInstance_3
- ▶ Create a tag of data type BIT called <Visibility>, for example.

Project Tree\Tags tab

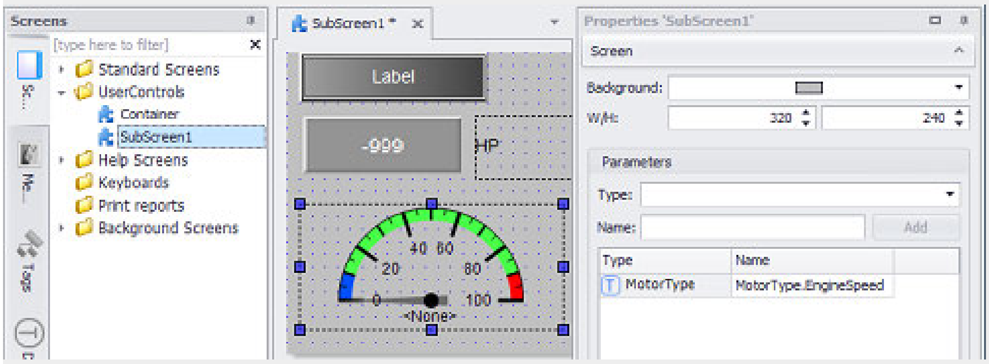


Creating a user control

To create a new UserControl **<Subscreen1>** with the aforementioned visualization objects, follow the steps below:

- ▶ Go to *Home\Project Tree\Screens\UserControls folder*.
- ▶ Create a new user control
  by right-clicking and selecting the *Add new UserControl* option from the context menu. Enter**<Subscreen1>** as the name.

- Add PLC text, value display/entry, text and gauge visualization object, one after the other, to UserControl **<Subscreen1>**.



Background:
W/H: 320 240
Type:
Name:

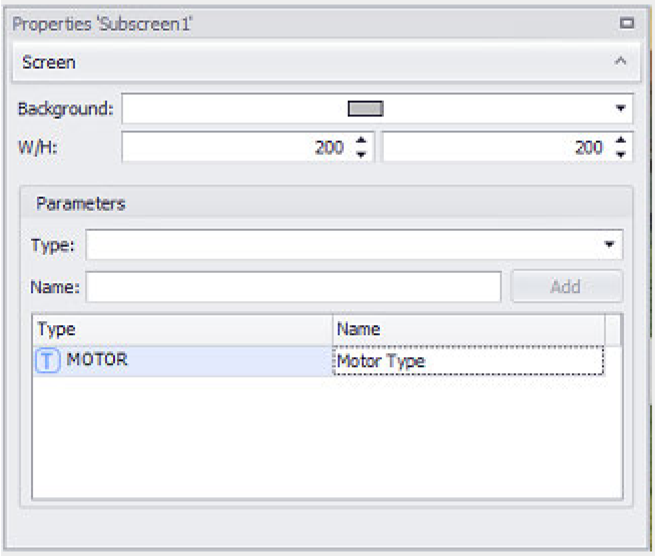
Type	Name
MotorType	MotorType.EngineSpeed

Using input parameters in a user control

Use Parameters to create an input parameter called Motor Type with the MOTOR user-defined data type for the **<Subscreen1>** UserControl.

In addition, you will be defining a second input parameter called Visibility. This input parameter will be of data type bit and will be used to make it possible to hide and show the gauges on all UserControls at runtime.

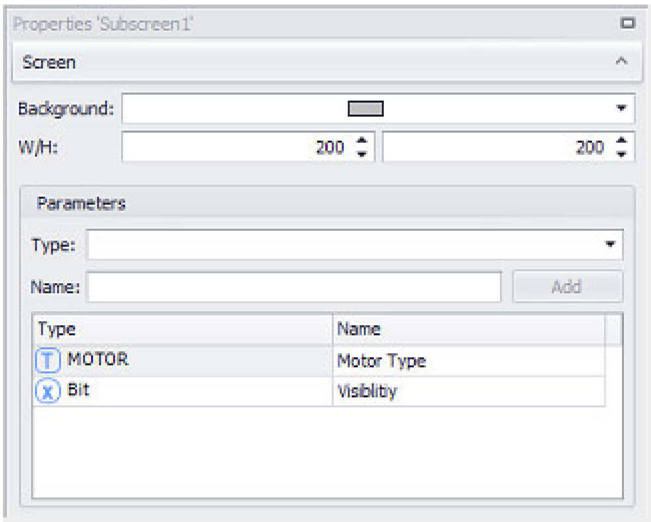
- Click on the drop-down menu arrow for the Type field and select the MOTOR user-defined data type. Then enter Motor Type as the <Name> for the input parameter.
- Click on the drop-down menu arrow for the Type field and select the bit data type. Then enter Visibility as the <Name> for the input parameter.



Background:
W/H: 200 200
Type:
Name:

Type	Name
MOTOR	Motor Type

Figure 189: UserControl properties **<Subscreen1>**


Properties 'Subscreen1'

Screen

Background:

W/H: 200 200

Parameters

Type:

Name:

Type	Name
MOTOR	Motor Type
Bit	Visiblity

Figure 190: Properties of **<Subscreen1>** user control after the two assignment actions

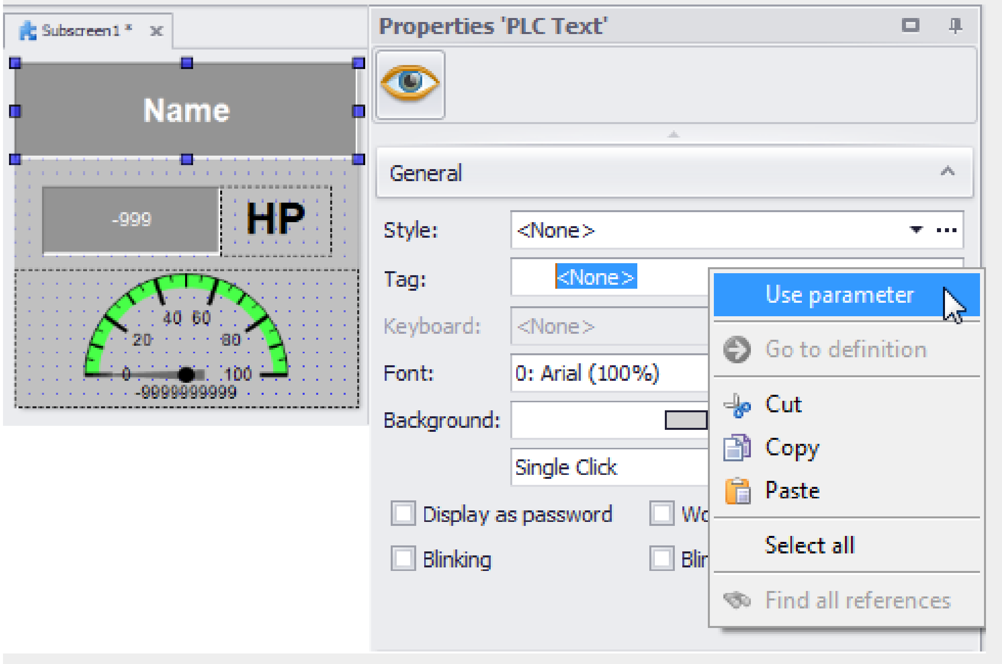
Assigning input parameters to the visualization objects in the UserControl

You will be assigning input parameters, instead of tags, to the individual visualization objects such as the PLC texts.

To assign the "Name" element from the user-defined "MOTOR" data type to the PLC text visualization object, follow the steps below:

→ Only structure elements and tags of data types that can be used with the corresponding visualization object will be available for selection.

- Place the cursor inside the Tag field.
- Right-click to open the context menu and select the Use parameter option


Properties 'PLC Text'

General

Style: <None>

Tag: <None>

Keyboard: <None>

Font: 0: Arial (100%)

Background:

Single Click

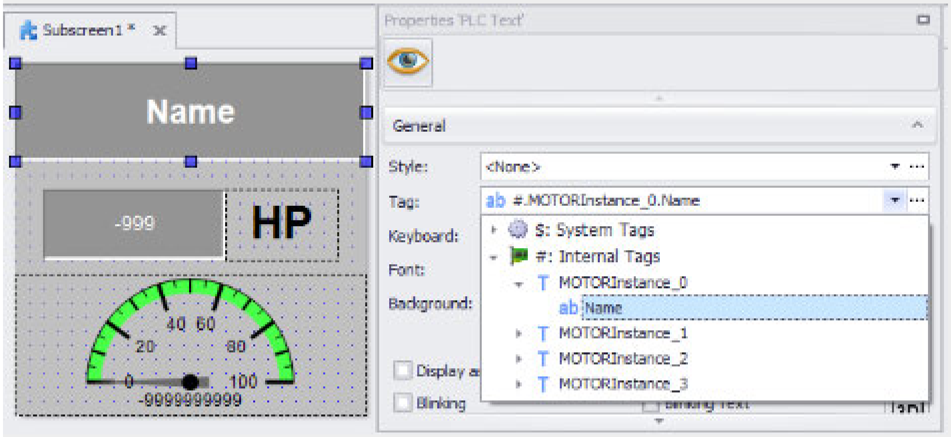
□ Display as password □ Wo

□ Blinking □ Blin

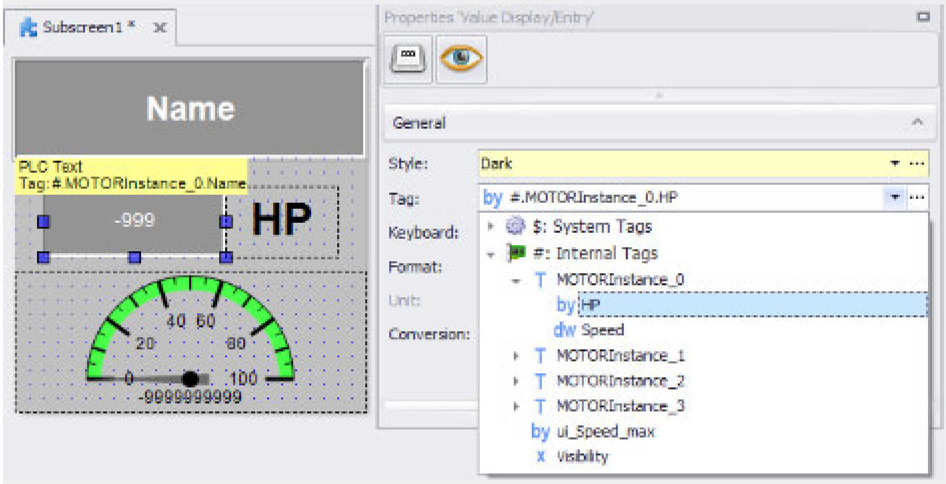
The field's name will change from Tag to Parameter.

- Click on the drop-down menu arrow.

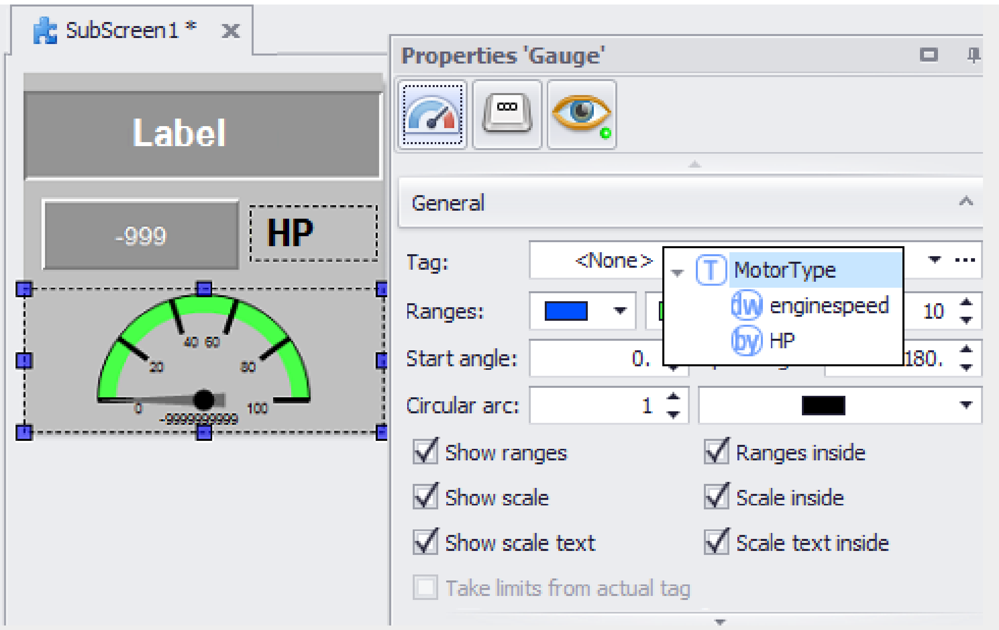
All the input parameters defined for this UserControl in the Extended properties dialog box will be shown.



- Select the structure element that should be displayed with the visualization object, e.g., Motor Type.Name.
- Use the same procedure to assign the Motor Type.HP input parameter to the value display/entry object.



- Use the same procedure to assign the Motor Type.Speed input parameter to the gauge object.



Properties 'Gauge'

General

Tag: <None>
- MotorType
- enginespeed
- HP

Ranges: 10

Start angle: 0. 180.

Circular arc: 1

☑ Show ranges ☑ Ranges inside

☑ Show scale ☑ Scale inside

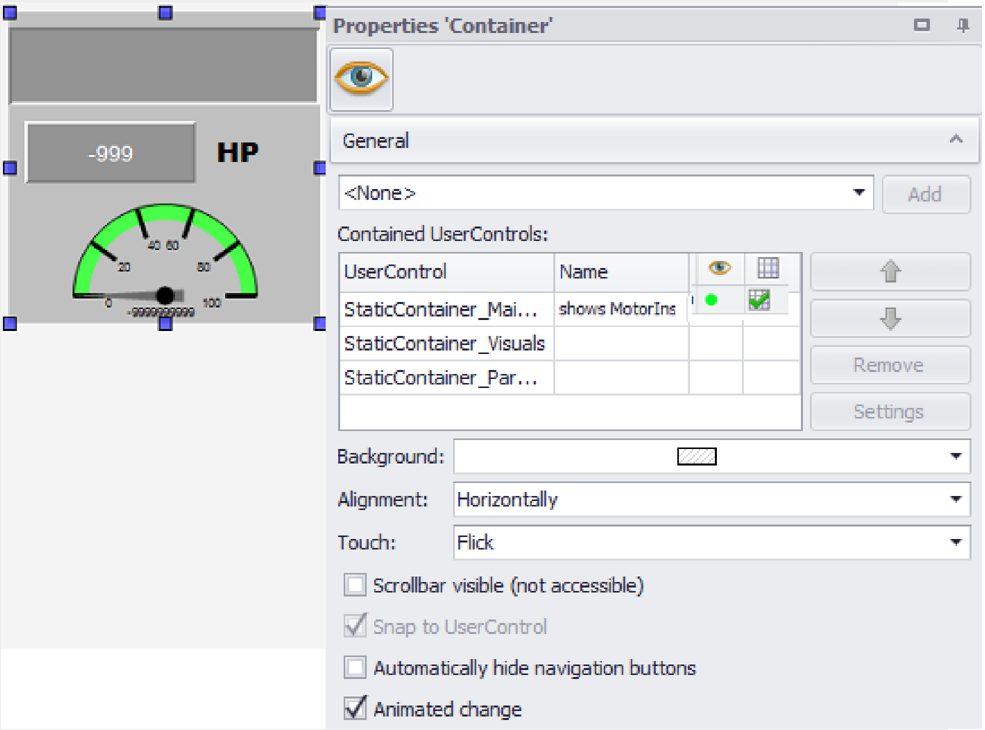
☑ Show scale text ☑ Scale text inside

□ Take limits from actual tag

Creating a screen and filling out input parameters with tags

Create a screen called MotorDisplay.

- Go to *Project Tree\Screens\UserControls folder* and, for each motor, drag and drop a **<Subscreen1>** object onto the MotorDisplay screen.
- Select the first UserControl on the top left.



Properties 'Container'

General

<None> Add

Contained UserControls:

UserControl	Name		
StaticContainer_Mai…	shows MotorIns	●	☑
StaticContainer_Visuals			
StaticContainer_Par…			

Remove
Settings

Background: ▨

Alignment: Horizontally

Touch: Flick

□ Scrollbar visible (not accessible)

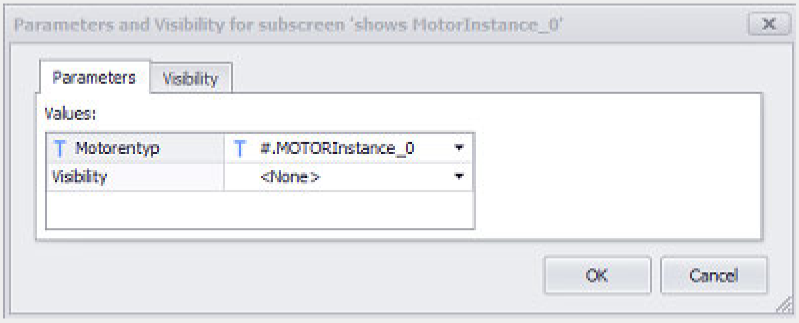
☑ Snap to UserControl

□ Automatically hide navigation buttons

☑ Animated change

Figure 191: "MotorDisplay" screen with the top left **<Subscreen1>** UserControl selected. On the right: The Properties 'Container' pane

- In the Properties 'Container' pane, double-click on ▦ to open the Properties 'User Control' pane and click on the Settings button to open the Parameters and Visibility for UserControls 'User Control' dialog box.



Parameters and Visibility for subscreen 'shows MotorInstance_0'

Parameters | Visibility

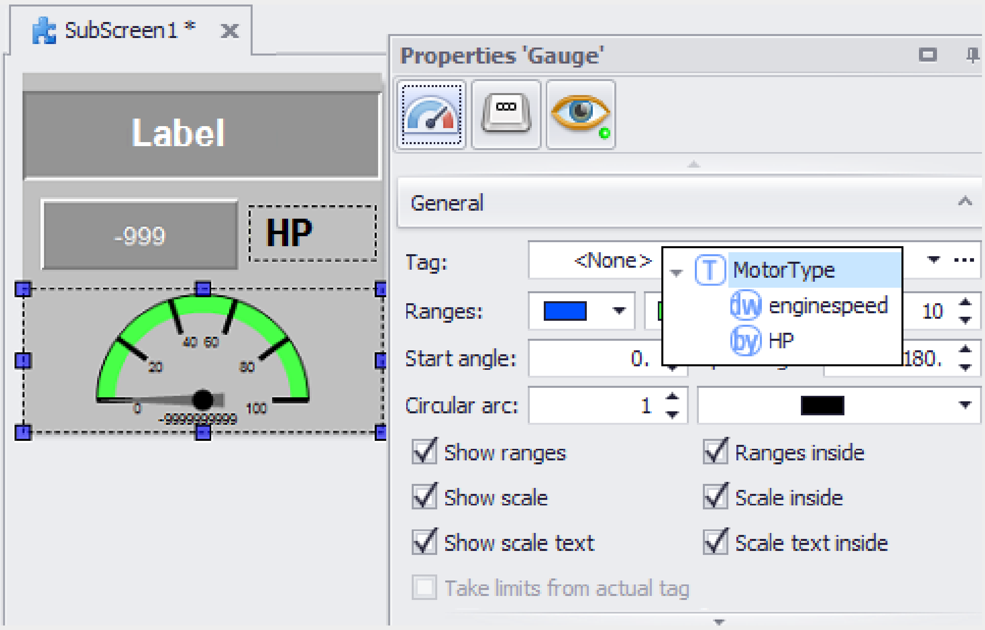
Values:

T Motorentyp	T #.MOTORInstance_0
Visibility	<None>

- Click on the drop-down menu arrow to assign the MotorInstance_0 tag to the Motor Type input parameter in this instance of the **<Subscreen1>** user control.
- Enter a comment into the Name field, e.g., <Shows MotorInstance_0>
- You can also assign the Show tag to the Visibility input parameter by clicking on the drop-down menu arrow, provided you have already created the Visibility input parameter.
- Click on OK.

Using input parameters instead of tags for accessibility/visibility in the UserControl

Instead of using tags for the accessibility or visibility of a visualization object, you can use input parameters. The corresponding visualization object must be in the UserControl.



Properties 'Gauge'

General

Tag: <None>

T MotorType
dw enginespeed
by HP

Ranges: 10

Start angle: 0. 180.

Circular arc: 1

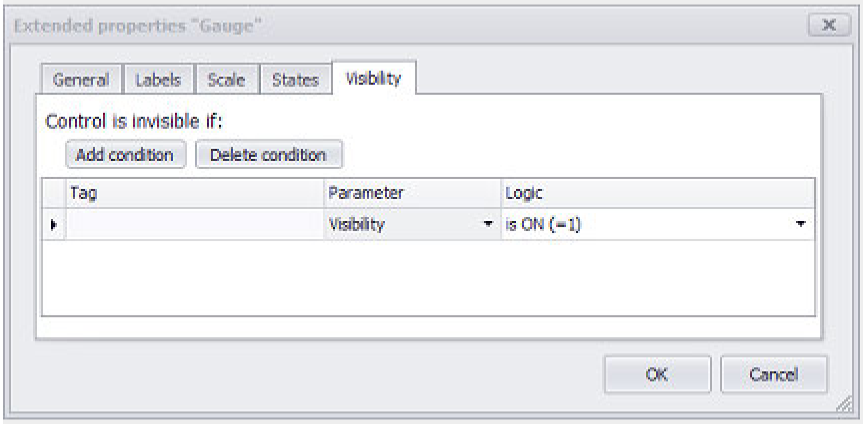
☑ Show ranges ☑ Ranges inside
☑ Show scale ☑ Scale inside
☑ Show scale text ☑ Scale text inside
☐ Take limits from actual tag

- Click on the visualization object on the UserControl, e.g., gauge.
- Click on the 👁 button to access the Visibility property.

- ▶ Place the cursor inside the Tag field.
- ▶ Right-click to open the context menu and select the Use parameter option.
- ▶ Place the cursor inside the Tag field.
- ▶ Right-click to open the context menu and select the Use parameter option.

The focus will jump to the Parameter column.

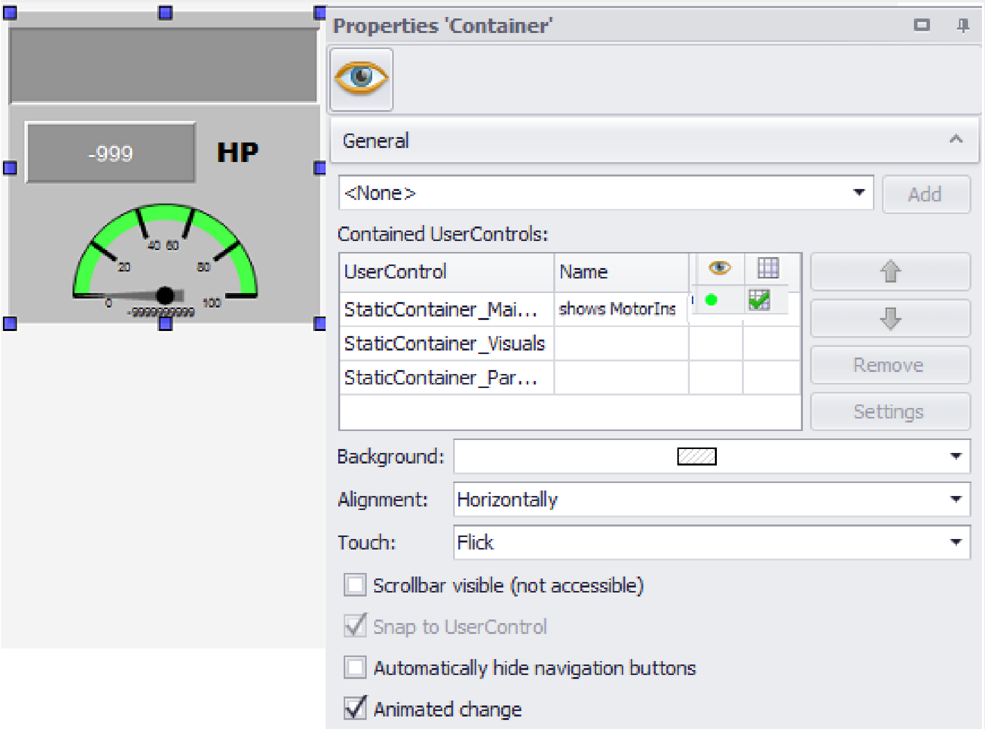
- ▶ Use the drop-down menu to select the Visibility input parameter you created previously.
- ▶ Click on OK.



Extended properties "Gauge"

Control is invisible if:

Tag	Parameter	Logic
	Visibility	is ON (=1)



Properties 'Container'

<None>

Contained UserControls:

UserControl	Name	👁	▦
StaticContainer_Mai…	shows MotorIns	●	☑
StaticContainer_Visuals			
StaticContainer_Par…			

Background: ▨

Alignment: Horizontally

Touch: Flick

☐ Scrollbar visible (not accessible)

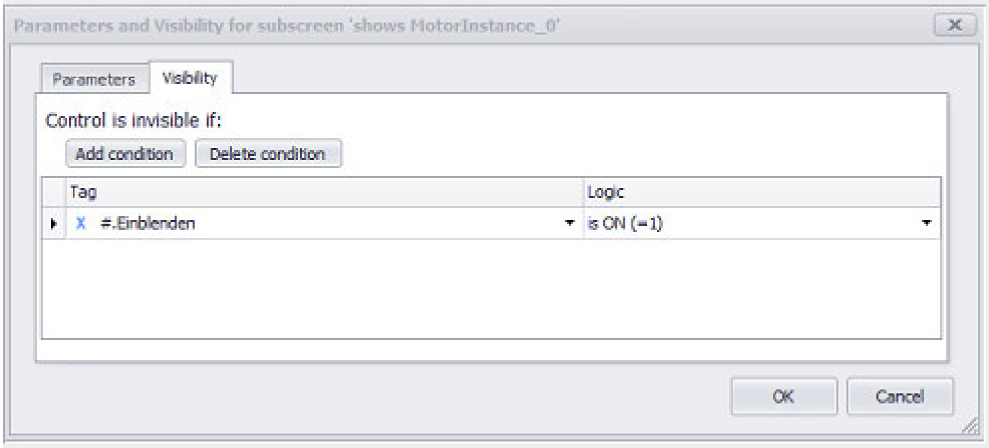
☑ Snap to UserControl

☐ Automatically hide navigation buttons

☑ Animated change

- ▶ Click on the Visibility button to open the visibility settings for the **<Subscreen1>** UserControl.

- ▶ Use the drop-down menu in the Tag column to select the Show tag
- ▶ Click on OK.



Parameters and Visibility for subscreen 'shows MotorInstance_0'

Control is invisible if:

Tag	Logic
#.Einblenden	is ON (=1)

The runtime result on the screen will look as shown below:

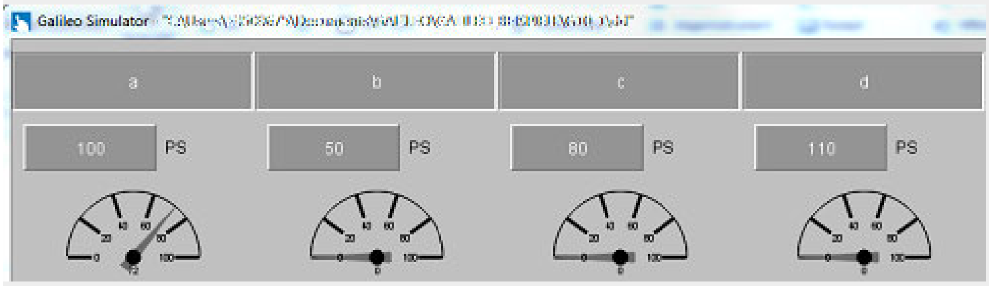


Figure 192: MotorDisplay screen on the device screen at runtime

2.4.2 Nesting user controls

If you have created a user control, you can reuse it and insert it inside other user controls.

This means that you can use user controls as many times as you want and nest them if user controls contain other user controls.

Example showing an object group

Say you want to create a group of objects so that you can manage it centrally.

To do this, create a user control with these contents as a source.

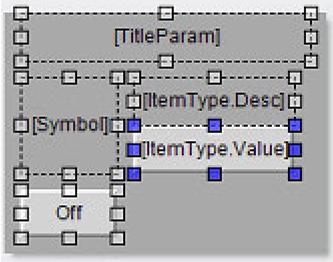


Figure 193: Example: A group of individual objects

You will only need to configure the properties for the user control - Screen (background, width/height); Parameters - once. These defined properties will then be automatically applied to other instances of the user control.

This means that the user control can

- Be used as many times as needed
- Be combined with other user controls into a new, additional user control that can be saved
- Be added to containers

anywhere where you need it.

The user controls you create will be directly available through a drop-down menu in the Controls tab.

This dramatically reduces the amount of time and effort required to make changes to screen contents, since making changes to the individual user control will automatically apply these changes to all of its instances.

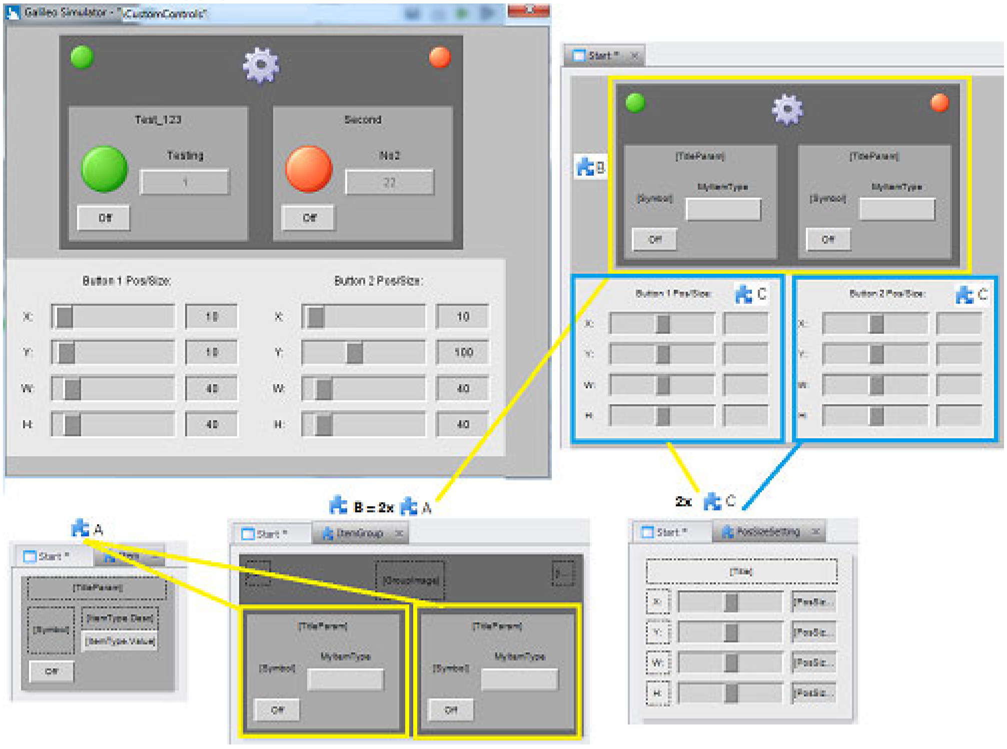
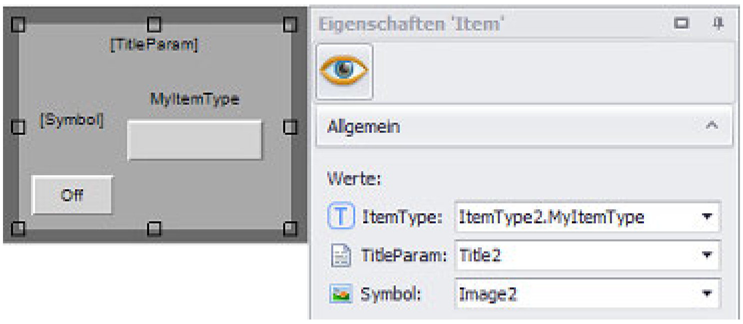


Figure 194: User control nesting

Assigning parameters

If you use the user control, you can assign values to the corresponding parameters.

This can be done either directly in the object's properties or, if the user control is being used in a container, in the "Parameters and Visibility" dialog box.



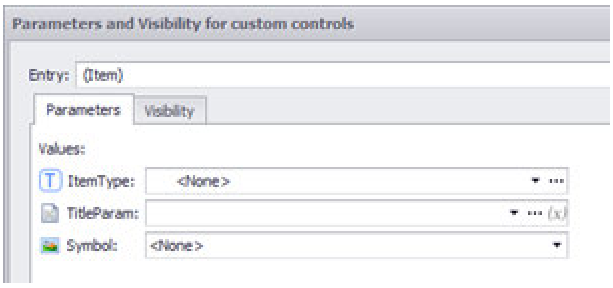
Werte:

ItemType: ItemType2.MyItemType

TitleParam: Title2

Symbol: Image2

Figure 195: Properties pane for a user control called **<Item>** that features parameters



Entry: (Item)

Values:

ItemType: <None>

TitleParam:

Symbol: <None>

Figure 196: Dialog box for a user control called **<Item>** that features parameters

When nesting user controls, there is another way to use available parameters.

Application example

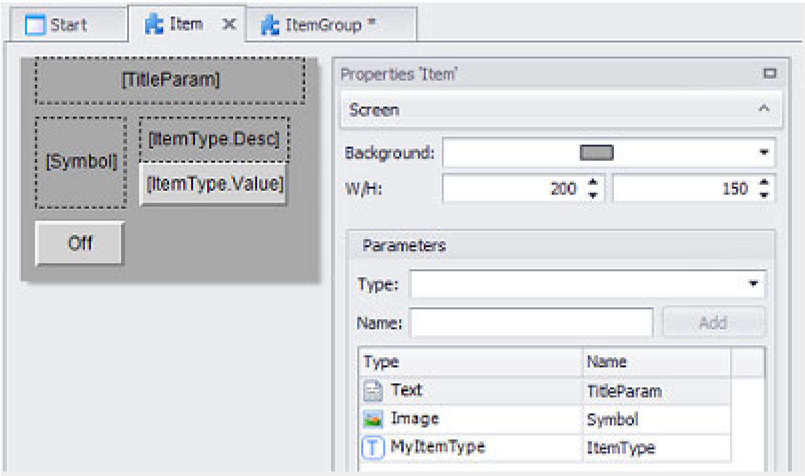
Say a user control named **<Item>** requires certain parameters

Moreover, say you want another user control named **<ItemGroup>** to contain multiple user controls named **<Item>**.

Finally, say you want the parameters defined for the **<ItemGroup>** user control to also apply to the **<Item>** user controls it contains.

To do this, all you would need to do would be to assign the **<ItemGroup>** user control the values for the parameters.
These parameter values will then be automatically inherited by the **<Item>** user controls inside.

To illustrate:

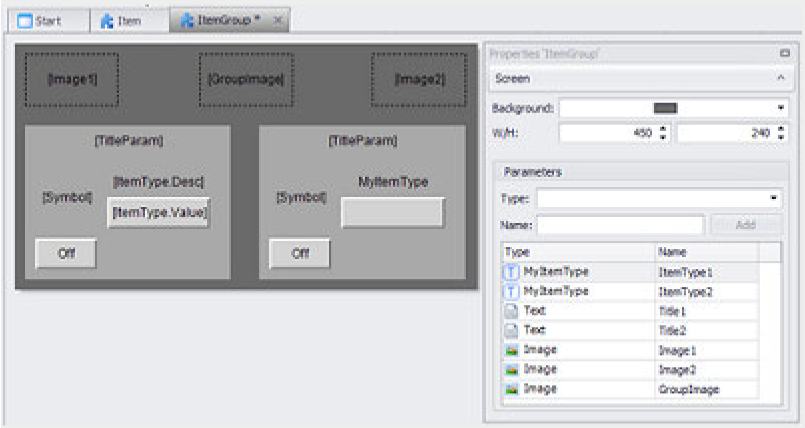


Background:
W/H: 200 150

Type:
Name:

Type	Name
Text	TitleParam
Image	Symbol
MyItemType	ItemType

Figure 197: User control named **<Item>** and featuring the following parameters: <TitleParam> Text, <Symbol> Image, and a parameter named <ItemType> consisting of a UDT instance of type "MyItemType"



Background:
W/H: 450 240

Type:
Name:

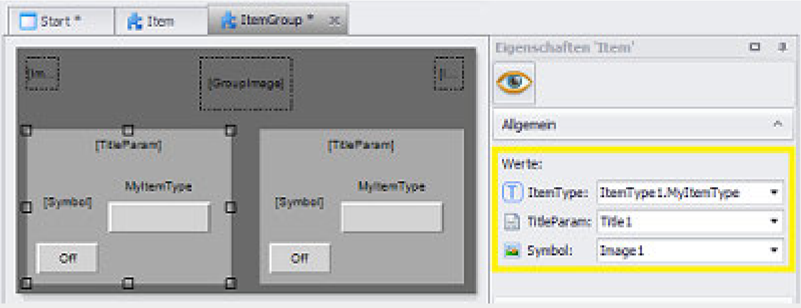
Type	Name
MyItemType	ItemType1
MyItemType	ItemType2
Text	Title1
Text	Title2
Image	Image1
Image	Image2
Image	GroupImage

Figure 198: A different user control named **<ItemGroup>** that contains two **<Item>** user controls

The **<ItemGroup>** user control expects seven parameters, as the table in the Properties pane shows.
Say you want to use the **<Item>** user control on the left to display the <ItemType1>, <Title1>, and <Image1> parameters and the one on the right to display the <ItemType2>, <Title2>, and <Image2> parameters.

In addition, you will also be using the <Image1> and <Image2> parameters directly in the **<ItemGroup>** user control.

When one of the user controls is selected, the Properties pane will first show the following:



ItemType: ItemType1.MyItemType

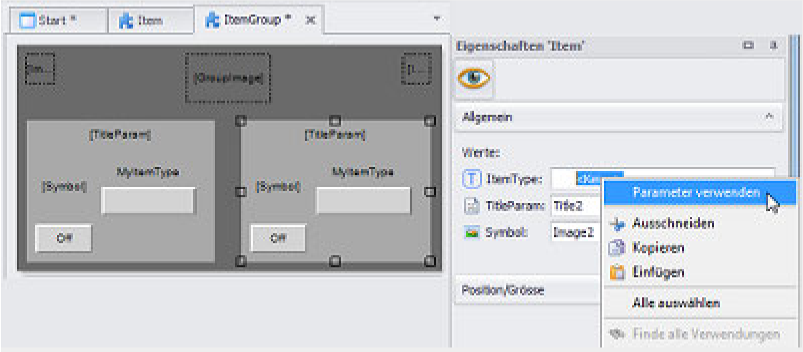
TitleParam: Title1

Symbol: Image1

All the parameters defined for the **<Item>** user control are listed.

By default, the input fields for assigning values will be in the Values mode, in which directly available project data can be assigned (e.g., tags you have created or images from the project tree).

This mode can be switched to Use parameters by using the context menu accessed by right-clicking.

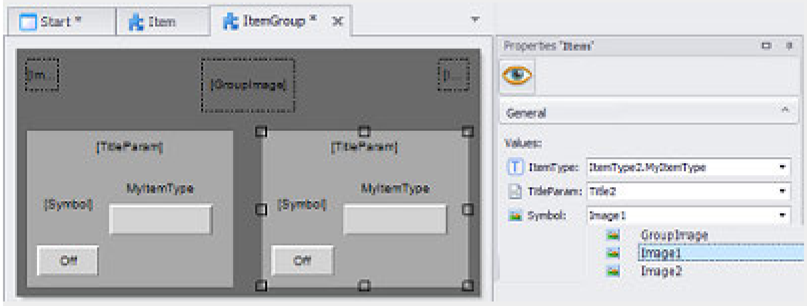


ItemType: <Kein...

TitleParam: Title2

Symbol: Image2

When you select Use parameters, the input field will offer the compatible parameters from the **<ItemGroup>** user control.



ItemType: ItemType2.MyItemType

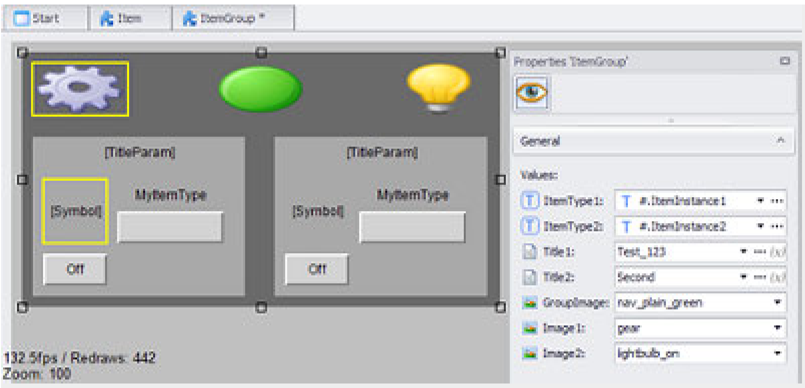
TitleParam: Title2

Symbol: Image1

If <Image1> is selected now, this means that:

As soon as the <ItemGroup> user control is used and an image is assigned to the <Image1> parameter, the image will be used here as well.

A gear image is stored as <Image1> in this example.



Properties 'ItemGroup'

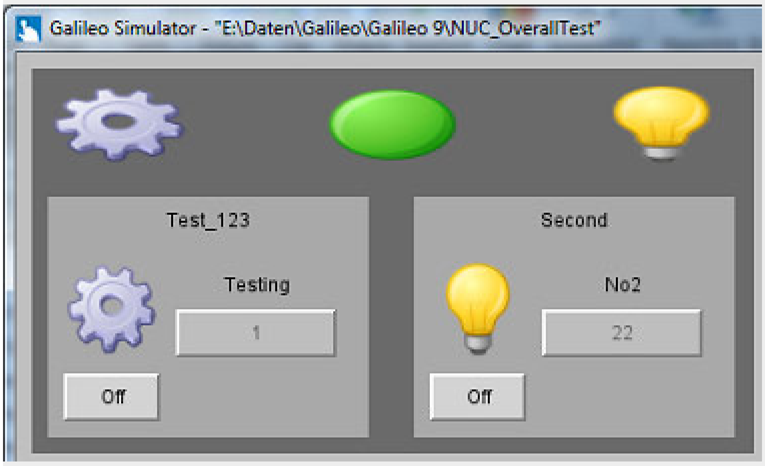
General

Values:

ItemType1:	#.ItemInstance1
ItemType2:	#.ItemInstance2
Title1:	Test_123
Title2:	Second
GroupImage:	nav_plain_green
Image1:	gear
Image2:	lightbulb_on

If you click on the name of <Image1>, you will be able to see where the <Image1> parameter is used in the **<ItemGroup>** user controls.

When executed in the simulator, the following will appear at runtime:


Galileo Simulator - "E:\Daten\Galileo\Galileo 9\NUC_OverallTest"

→ section "UserControls", page 267
→ section "Using containers", page 294
→ section " Containers", page 298

2.5 Using containers

One of the special characteristics of containers consists of how they are displayed. A container has a set display size that can be defined, and the user controls in it can be scrolled through vertically or horizontally at runtime. The user controls in the container should all have the same size.

A container can be reused for as many screens as necessary; in addition, containers can be used on any screen, with the exception of UserControls.

2.5.1 Example with a container containing three images

Say you want a container to display three images that can be vertically scrolled through. This example will use the images from DemoApplication2.prj, but the example itself is not actually found in DemoApplication2.prj.

To implement a container on the start screen, follow the steps below:

Copying the images to Project images

- ▶ Open your directory structure and look for the following images in the installation path
  for the DemoApplication2.prj\images examples in order to drag and drop them onto the Project images folder: <Eaton_DiscValve_Motor.png>, <Eaton_Vane_Motor.png>, and <Eaton_VIS_Motor.png>.

Described in section:

→ section "Substep 1: Adding an image (medium) to the project tree", page 174

Creating a user control

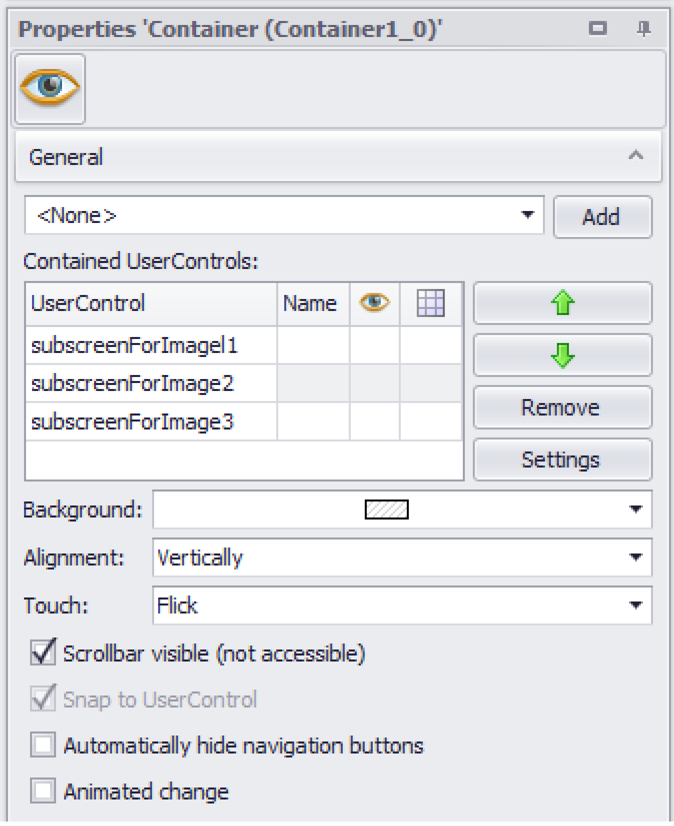
- ▶ Go to *Project tree\Screens\User Controls*.
- ▶ Create a new user control by clicking on the *Add new UserControl* option in the context menu, e.g., <subscreenForImage1>.
- ▶ Enter a width and height of 160 pixels for the subscreen.
- ▶ Select a blue tone as Background Color.
- ▶ Go to *Project Tree\Media\Project images* and drag & drop the first image onto the UserControl, e.g., <Eaton_DiscValve_Motor.png> onto <subscreenForImage1>.
- ▶ Repeat the steps and create two additional UserControls, e.g. <subscreenForImage2>, <subscreenForImage3>.

Create a container

- ▶ Switch to the Start screen.
- ▶ In the *Controls* tab, click on Container.
- ▶ Position the object
  Move the cursor to the screen and draw the area for the visualization object.

- In the 'Container' properties pane, use the drop-down menu to select the first UserControl, e.g., <submaskForImage1>, then click on the Add button.
- Repeat the step above to add the other two User Controls or
- dragging and dropping them from *Project Tree\Screens\User Controls* into the container on the screen.
- Select the <Vertically> option in the Alignment field.
- Enable the Scrollbar visible checkbox.
- Start the Galileo Simulator with function key F5 or by clicking on Home\ . Build.

You will be able to scroll through the three UserControls containing an image in either one of two ways on the device screen or on the PC: by moving the scrollbar up and down or by using a swipe gesture inside the container.



Properties 'Container (Container1_0)'

General

<None> ▾ Add

Contained UserControls:

UserControl	Name		
subscreenForImage1			
subscreenForImage2			
subscreenForImage3			

Background: ▨

Alignment: Vertically

Touch: Flick

☑ Scrollbar visible (not accessible)

☑ Snap to UserControl

☐ Automatically hide navigation buttons

☐ Animated change

Figure 199: Properties for container from example

Result on device screen
img src="../../../ResouresProjekt/imagesProjekt/demo_container.PNG" />
Figure 200: Device screen with container and scrollbar

2.5.2 Example with input parameters and a container containing three images

Where can input parameters come in handy? Using input parameters makes it possible to use a UserControl or a visualization object for a large number of parameter sets of the same type. This eliminates a lot of the work that goes into creating the same type of visualization object or UserControl many times.

Input parameters can be used for the following types of objects:

- Script
- Text
- Image
- Primitive Data Types
- User defined Data Types

In order to further illustrate this, we will be using the previous example, → "Example with a container containing three images", page 294, but with the use of input parameters. In order to be able to follow this example, you will have already needed to implement the previously described one.

You will be using input parameter <Image_No> to reduce the number of UserControls from three to one, i.e., <subscreenForImage1>.

To use input parameters, follow the steps below:

Define an input parameter for User Controls

- Open <subscreenForImage1> by double-clicking on *Project Tree\Screens\User Controls\<subscreenForImage1*.
- Delete the image from the previous example.
- In the properties for <subscreenForImage1>, click on the button.
- The Extended properties "subscreenForImage1" dialog box will appear.
- Use the Type drop-down menu to select the type of content that should be replaced with an input parameter. Select <Image>.
- Enter a name for the input parameter into the Name field, e.g., <Image_No>.
- Click on OK.

You have now created the Image_No input parameter.

Link the visualization object to the input parameter

- Click on *Controls*Image.
- Position the object
  Move the cursor to the screen and draw the area for the visualization object.
- Place the cursor inside the Image field in the Properties 'Image' pane.
- Right-click to open the context menu and select the Use parameter option.
- Select the <Image_No> parameter.

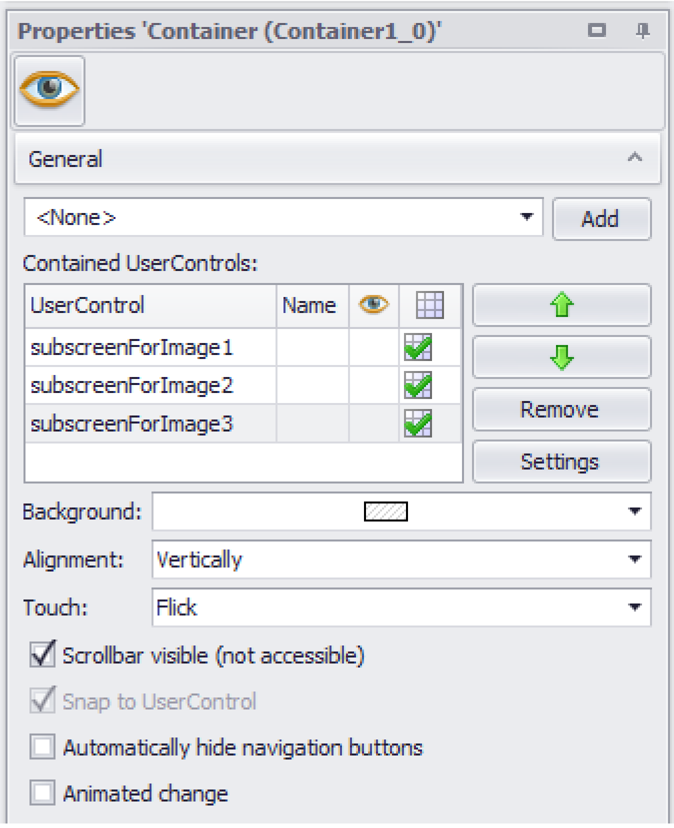
The image on <subscreenForImage1> will be linked to the input parameter.

Use the User Controls multiple times in a container and instantiate the input parameter

- Create a new standard screen, e.g., <ScreenWithContainer>.
- Consider creating a screen change that has this new standard screen as the target screen. Otherwise, you will not be able to switch to it at runtime.
- Click on *Controls*Container.
- Position the object
  Move the cursor to the screen and draw the area for the visualization object.
- In the 'Container'User Controls properties, select the <subscreenForImage1> from the drop-down menu.
- Click three times on the Button Add.
- In the User Controls column, click on <subscreenForImage1> and then on the Parameters button.
- The Parameters and Visibility for User Controls <subscreenForImage1> dialog box will open.
- In the drop-down menu for Image_No, select the first image, e.g., <Eaton_ DiscValve_Motor>.
- Click on OK.
- In the User Controls column, click on the second instance of <subscreenForImage1> and then click on the Parameters button.
- In the drop-down menu for Image_No, select the second image, e.g., <Eaton_ Vane_Motor>.
- Click on OK.
- In the User Controls column, click on the third instance of <subscreenForImage1> and then click on the Parameters button.
- In the drop-down menu for Image_No, select the third image, e.g., <Eaton_VIS_ Motor>.
- Click on OK.

You have now filled the container for the <ScreenWithContainer> screen with the same <subscreenForImage1> UserControl three times. In addition, you have instantiated the input parameter for the <subscreenForImage1> User Controls with three images.

The result on the device screen will be exactly the same as the result for the previous example → "Result on device screen", page 295



Properties 'Container (Container1_0)'

General

<None> Add

Contained UserControls:

UserControl	Name	👁	▦
subscreenForImage1			☑
subscreenForImage2			☑
subscreenForImage3			☑

Background: ▨

Alignment: Vertically

Touch: Flick

☑ Scrollbar visible (not accessible)

☑ Snap to UserControl

□ Automatically hide navigation buttons

□ Animated change

Figure 201: Properties for the container on the <ScreenWithContainer> screen; the pane contains the <subscreenForImage> UserControl three times

2.5.3 Containers

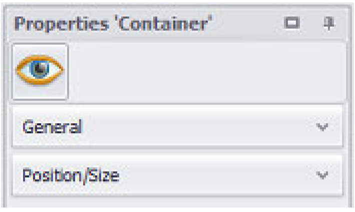
A container contains user controls exclusively. It can contain more than one user control.

One of the special characteristics of containers consists of how they are displayed. A container has a set display size that can be defined, and the UserControls in it can be scrolled through vertically or horizontally at runtime. This makes it possible to implement a swipe gesture, which is prevalent on touch devices these days.
The UserControls in the container should all have the same size. A container can be reused for as many screens as necessary; in addition, containers can be used on any screen, with the exception of UserControls.

This Container visualization object can be viewed as an input or output object, depending on the visualization objects found on the UserControls it contains. The Container visualization object does not feature any states. However, there are functions that can be applied to Container visualization object.

For a detailed example showing how to use this visualization object, please refer to the → "Using containers", page 294 section.

Controls\ tab



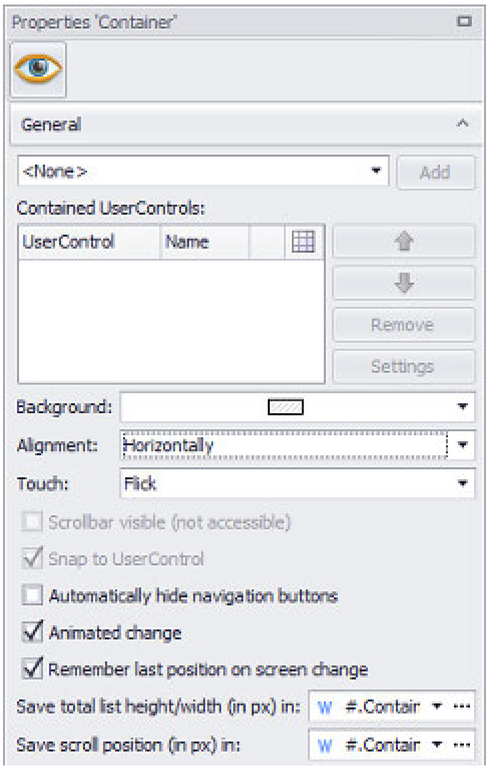
Properties 'Container'

→ "Quick access menu Container", page 304

→ "General", page 299

2.5.3.1 General

Controls\ Container\General



<None>

Contained UserControls:

Background:

Alignment: Horizontally

Touch: Flick

☐ Scrollbar visible (not accessible)

☑ Snap to UserControl

☐ Automatically hide navigation buttons

☑ Animated change

☑ Remember last position on screen change

Save total list height/width (in px) in: #.Contair

Save scroll position (in px) in: #.Contair

Figure 202: Window Properties 'Container'

The field with the **<None>** drop-down menu can be used (only as long as no selection has been made) to directly select a user control from the existing ones in the project tree.

You can also add user controls to a container by dragging and dropping them directly from the project tree.
To do so, select the user control in Project Tree/UserControls, hold the right mouse button down, drag the user control onto the container, and use ⬇ "Add to container" to add it to the container.
Add

This button will only be enabled if a user control is selected previously. Clicking on the Add button will add the selected user control so that it is contained by the container.

User Controls contained

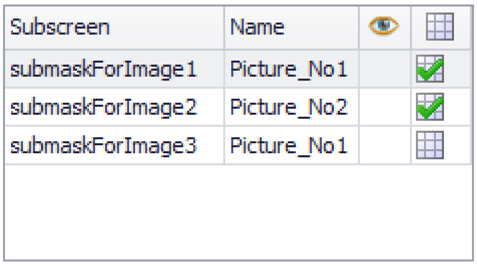
Shows the objects in the container.
A maximum total of 255 User Controls can be added. Outside of this maximum, there is no limit to the number of times a specific user control can be contained in the same container.
The maximum size of all contained User Controls is 32768 px (size with horizontal orientation = width of all User Controls; size with vertical orientation = height of all User Controls)
This is relevant when working with input parameters. For a description of how to make it easier to implement an application by using input parameters, please refer to the→ section "Example with input parameters and a container containing three images", page 295 section.

You can select multiple user controls in the UserControl column.

Name, 👁Visibility, and ▦ Parameters areas

Every user control in the container can be assigned a name in order to make things more manageable. This name will not appear in the Controls Manager, but will instead be assigned to the parameter.

- UserControl: User control found in the container
- Name: Name of "user control" parameter
- ☑: A checkmark indicates that an input parameter has been assigned to the UserControl.



Subscreen	Name	👁	▦
submaskForImage1	Picture_No1		☑
submaskForImage2	Picture_No2		☑
submaskForImage3	Picture_No1		▦

Figure 203: Properties 'Container' pane\General\UserControl with two user controls with input parameters and one without any input parameters

Up/Down ⇧ ⇩

This button will be enabled if a user control is selected first in the UserControl column. The selected user control will be moved one place up every time you click on ⇧ and one place down every time you click on ⇩. The user controls' place in the list determines the order in which they can ultimately be scrolled through in the container.

Parameter

This button will be enabled if a UserControl is selected first in the UserControl column.

Clicking on the Parameters button will open the dialog box

Visibility

This button will be enabled if a UserControl is selected first in the UserControl column. Opens the Visibility tab in the Extended properties dialog box. The settings in the tab can be used to set conditions for the visibility of all selected UserControl; refer to → section "Visibility", page 259

Remove

This button will be enabled if one or more user controls are selected first in the UserControl column. Removes the selected user controls from the container.

Alignment

By selecting the <Horizontally> or <Vertically> option in the Alignment drop-down menu, you can define the direction in which a scrollbar will be shown and/or in which device operators will be able to scroll through the container with a swipe gesture. In order for this to work, the Scrollbar visible checkbox must be enabled and/or the <Flick> or <Drag> option needs to be selected in the Touch drop-down menu.

Horizontally

Enables device operators to scroll through all the UserControls in the container to the left or right with swipe gestures.

Vertically

Enables device operators to scroll through all the UserControls in the container up or down with swipe gestures.

Touch

→ This function is only supported by XV300, XP500 devices and GALILEO Open.

If you select the <Flick> or <Drag> option in the Touch drop-down menu, the container will become a touch zone on the device at runtime, enabling device operators to scroll through all the user controls in the container with flick gestures. The option you select in the Alignment drop-down menu will determine the direction for the flick gestures.

- No Touch
  It will not be possible for the device operator to use flick gestures on the container at runtime. However, you can still add a scrollbar to the container.
- Flick
  Enables device operators to scroll through all the user controls in the container with flick gestures.

The screen will automatically snap to the user controls. If a scrollbar is shown on the container, it will not be possible to use it at runtime.

- Drag
  Enables device operators to scroll through all the user controls in the container with drag gestures.
  The screen will not snap to the user controls automatically. This means that device operators will be able to slide continuously between user controls so that parts of two different adjacent user controls can be shown at the same time.

Show Scrollbar

→ This function is only supported by XV300, XP500 devices and GALILEO Open.

When this option is enabled, a scrollbar will be shown so that device operators can use it (in addition to swipe gestures) to scroll through all the UserControls in the container. If this property is enabled, device operators will only be able to use the scrollbar at runtime if the <Drag> or <No Touch> option is selected in the Touch drop-down menu.

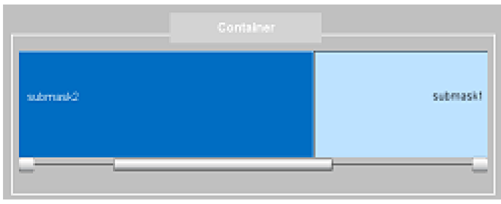


Figure 204: Container with two user controls and horizontal scrollbar

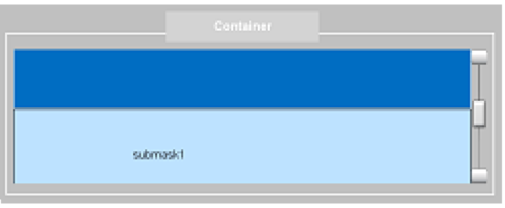


Figure 205: Container with two user controls and vertical scrollbar

Snap to UserControl

This checkbox will be enabled by default. It can be enabled and disabled if the <Drag> option is previously selected in the Touch drop-down menu. It can also be enabled and disabled if the <No Touch> option is first selected in the Touch drop-down menu and the Scrollbar visible checkbox is enabled.

The Snap to UserControl effect will be shown at runtime whenever the device operator "leaves" the touch zone or scrollbar. If the UserControl is not flush with the container on the left or on top at this moment, it will automatically snap to a position where it is.

Automatically hide navigation buttons

If this checkbox is enabled, some of the function keys you add for navigation purposes will be hidden at runtime.

For example, the screen has a function key visualization object that calls the <First Entry> function. If the device operator navigates to the first UserControl at runtime, the function key will be automatically hidden. The same will happen, for instance, with a function key visualization object that calls the <Last Entry> function if the device operator navigates to the last screen at runtime. This basic principle also applies to function keys that call the <Move Next Entry> and <Move Previous Entry> functions.

→ If you use this option, do not configure separate visibility settings for the <Move Next Entry> (and similar) function keys. This can lead to unexpected results.

Animated change

This property can only be enabled if the <Flick> option is selected in the Touch drop-down menu. The term "animated change" means that when a device operator moves user-defined objects at runtime, the transition between two user-defined objects will be clearly shown as a scrolling transition.

If Animated change is not enabled, the user control being shown will be replaced by the new one without a transition animation.

Save last position on screen change

If this option is enabled, the current scroll position within the container will be saved when there is a screen change.
If the user accesses the same screen with the container later on, it will automatically jump to the saved position.

→ This option will only work if nothing changes in terms of the visibility of the individual objects in the container in between the screen changes.

Hide scrollbar indicator

Enabling this checkbox will hide the visual scroll position indicator.

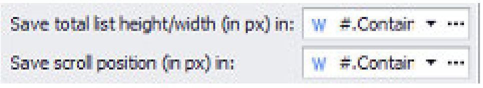
→ This function is only supported by XV30 devices and GALILEO Open.

List height and scroll position

→ This function is only supported by XV30 devices and GALILEO Open.

In order to work with objects for which a list is stored (parameter lists, alarm lists, recipes, and containers) more efficiently, you can save the position in the list.

It is also possible to scroll horizontally through containers depending on how they are configured.

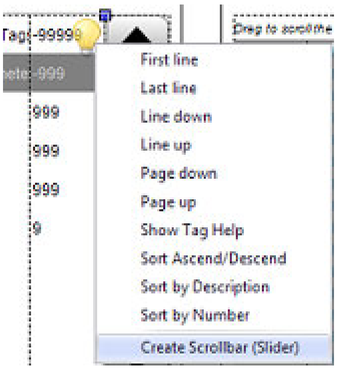


Save total list height/width (in px) in: W #.Contair ▾ ···
Save scroll position (in px) in: W #.Contair ▾ ···

Save list height/width (in px) in:	Saves the total list height or width (total with all items added up) in the specified tag respectively
Save scroll position (in px) in:	Saves the current scroll position within the total list in the specified tag.

Together with Save list height/width (in px) in, this tag can be used, for example, in order to use the slider in a slider object as a scrollbar in the list.

1. Assign tags to the Save list height (in px) in and Save scroll position (in px) in settings
2. In the screen designer, move the cursor to the upper right corner of the object. The 💡 icon will appear
3. Click on the 💡 icon and select the Generate scrollbar (slider) option in the drop-down menu



4. A slider object will be created and the tags will already be assigned correctly.

2.5.3.2 Quick access menu Container

As soon as the cursor moves across the upper right corner of the visualization object, a 💡 light bulb icon for the quick access menu will appear.

The following functions will be available in the quick access menu:

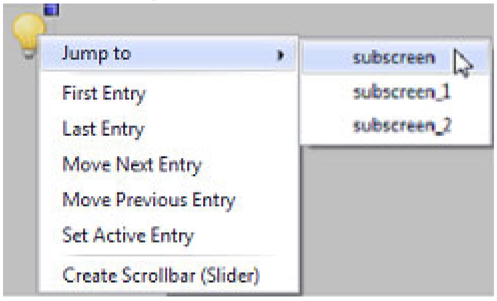


Figure 206: Quick access menu for Container

Go To

Opens a selection list of the UserControls in the container. When you select a UserControl, it will be shown on the screen.

First Entry, Last Entry, Move...

Adds a function key visualization object with the selected function from the Entry group, e.g., First Entry, Move Next Entry, etc.
For a description of these functions, please refer to → section "Container functions", page 305.

Generate scrollbar (slider)

This option will only be available if the relevant tags are filled out for the container object.

Adds a scrollbar to the selected container in the form of a slider object. This means that the scrollbar properties can be configured the same way as they would for a normal slider object.

2.5.4 Container functions

The container functions are as follows:

2.5.4.1 First Entry

This function takes the UserControl at the top of the list and displays it in the target container.

Number of the Function	4611 (1203h)
Executable from the PLC	No
Function with ON/OFF state	No
Parameter 1	Position of target container on screen
Function available in standard screens	Yes
Function available with keyboards	No
Function available in script	No
Function available in application entry script	No

2.5.4.2 Last Entry

This function takes the UserControl at the bottom of the list and displays it in the target container.

Number of the Function	4612 (1204h)
Executable from the PLC	No
Function with ON/OFF state	No
Parameter 1	Position of target container on screen
Function available in standard screens	Yes
Function available with keyboards	No
Function available in script	No
Function available in application entry script	No

2.5.4.3 Move Next Entry

This function takes the next user control in the list and displays it in the target container.

→ Not all device types support this function

Number of the Function	4608 (1200h)
Executable from the PLC	No
Function with ON/OFF state	No
Parameter 1	Position of target container on screen

Function available in standard screens	Yes
Function available with keyboards	No
Function available in script	No
Function available in application entry script	No

2.5.4.4 Move Previous Entry

This function takes the previous UserControl in the list and displays it in the target container.

→ Not all device types support this function

Number of the Function	4609 (1201h)
Executable from the PLC	No
Function with ON/OFF state	No
Parameter 1	Position of target container on screen
Function available in standard screens	Yes
Function available with keyboards	No
Function available in script	No
Function available in application entry script	No

2.5.4.5 Set Active Entry

→ Not all device types support this function

→ This function takes the user control defined in parameter 2 and displays it in the target container.

Number of the Function	4610 (1202h)
Executable from the PLC	No
Function with ON/OFF state	No
Parameter 1	Position of target container on screen
Parameter 2	ID of the user control being displayed
Function available in standard screens	Yes
Function available with keyboards	No
Function available in script	No
Function available in application entry script	No

2.6 Keyboards

GALILEO supports various system keyboards that enable device operators to enter numeric and alphanumeric values and texts on the device at runtime.

The following system keyboards are available:

- Inc/dec 1
- Inc/dec 10
- Numeric
- Numeric Digit Min/Max
- Numeric Min/Max
- Numeric Password
- Alpha numeric
- Alpha numeric ALT
- Alpha numeric ALT CAPS
- Alpha numeric CAPS
- Select Alphanumeric
- Select Numeric
- Select Numeric Digit

The system also provides keyboards for password management (see User Management Functions) and recipe management (see Recipe Functions).

You can also create your own custom keyboards if necessary. These custom keyboards can be created as normal screens in the *Keyboards* folder in *Project Tree\Screens*. The screen type will be set based on the keyboard you want (please refer to Screen settings).

System keyboards and custom keyboards can be selected in the properties for the relevant visualization objects. Please note, however, that the Keyboard property is not available for all visualization objects. It can be selected for the Value Display/Entry, Bargraph, and PLC Text visualization objects.

- Numeric keyboards are used to enter numbers, e.g., for the → " Bargraph", page 356 and → " Value Display/Entry", page 506 visualization objects and for → "User Management", page 891 purposes.
- Alphanumeric keyboards are used to enter text, e.g., for the → " Value Display/Entry", page 506 visualization object, or to enter the user or password used for user management purposes.
- The Keyboards for Password Administration are used to create a user-defined dialog for logging in, changing the password etc..

A separate keyboard layout can be selected for each projected language. So users of different countries can work with their accustomed keyboard layout.

→ Custom keyboards behave the same way as system keyboards. No values will be transferred (and, accordingly no display elements will be updated) as long as a custom keyboard or system keyboard is open.

All system keyboards can be called with a function by using the Function Key visualization object, with the exception of the Select Numeric, Select Numeric Digit, and Select Alphanumeric system keyboards. The reason why these system keyboards are special is described in the section further below.

All system keyboards have a name that reflects the corresponding functions, e.g., → "Alpha numeric Keyboard", page 332.

The appearance of system keyboards can be modified to a certain degree:

- A style can be assigned to the value display and to the buttons
- The window and header background colors can be changed as necessary
- Color of the keyboard outline

→ GALILEO supports custom keyboards, which can be defined and expanded as required.

2.6.1 Select Numeric, Select Numeric Digit

The Select Numeric and Select Numeric Digit keyboards are used exclusively to enter values with an external device

such as a keyboard or barcode scanner. In this case, the external device is connected to the device using the USB port. The Select Numeric and Select Numeric Digit keyboards will not show a keyboard on the screen at runtime. After the visualization object button is tapped on, the values from the external device will be assigned to the visualization object's tags once RETURN is entered, after which these values will be displayed in the visualization object.

The only difference between the Select Numeric and Select Numeric Digit system keyboards is how decimal places are entered with them. In the case of the Select Numeric function, the numbers entered will be inserted into the display from the right (like in a cash register). With the decimal point being fixed at the set position (i.e., in this case, the decimal points cannot be entered using the keyboard). In contrast, when using the Select Numeric Digit function, the decimal point can be inserted with the . key (like with a pocket calculator).

2.6.2 Select Alphanumeric

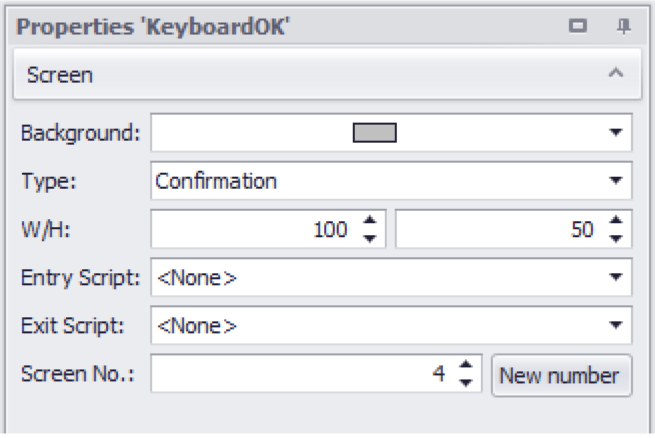
The Select Alphanumeric keyboard is used exclusively to enter characters with an external device

such as a keyboard. In this case, the external device is connected to the device using the USB port. The Select Alphanumeric keyboard will not display a keyboard on the screen at runtime. After the visualization object button is tapped on, the characters from the external device will be assigned to the visualization object's tags once RETURN is entered, after which these characters will be displayed in the visualization object.

2.6.3 Keyboards

This screen type is used to create custom keyboards. Not all object types are permissible in these keyboards. For a description of how to create a custom keyboard, please refer to the → section "User Defined Keyboards", page 316 section.

Project Tree\Screens folder\Keyboards



Properties 'KeyboardOK'

Screen

Background:

Type: Confirmation

W/H: 100 50

Entry Script: <None>

Exit Script: <None>

Screen No.: 4

Figure 207: Properties pane for the "keyboardOK" keyboard

Background color

This setting can be used to set the basic color for the screen.

You can change the color if you want. The following options are available for this purpose:

- Transparent
- Standard Colors
- More Colors...

The most recently used colors will be shown (user-specific).

For detailed information on the available colors, please refer to the following section: → section "Color choice - Object's Color Palette", page 234

Type

Drop-down menu with the various available keyboard types. This includes custom keyboards and all system keyboards, e.g., Alphanumeric.

Keyboard types limit the locations in the project where the corresponding keyboards will be available for selection.

For example, keyboards of type <User Management: Change Password> are available in the drop-down list used to select the keyboard used to change passwords (*Home\Project Configuration\User Management*⚙ *\General\Keyboards section).*

Keyboards of type <User Management: Change Group> will not be available in the drop-down list.

In addition, the special functions available will also depend on the selected keyboard type.

→ If one of the custom keyboards is not available in a drop-down menu where you want it to be available, this probably means that you selected the wrong keyboard type when creating the keyboard.

W/H

Screen width and height, in pixels.

Entry Script / Exit Script

Entry Script

This function is used to run an event script when the screen is opened (before the screen is displayed).
The event script specified in the Entry Script field will be run before the screen is opened.

Exit Script

This function is used to run an event script when the screen is exited.
The event script specified in the Exit Script field will be run before the screen is closed.

For a description of event scripts, please refer to the → section " Scripts", page 952 section.

Mask number New number

When created in a project, every screen will automatically be assigned a screen number as a unique ID. A project cannot contain more than one screen with the same number. When assigning a screen number manually, you can assign any number of your choice.

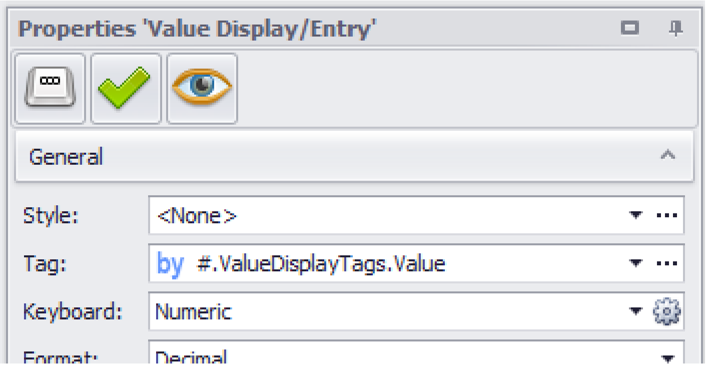
These screen numbers become relevant when the PLC triggers a screen change or when the PLC needs to determine which screen is currently being shown by the device. The way in which the PLC can trigger a screen change is described in detail in a different section – please refer to → section "Controlling functions with the PLC", page 602.

If you click on New number, the next free screen number in the project will be automatically entered into the Screen No. field. Screen numbers freed up when a screen is deleted will also be taken into account when determining the next free screen number.

2.6.4 System Keyboards

GALILEO provides various system keyboards that enable device operators to enter numeric and alphanumeric values and texts on the device at runtime (e.g., numeric keyboard). These system keyboards will be used automatically without any special configuration if a keyboard is selected for an object (e.g., value display):

Object tab\ 1.2


Properties 'Value Display/Entry'

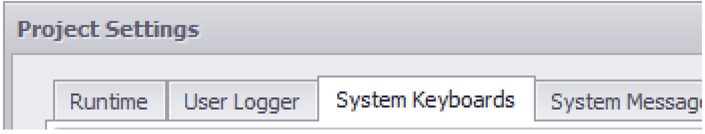
General

Style: <None>
Tag: #.ValueDisplayTags.Value
Keyboard: Numeric
Format: Decimal

Figure 208: *Properties pane as an example*

The display settings for all system keyboards can be configured in the System Keyboards tab in Project Configuration. All settings, with the except of positioning settings, will also apply to custom keyboards.

Home\\Project Configuration\ *Settings tab*


Project Settings

Runtime | User Logger | System Keyboards | System Messag

Figure 209: *Project Settings/System Keyboards window*

There are two different types of keyboard designs:

Classical

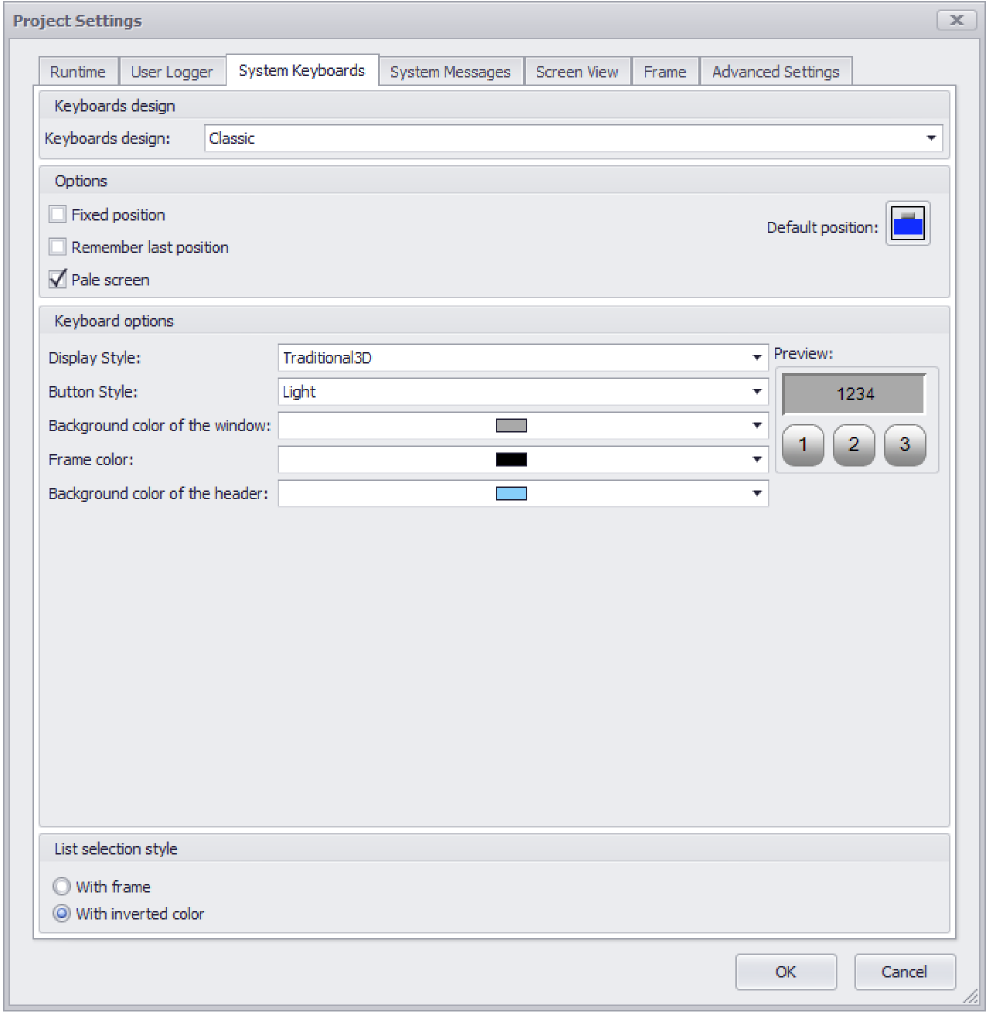
The classic keyboard layout and functionality
that has been used for all keyboards up to GALILEO Version 10.5

Extended

All options in the keyboard design drop-down menu with the exception of "Classic"
Available on GALILEO Version 10.5 and higher
Various designs and expanded functionality,
e.g., cursor positioning for entering values,
cursor function keys when using custom keyboards

Keyboard design Classic

Home\Project Configuration\ Settings tab



Keyboards design

Keyboards design: Classic

Options

□ Fixed position

□ Remember last position

☑ Pale screen

Default position:

Keyboard options

Display Style: Traditional3D

Button Style: Light

Background color of the window: ▨

Frame color: ■

Background color of the header: ▭

List selection style

○ With frame

● With inverted color

Figure 210: *Project Settings/System Keyboards window*

Options

Fixed Position

If this option is enabled, an active keyboard cannot be moved around

Remember last position

If this option is enabled, the keyboard's position will be saved when closed. The next time the same keyboard is opened, it will be automatically moved to this saved position.

Gray out screen

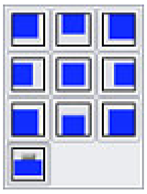
If this option is enabled, the view behind the keyboard will be dimmed.

Default position:

Used to specify the keyboard's default position relative to the object calling it. The alignments available for selection include "top left," "center center," "bottom center," etc. plus "docked."
If you select the "docked" option, the Galileo runtime system will decide where the keyboard can best be displayed. If the "Remember last position" option is enabled, this setting will only be applied the very first time the keyboard is opened.

Default position

The keyboard will be positioned based on the option selected here. The default position on the screen can be defined by clicking inside the selection box as follows:



Top left , top center or top right Center left , center center or center right Bottom left , bottom center or bottom right or Docked on the button

Keyboard options

Display style: Used to select a style from the project for showing "display" objects

Button style: Used to select a style from the project for showing "button" objects

Background color of the window: The background color for the keyboard window on the screen.

Frame color: The color for the keyboard's frame

Background color of the header: A title can be assigned to certain system keyboards. In the case of numeric keyboards, this color will also be the background color for the limit value display, e.g., blue.

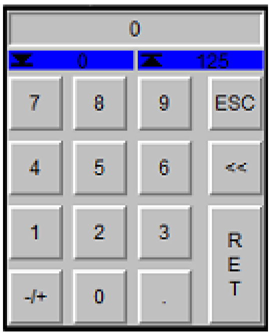


Figure 211: Numeric keyboard with display field <0>

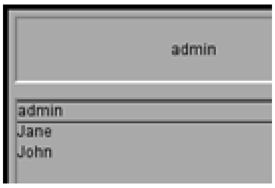
Preview:

Generates a preview for the currently configured settings.

List selection style

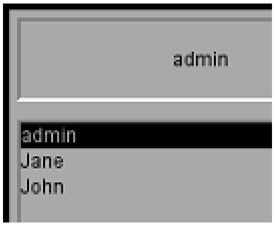
◉ One of the two options needs to be selected

With frame The selected item will be surrounded with a box.



With inverted color

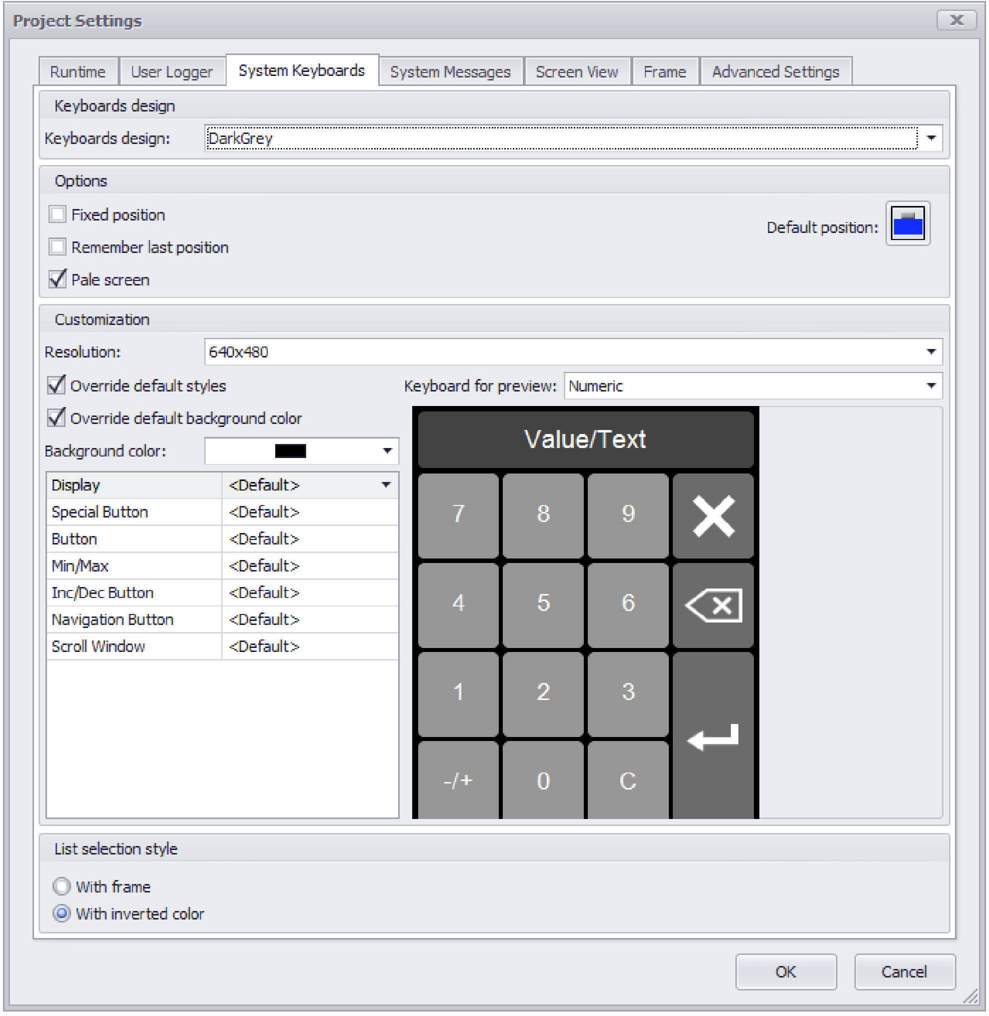
The selected item will be highlighted by inverting the corresponding colors.



Keyboard design Exhanced

All options in the keyboard design drop-down menu with the exception of "Classic"

Home\Project Configuration\ Settings tab



Project Settings

Runtime | User Logger | System Keyboards | System Messages | Screen View | Frame | Advanced Settings

Keyboards design

Keyboards design: DarkGrey

Options

□ Fixed position

□ Remember last position

☑ Pale screen

Default position:

Customization

Resolution: 640x480

☑ Override default styles

Keyboard for preview: Numeric

☑ Override default background color

Background color: ■

Display	<Default>
Special Button	<Default>
Button	<Default>
Min/Max	<Default>
Inc/Dec Button	<Default>
Navigation Button	<Default>
Scroll Window	<Default>

List selection style

○ With frame

● With inverted color

Figure 212: *Project Settings/System Keyboards window*

Options

Fixed Position — If this option is enabled, an active keyboard cannot be moved around

Remember last position — If this option is enabled, the keyboard's position will be saved when closed. The next time the same keyboard is opened, it will be automatically moved to this saved position.

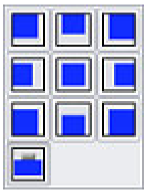
Gray out screen — If this option is enabled, the view behind the keyboard will be dimmed.

Default position: — Used to specify the keyboard's default position relative to the object calling it. The alignments available for selection include "top left," "center center," "bottom center," etc. plus "docked."
If you select the "docked" option, the Galileo runtime system will decide where the

keyboard can best be displayed. If the "Remember last position" option is enabled, this setting will only be applied the very first time the keyboard is opened.

Default position

The keyboard will be positioned based on the option selected here. The default position on the screen can be defined by clicking inside the selection box as follows:



Top left , top center or top right Center left , center center or center right Bottom left , bottom center or bottom right or Docked on the button

Customization

Resolution:	Can be used to select the resolution for the keyboard preview. (The settings do not refer to the resolution set up here)
Overwrite default style:	Each keyboard design defines its own default style. If you enable this option, it will be possible to overwrite styles for displays, buttons, etc.
Overwrite default background color:	Each keyboard design defines its own default background color. If you enable this option, it will be possible to overwrite the background color.
Back color:	Used to select the background color
Display/Default table	Configuration of elements
Keyboard for preview:	Used to select the keyboard type for the preview while taking the current configuration into account.

How can a system keyboard be positioned?

When a system keyboard is opened, the position configured in *Home\Project Configuration\Settings\System Keyboards\Positioning* will be used.

Device operators can change the position of a system keyboard at runtime by touching the display field for the current value and moving the keyboard by dragging & dropping it on the device screen.

2.6.5 User Defined Keyboards

GALILEO supports various system keyboards that enable device operators to enter numeric and alphanumeric values and texts on the device at runtime. In addition, the system provides special keyboards that can be used for user management and recipe management purposes.

You can also create your own custom keyboards if necessary. These custom keyboards can be created as normalkeyboard screens in the *Screens\Keyboards* folder in the project tree. The screen type will be set based on the keyboard you want (please refer to → "Screen types", page 263).

When a custom keyboard is opened, it will assume all the settings (with the exception of the position) configured in *Home\Settings\System Keyboards*. For a description of these properties, please refer to the → section "System Keyboards", page 311 section.

How can a custom keyboard be positioned?

Device operators can change the position of a custom keyboard at runtime by touching the display field for the current value and moving the keyboard by dragging & dropping it on the device screen. In order for this to work, however, a display field or a touch-sensitive field must have been set up. A display field can be implemented with a function key visualization object that calls the Display 'Value/Text' function in the Characters group. Meanwhile, a touch-sensitive field can be implemented with a function key visualization object that calls the Key 'Window Move' function in the Characters group.

When a keyboard is opened, the position configured in *Home\Settings\System Keyboards\Positioning* will be used. The following functions can also be used to change the keyboard position on the device in real time: → section "Display 'Value/Text'", page 322 and → "Key 'Window Move'", page 327

If a custom keyboard is drawn with a "transparent" background, the visualization objects on the screen underneath it will be visible. Non-overlapping objects will be redrawn if the PLC values change, for example.

- Numeric keyboards are used to enter numbers, e.g., for the bargraph and value display/entry objects and for user management purposes.
- Alphanumeric Keyboards are used for entering texts such as with the PLC Text object or for entering the user or password in the User Management.
- The Keyboards for Password Administration are used to create a user-defined dialog for logging in, changing the password etc..

→ Custom screens behave like system screens., No values will be transferred (and, accordingly no display elements will be updated) as long as a custom keyboard or system keyboard is open. The following applies to system screens:
Within system screens, the screen used to call the system screen is still considered the "current" screen.
Tags on the "current" screen will be transferred even though a system screen is open.

Using a system keyboard as a template

The easiest way to create a custom keyboard is to use a system keyboard as a template and modify it as necessary for your needs.

To create a custom keyboard, follow the steps below:

- Go to *Project tree\Screens\Keyboards*.
- Import one of the system keyboards using the following command sequence: *Keyboards\Context menu\Import system keyboard*, e.g., <UserM:Change PW>.

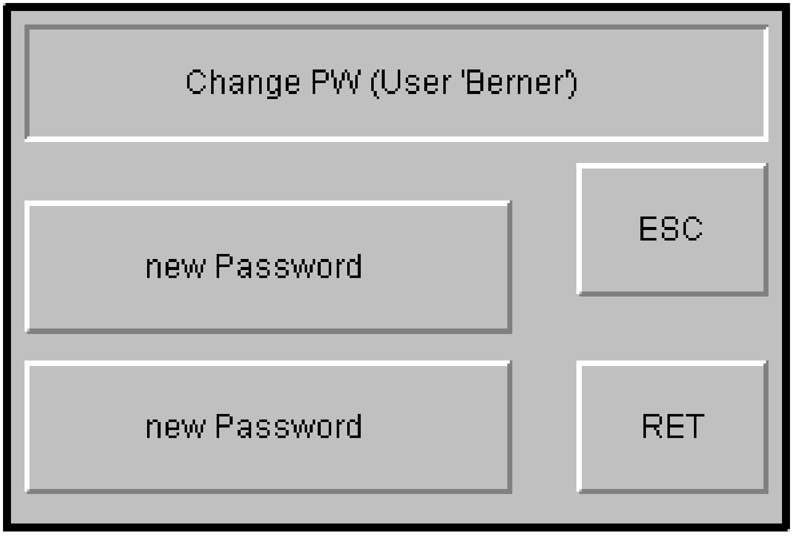


Figure 213: System keyboard appearing after the function key is tapped on

- Rename the keyboard screen, e.g., <myChangePW>.
- Redesign the keyboard screen as necessary, e.g., by deleting or adding function keys.

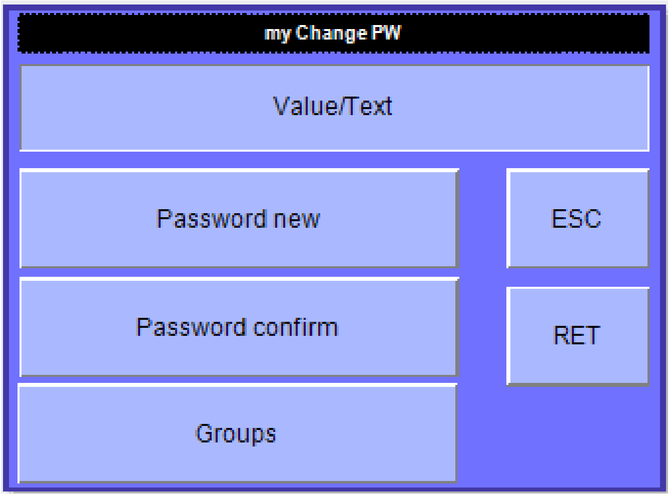


Figure 214: <myChangePW> custom keyboard with <Value/Text> display field

- Call the custom keyboard at the location you want.

→ If you have already created a style, you can use it when importing the system keyboard. The buttons will change to match the style; please refer to → section "Style management", page 458 .

2.6.5.1 Numeric Keyboards

The following functions in the <Characters> group are available for creating a numeric keyboard (number input):

→ "Display 'Minimum'", page 322
→ "Display 'Maximum'", page 322
Key '<<' (Backspace) -→ "Display 'Value/Text'", page 322
Key 'Clear'
Key 'Dec 1'
Key 'Dec 10'
Key 'ESC' → "Display 'Value/Text'", page 322
Key 'Inc1'
Key 'Inc10'
→ "Key "number"", page 324
→ "Key "number" with "RET"", page 324
Key 'RET' - Display 'Value/Text'
Key 'RET' (R\nE\nT)
→ "Key "RET" and "Call Script Function"", page 325
Key 'SIGN'
Key 'UNICODE'"0" to key "9"
→ "Key 'Window Move'", page 327
→ "Inc/Dec 1", page 334 and → "Inc/Dec 10", page 335
"ESC" key
"RET" key
→ "Key "RET" and "Call Script Function"", page 325
"Backspace" key
"." key
Call Script Function on Mask Entry

2.6.5.2 Alphanumeric Keyboards

The following functions are available for creating an alphanumeric keyboard (text input):

"a" key to "z" key
"A" key to "Z" key
"ESC" key
"RET" key
Key "RET" with "Call Script Function"
"Backspace" key
"Clear" key
Various punctuation characters such as "."key, ":" key, ";" key, "," key, "!" key and "#" key
"@" key
Key "number"

Key "number" with "RET"
Call Script Function on Mask Entry

2.6.5.3 User management keyboards

The following keyboards are available for user management functions:

UserM: "Lock"
UserM: "Password"
UserM: "Password confirm"
UserM: "Password new"
UserM: "Group Selection"
UserM: "Sel. Group -userdef"
UserM: "User Selection"
UserM: "Sel. User -userdef"
UserM: "Unlock"
UserM: "User"

2.6.6 Character input functions (characters/stream characters)

Every button in a system keyboard or custom keyboard is also available individually as a function.

Certain buttons are also available as stream characters. The difference between "normal" buttons and stream buttons is as follows:

- Stream button values are transmitted immediately after the corresponding button is tapped on, i.e., each character in a value will be transmitted as soon as it is entered.
  For example: If the value for a tag is set to 1234, the tag will assume the following values in succession: 1, 12, 123, and 1234. Likewise, these values will also be directly transmitted in succession to the PLC.
- In contrast, normal button values are transmitted only after the RET button is tapped on. Likewise, these values will also be transmitted to the PLC only after the RET button is tapped on.

2.6.7 Available buttons

UNICODE characters

(Space), << (Backspace) , C (Clear) , Dec 1 , Dec 10, Inc 1 , Inc 10 , ESC , RET , +/- (Sign)

The following buttons are also available as stream buttons:

0 … 9

<< (Backspace) , C (Clear) , Inc 1 , Inc 10 , Dec 1 , Dec 10 , +/- (Sign)

Functions that are no longer supported in GALILEO:

Display "Value" in Hex
Display "Minimum" in Hex
Display "Maximum" in Hex
Display "Value/Text" Password

2.6.7.1 Dec 1 (assigned)

Tapping on this function key will decrease the tag value by 1. This function is only available as a stream button (when a stream button is tapped on, the corresponding tag value is immediately transmitted to the PLC after the corresponding change is made).

Number of the Function	1681 (691h)
Executable from the PLC	No
Function with ON/OFF state	No
Parameter entered	Name of the Tags
Function available in standard screens	Yes
Function available with keyboards	No
Function available in script	No
Function available in application entry script	No

2.6.7.2 Dec 10 (assigned)

Tapping on this function key will decrease the tag value by 10. This function is only available as a stream button (when a stream button is tapped on, the corresponding tag value is immediately transmitted to the PLC after the corresponding change is made).

Number of the Function	1682 (692h)
Executable from the PLC	No
Function with ON/OFF state	No
Parameter entered	Name of the Tags
Function available in standard screens	Yes
Function available with keyboards	No
Function available in script	No
Function available in application entry script	No

2.6.7.3 Display 'Scroll Window'

Display of a list in a User Defined Keyboards. The list is filled with appropriate data depending on the type of user -defined keyboard.

Number of the Function	1172 (494h)
Executable from the PLC	No
Function with ON/OFF state	No
Function available in standard screens	No
Function available with keyboards	Yes
Function available in script	No
Function available in application entry script	No

2.6.7.4 Display 'Maximum'

Shows the maximum value of a numeric User Defined Keyboards.

Number of the Function	1184 (4A0h)
Executable from the PLC	No
Function with ON/OFF state	No
Function available in standard screens	No
Function available with keyboards	Yes
Function available in script	No
Function available in application entry script	No

2.6.7.5 Display 'Minimum'

Shows the minimum value of a numeric User Defined Keyboard.

Number of the Function	1183 (49Fh)
Executable from the PLC	No
Function with ON/OFF state	No
Function available in standard screens	No
Function available with keyboards	Yes
Function available in script	No
Function available in application entry script	No

2.6.7.6 Display 'Value/Text'

Show the text/value entered in a numeric User Defined Keyboards. If the device operator touches this display field at runtime, they will be able to move the keyboard anywhere in the screen by dragging & dropping it. The keyboard's position can also be changed with the Key 'Window Move' function.

→ The background setting of this object must NOT be «Transparent».

Number of the Function	1185 (4A1h)
Executable from the PLC	No
Function with ON/OFF state	No
Function available in standard screens	No
Function available with keyboards	Yes
Function available in script	No
Function available in application entry script	No

2.6.7.7 Inc 1 (assigned)

Tapping on this function key will increase the tag value by 1. This function is only available as a stream button (when a stream button is tapped on, the corresponding tag value is immediately transmitted to the PLC after the corresponding change is made).

Number of the Function	1683 (693h)
Executable from the PLC	No
Function with ON/OFF state	No
Parameter entered	Name of the Tags

Function available in standard screens	Yes
Function available with keyboards	No
Function available in script	No
Function available in application entry script	No

2.6.7.8 Inc 10 (assigned)

Tapping on this function key will increase the tag value by 10. This function is only available as a stream button (when a stream button is tapped on, the corresponding tag value is immediately transmitted to the PLC after the corresponding change is made).

Number of the Function	1684 (694h)
Executable from the PLC	No
Function with ON/OFF state	No
Parameter entered	Name of the Tags
Function available in standard screens	Yes
Function available with keyboards	No
Function available in script	No
Function available in application entry script	No

2.6.7.9 Key "Alt Layout" User

This function is used to switch to a user defined keyboard.

Number of the Function	1222 (4A6h)
Executable from the PLC	No
Function with ON/OFF state	No
Parameter 1	User defined keyboard
Function available in standard screens	No
Function available with keyboards	Yes
Function available in script	No
Function available in application entry script	No

2.6.7.10 Key "AltGr" User

This function is used to switch to a user defined keyboard.

Number of the Function	1198 (4AEh)
Executable from the PLC	No
Function with ON/OFF state	No
Parameter 1	User defined keyboard
Function available in standard screens	No
Function available with keyboards	Yes
Function available in script	No
Function available in application entry script	No

2.6.7.11 Key "CAPS" User

This function is used to switch to a user defined keyboard.

Number of the Function	1187 (4A3h)
Executable from the PLC	No

Function with ON/OFF state	No
Parameter 1	User defined keyboard
Function available in standard screens	No
Function available with keyboards	Yes
Function available in script	No
Function available in application entry script	No

2.6.7.12 Key "Ctrl" User (Terminal)

This function is used to switch to a user defined keyboard.

Number of the Function	1221 (4A5h)
Executable from the PLC	No
Function with ON/OFF state	No
Parameter 1	User defined keyboard
Function available in standard screens	No
Function available with keyboards	Yes (type "Terminal")
Function available in script	No
Function available in application entry script	No

2.6.7.13 Key "number"

This function is used to enter a numeric value in conjunction with custom keyboards. In contrast to the Key "UNICODE" function, this function can be assigned any numeric value of data type FLOAT. The numeric value is applied by using the RETURN function.

Number of the Function	1191 (4A7h)
Executable from the PLC	No
Function with ON/OFF state	No
Parameter 1	Number from -3.402823466*10^{38} to +3.402823466*10^{38}
Function available in standard screens	Yes
Function available with keyboards	Yes
Function available in script	No
Function available in application entry script	No

2.6.7.14 Key "number" with "RET"

Just like the Key "number" function, this function is also used to enter any numeric value of data type FLOAT, except this function also triggers the RETURN button (OK). If this function is used within a custom keyboard, the numeric value will be applied and the keyboard will be closed.

Number of the Function	1193 (4A9h)
Executable from the PLC	No
Function with ON/OFF state	No
Parameter 1	Number from -3.402823466*10^{38} to +3.402823466*10^{38}
Function available in standard screens	Yes
Function available with keyboards	Yes
Function available in script	No
Function available in application entry script	No

2.6.7.15 Key "RET" and "Call Script Function"

This function automatically initiates the RETURN key (OK), and the event script specified as a parameter is called directly afterwards. Different functions can be carried out within the Event Script. If this function is used within a User Defined Keyboards, the numeric value is accepted, the keyboard is closed and the event script is called.

Number of the Function	1195 (4ABh)
Executable from the PLC	No
Function with ON/OFF state	No
Parameter entered	Name of the script
Function available in standard screens	Yes
Function available with keyboards	Yes
Function available in script	No
Function available in application entry script	No

2.6.7.16 Key "Scroll cursor left"

When this key is tapped, the input cursor will move one place to the left.
This function can only be used with alphanumeric keyboards.

Number of the Function	0x04C9
Executable from the PLC	No
Function with ON/OFF state	No
Function available in standard screens	No
Function available with keyboards	Yes
Function available in script	No
Function available in application entry script	No

2.6.7.17 Key "Scroll cursor right"

When this key is tapped, the input cursor will move one place to the right.
This function can only be used with alphanumeric keyboards.

Number of the Function	0x04CA
Executable from the PLC	No
Function with ON/OFF state	No
Function available in standard screens	No
Function available with keyboards	Yes
Function available in script	No
Function available in application entry script	No

2.6.7.18 Key "Scroll cursor to begin"

When this key is tapped, the input cursor will move to the starting position.
This function can only be used with alphanumeric keyboards.

Number of the Function	0x04CB
Executable from the PLC	No
Function with ON/OFF state	No
Function available in standard screens	No

Function available with keyboards	Yes
Function available in script	No
Function available in application entry script	No

2.6.7.19 Key "Scroll cursor to end"

When this key is tapped, the input cursor will move to the end position.
This function can only be used with alphanumeric keyboards.

Number of the Function	0x04CC
Executable from the PLC	No
Function with ON/OFF state	No
Function available in standard screens	No
Function available with keyboards	Yes
Function available in script	No
Function available in application entry script	No

2.6.7.20 Key "string"

This function is used to enter a text (string) in conjunction with custom keyboards. In contrast to the "Key a", "Key b", etc. functions, this function can be assigned any string. The string is applied using the RETURN function. Alphanumeric keyboards are used for text input in conjunction with, e.g., PLC Text visualization objects, password input, and user input.

Number of the Function	1192 (4A8h)
Executable from the PLC	No
Function with ON/OFF state	No
Function available in standard screens	Yes
Function available with keyboards	Yes
Function available in script	No
Function available in application entry script	No

2.6.7.21 Key "string" with "RET"

This function is also used to enter a text (string), in the same way as the function Key "string" however, this function also activates the RETURN key (OK). If this function is used within a custom keyboard, the numeric value will be applied and the keyboard will be closed.

Number of the Function	1194 (4AAh)
Executable from the PLC	No
Function with ON/OFF state	No
Function available in standard screens	Yes
Function available with keyboards	Yes
Function available in script	No
Function available in application entry script	No

2.6.7.22 Key "UNICODE"

This function is used to enter a character in conjunction with alphanumeric and numeric custom keyboards. This function can be used to assign any UNICODE value.

The string is applied using the RETURN function. Alphanumeric keyboards are used for text input in conjunction with, e.g., PLC Text visualization objects, password input, and user input.

Number of the Function	1196 (4ACh)
Executable from the PLC	No
Function with ON/OFF state	No
Parameter entered	UNICODE value of a character
Function available in standard screens	Yes
Function available with keyboards	Yes
Function available in script	No
Function available in application entry script	No

2.6.7.23 Key 'Window Move'

This function can be used to change the keyboard position within the entire screen. If the device operator touches this touch-sensitive field at runtime, they will be able to move the keyboard anywhere in the screen by dragging & dropping it.

In the GALILEO application, the function key visualization object will display the name of the function; on the device, it will not show any text.

The keyboard's position can also be changed with the Display 'Value/Text' function.

Number of the Function	1186 (4A2h)
Executable from the PLC	No
Function with ON/OFF state	No
Function available in standard screens	No
Function available with keyboards	Yes
Function available in script	No
Function available in application entry script	No

2.6.7.24 Keyb. Mask CANCEL + Call Script

This function calls the event script and sends a negative feedback to the calling function, so that this is not executed. This function can only be used in conjunction with the user-defined keyboards.

Number of the Function	1218 (4C2h)
Executable from the PLC	No
Function with ON/OFF state	No
Parameter entered	Name of the script
Function available in standard screens	No
Function available with keyboards	Yes
Function available in script	No
Function available in application entry script	No

2.6.7.25 Keyboard Mask OK + Call Script

This function calls the event script and sends a positive signal to the calling function so that this is not executed. This function can only be used in conjunction with the user-defined keyboards.

Number of the Function	1217 (4C1h)
Executable from the PLC	No
Function with ON/OFF state	No
Parameter entered	Name of the script
Function available in standard screens	No
Function available with keyboards	Yes
Function available in script	No
Function available in application entry script	No

2.6.7.26 Keyboard Mask YES

This function is identical to Keyboard Mask OK, with the exception that, for the Load Recipe Entry function, the current recipe will be saved before the new recipe is loaded.

This function can only be used in conjunction with User Defined Keyboards for the confirmation check.

Number of the Function	1199 (4AFh)
Executable from the PLC	No
Function with ON/OFF state	No
Function available in standard screens	No
Function available with keyboards	Yes
Function available in script	No
Function available in application entry script	No

2.6.7.27 Keyboard Mask YES + Call Script

This function is identical to Keyboard Mask OK + Call Script, with the exception that, for the Load Recipe Entry function, the current recipe will be saved before the new recipe is loaded.

This function can only be used in conjunction with User Defined Keyboards for the confirmation check.

Number of the Function	1216 (4C0h)
Executable from the PLC	No
Function with ON/OFF state	No
Parameter entered	Name of the script
Function available in standard screens	No
Function available with keyboards	Yes
Function available in script	No
Function available in application entry script	No

2.6.7.28 Keyboard Mask CANCEL

This function sends a negative signal to the calling function so that this is not executed. This function can only be used in conjunction with the user-defined keyboards.

Number of the Function	1165 (48dh)
Executable from the PLC	No

Function with ON/OFF state	No
Function available in standard screens	No
Function available with keyboards	Yes
Function available in script	No
Function available in application entry script	No

2.6.7.29 Keyboard Mask Display

This function corresponds to the display of a value (numeric keyboards) and string (alphanumeric keyboards) within Keyboards. This function can only be used in conjunction with the user-defined keyboards.

Number of the Function	1164 (48ch)
Executable from the PLC	No
Function with ON/OFF state	No
Function available in standard screens	No
Function available with keyboards	Yes
Function available in script	No
Function available in application entry script	No

2.6.7.30 Keyboard Mask OK

This function sends a positive signal to the calling function so that this is not executed. This function can only be used in conjunction with the user-defined keyboards.

Number of the Function	1166 (48Eh)
Executable from the PLC	No
Function with ON/OFF state	No
Function available in standard screens	No
Function available with keyboards	Yes
Function available in script	No
Function available in application entry script	No

2.6.7.31 UserM: "Group Sel. -userdef"

This function is used in custom password keyboards in order to select a group. When run, the corresponding custom keyboard will appear.

Number of the Function	1202 (4B2h)
Executable from the PLC	No
Function with ON/OFF state	No
Parameter 1	User defined keyboard
Function available in standard screens	No
Function available with keyboards	Yes
Function available in script	No
Function available in application entry script	No

2.6.7.32 UserM: "Group Selection"

This function is used in custom password keyboards in order to select a group.

Number of the Function	1177 (499h)

Executable from the PLC	No
Function with ON/OFF state	No
Function available in standard screens	No
Function available with keyboards	Yes
Function available in script	No
Function available in application entry script	No

2.6.7.33 UserM: "Lock"

The function is used in user-defined password keyboards for locking a user.

Number of the Function	1203 (4B3h)
Executable from the PLC	No
Function with ON/OFF state	No
Function available in standard screens	No
Function available with keyboards	Yes
Function available in script	No
Function available in application entry script	No

2.6.7.34 UserM: "Password"

This function is used in custom password keyboards in order to enter the password. When run, the keyboard selected in *Home\User Management\General\Keyboards* will appear so that the password can be entered.

Number of the Function	1178 (49Ah)
Executable from the PLC	No
Function with ON/OFF state	No
Function available in standard screens	No
Function available with keyboards	Yes
Function available in script	No
Function available in application entry script	No

2.6.7.35 UserM: "Password confirm"

This function is used in custom password keyboards, in conjunction with password changes, in order to enter the new password again for confirmation purposes. When run, the keyboard selected in *Home\User Management\General\Keyboards* will appear.

Number of the Function	1180 (49ch)
Executable from the PLC	No
Function with ON/OFF state	No
Function available in standard screens	No
Function available with keyboards	Yes
Function available in script	No
Function available in application entry script	No

2.6.7.36 UserM: "Password new"

This function is used in custom password keyboards, in conjunction with password changes, in order to enter the new password. When run, the keyboard selected in

Home\User Management\General\Keyboards will appear so that the password can be changed.

Number of the Function	1179 (49Bh)
Executable from the PLC	No
Function with ON/OFF state	No
Function available in standard screens	No
Function available with keyboards	Yes
Function available in script	No
Function available in application entry script	No

2.6.7.37 UserM: "UnLock"

This function is used in user-defined password keyboards for unlocking a user.

Number of the Function	1204 (4B4h)
Executable from the PLC	No
Function with ON/OFF state	No
Function available in standard screens	No
Function available with keyboards	Yes
Function available in script	No
Function available in application entry script	No

2.6.7.38 UserM: "User"

This function is used in custom password keyboards in order to enter a user. When run, the keyboard selected in *Home\User Management\General\Keyboards* will appear.

Number of the Function	1176 (498h)
Executable from the PLC	No
Function with ON/OFF state	No
Function available in standard screens	No
Function available with keyboards	Yes
Function available in script	No
Function available in application entry script	No

2.6.7.39 UserM: "Sel. User -userdef"

This function is used in custom password keyboards in order to select a user. When run, the corresponding custom keyboard will appear.

This function is used in custom password keyboards in order to select a user. When run, the corresponding custom keyboard will appear.

Number of the Function	1201 (4B1h)
Executable from the PLC	No
Function with ON/OFF state	No
Parameter 1	User defined keyboard
Function available in standard screens	No
Function available with keyboards	Yes
Function available in script	No
Function available in application entry script	No

2.6.7.40 UserM: "Select User"

This function is used in custom password keyboards in order to select a user.

Number of the Function	1200 (4B0h)
Executable from the PLC	No
Function with ON/OFF state	No
Function available in standard screens	No
Function available with keyboards	Yes
Function available in script	No
Function available in application entry script	No

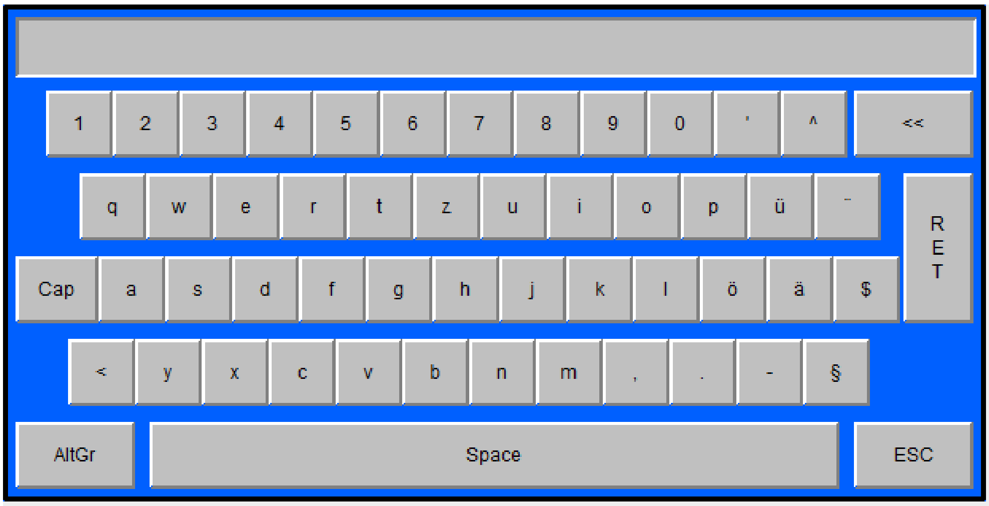
2.6.8 Keyboard functions

2.6.8.1 Alpha numeric Keyboard

This function displays the alphanumeric system keyboard for the current project language. There are multiple system keyboard layouts available, such as en_US, en_UK, de_CH, and de_DE. The default language for the project and the corresponding system keyboard layout can be configured in *Home\Languages*.

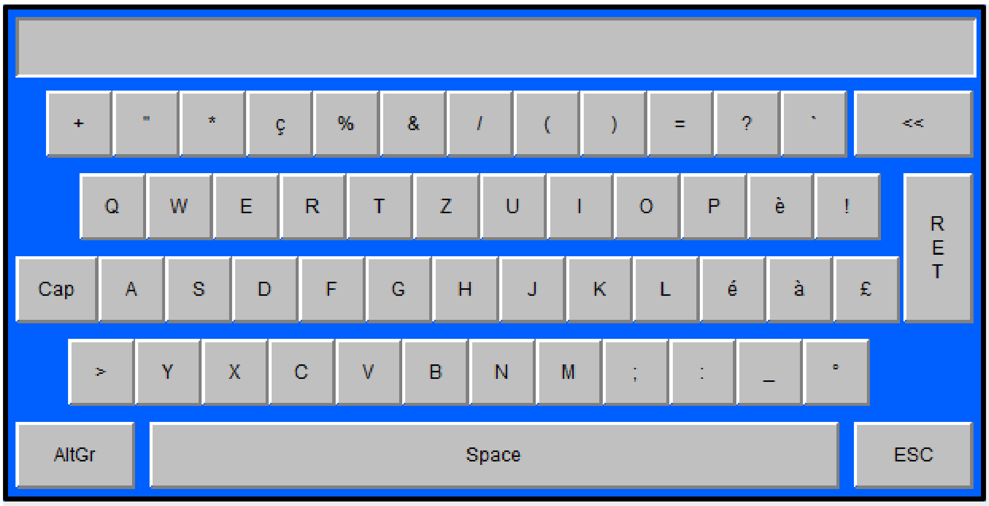
Number of the Function	260 (104h)
Executable from the PLC	No
Function with ON/OFF state	No
Type of return value	String
Function available in standard screens	Yes
Function available with keyboards	No
Function available in script	No
Function available in application entry script	No

Example showing an alphanumeric keyboard with the "de_CH" system keyboard layout.



Cap key

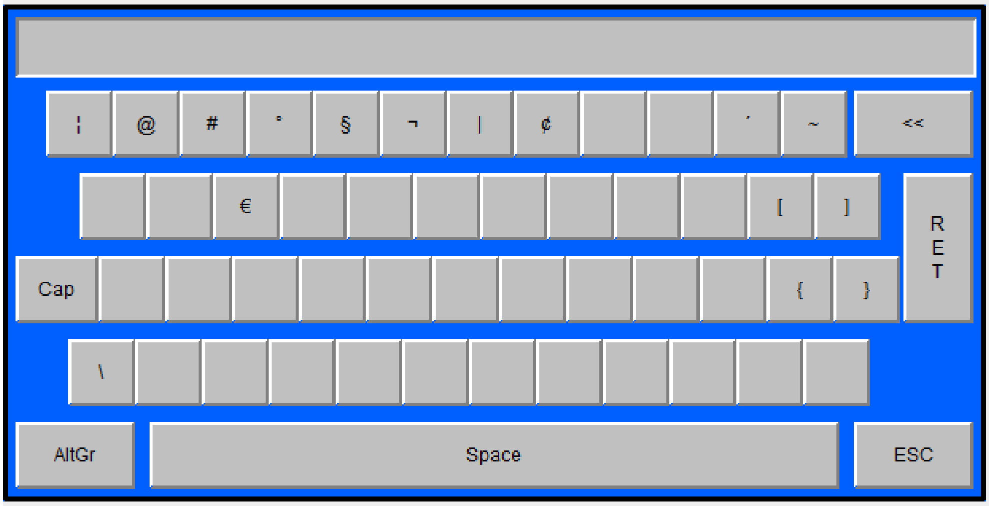
The button functions will change as follows after the Cap key is pressed.



If the Cap key is tapped on again, the button functions will switch back to lower case letters. Tapping on two keys at the same time is not supported.

AltGr button

The button functions will change as follows after the AltGr button is pressed.

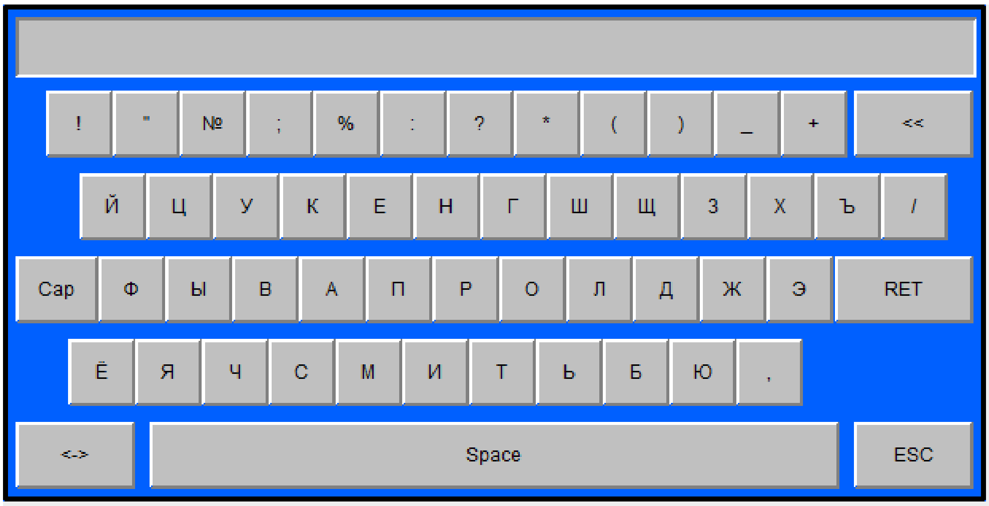


If the AltGr key is tapped on again, the button functions will return to the letters being displayed previously. Tapping on two keys at the same time is not supported.

Alternative layout button

Certain system keyboards support an alternative layout in order to make it possible to use different characters
(example: Russian keyboard with Cyrillic and Latin characters).

The button functions will change as follows after the alternative layout button is pressed.



Users can enter any text using uppercase and lowercase letters, special characters, and numbers.

These functions can be used to enter recipe names or PLC texts, for example.

2.6.8.2 Call user defined Keyboard

The Call User defined Keyboard function is used for input purposes with custom keyboards. The specific custom keyboard must be selected as the argument.

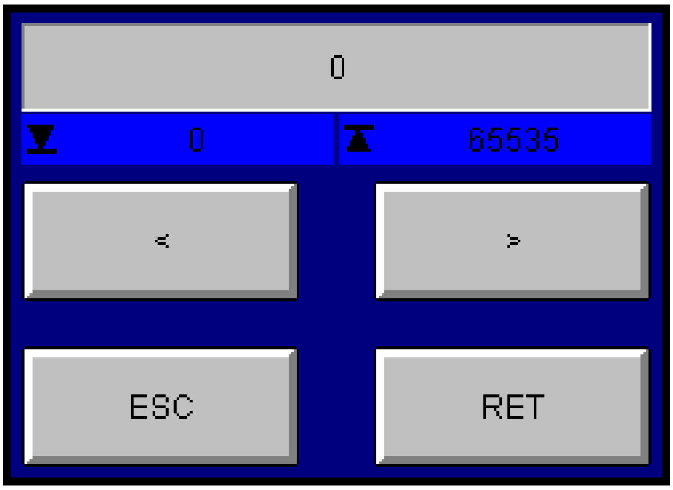
The Call User defined Keyboard function is used for input purposes with custom keyboards. The specific custom keyboard must be selected as the argument.

Number of the Function	378 (17Ah)
Executable from the PLC	No
Function with ON/OFF state	No
Parameter 1	User defined keyboard
Function available in standard screens	Yes
Function available with keyboards	No
Function available in script	No
Function available in application entry script	No

2.6.8.3 Inc/Dec 1

This function can be used to increment and decrement the value of a value display/entry or bargraph visualization object by a value of one.

Number of the Function	261 (105h)
Executable from the PLC	No
Function with ON/OFF state	No
Type of return value	Number
Function available in standard screens	Yes
Function available with keyboards	No
Function available in script	No
Function available in application entry script	No



The Minimum (shown on the left) and the Maximum (shown on the right) limit the setting range. Tapping on the RET button will transmit the value to the PLC. If the keyboard is closed with ESC, the value will not be applied and will not be transmitted to the PLC.

If a conversion is used for a tag value, the increment / decrement will be applied to the converted value.

Incrementing example

Say that the PLC value is "1000" and that a conversion rule that shows the value as "100" has been enabled in GALILEO.
If you were to run "Inc 1", the HMI value of "100" would be incremented by 1 to "101".
Meanwhile, the PLC value would change to "1010" due to the conversion rule.

→ Having a higher value resolution on Galileo than on the PLC side does not make much sense. In this case, incrementing a value by 1 would result in the value not changing on the PLC.

Value resolution example

Say that the PLC value is "100" and that a conversion rule that shows the value as "1000" has been enabled in GALILEO. Also say that this value has a higher value resolution than the PLC can handle.
This means that an HMI value of "1001" would be automatically rounded to "100" for the PLC.
Accordingly, running the "Inc 1" function would not change the PLC value.
Meanwhile, the PLC value would change to "1010" due to the conversion rule.

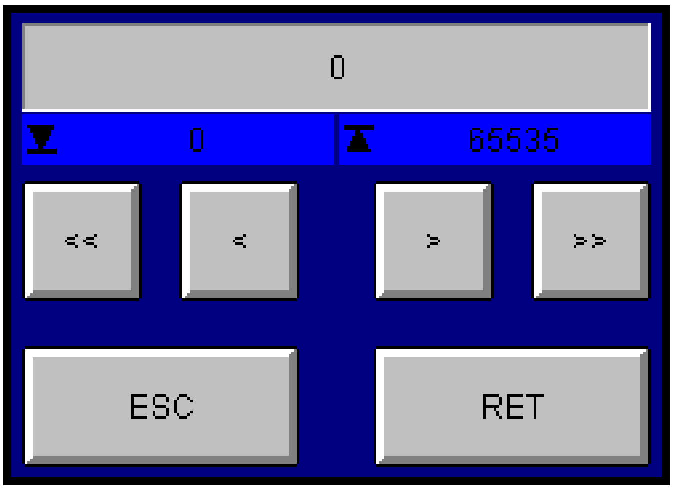
2.6.8.4 Inc/Dec 10

This function can be used to change the value of a value display/entry or bargraph visualization object by a value of one or ten.

Number of the Function	262 (106h)
Executable from the PLC	No
Function with ON/OFF state	No

Type of return value	Number
Function available in standard screens	Yes
Function available with keyboards	No
Function available in script	No
Function available in application entry script	No

The Minimum (shown on the left) and the Maximum (shown on the right) limit the setting range. Tapping on the RET button will transmit the value to the PLC. If the keyboard is closed with ESC, the value will not be applied and will not be transmitted to the PLC.



If a conversion is used for a tag value, the increment / decrement will be applied to the converted value.

Conversion factor > 1:100

Incrementing example

Say that the PLC value is "1000" and that a conversion rule that shows the value as "100" has been enabled in GALILEO.
If you were to run "Inc 10", the HMI value of "100" would be incremented by 1 to "110".
Meanwhile, the PLC value would change to "1100" due to the conversion rule.

→ Having a higher value resolution on GALILEO than on the PLC side does not make much sense. In this case, incrementing a value by 1 would result in the value not changing on the PLC.

Value resolution example

Say that the PLC value is "100" and that a conversion rule that shows the value as "1000" has been enabled in GALILEO. If the conversion factor > 1:100,
this value has a higher value resolution than the PLC can handle.
This means that an HMI value of "1010" would be automatically rounded to "1000" for the PLC.
Accordingly, running the "Inc 10" function would not change the PLC value.

2.6.8.5 None

If the None keyboard is selected for a value display/entry or bargraph visualization object,
it will not be possible to change the value on the device,
as a keyboard will not appear when the object is tapped on and the value will not be selected.

Number of the Function	256 (100h)
Executable from the PLC	No
Function with ON/OFF state	No
Function available in standard screens	Yes
Function available with keyboards	No
Function available in script	No
Function available in application entry script	No

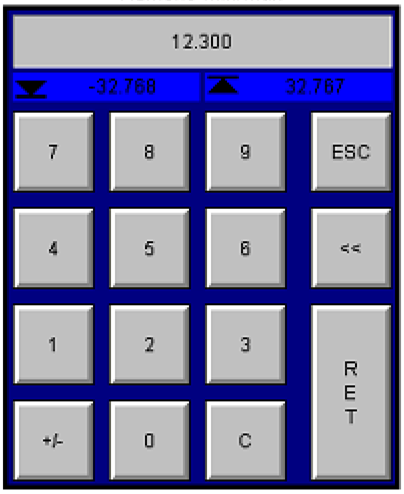
2.6.8.6 Numeric keyboards

Numeric, Numeric Min/Max, Numeric Digit Min/Max

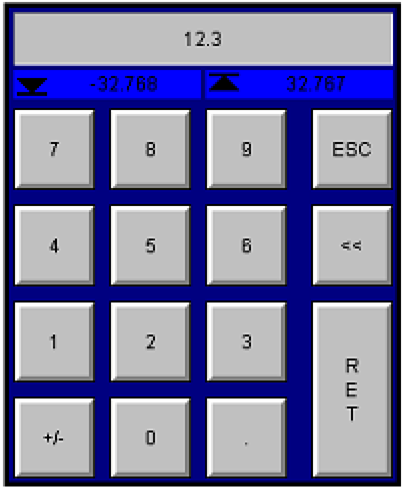
These keyboards can be used to change the values of visualization objects, e.g., value display/entry and bargraph objects. For example, they can be selected in the properties for the relevant visualization object, in the Keyboard field.

Number of the Function	Numeric:	257 (101h)
	Numeric Min/Max:	258 (102h)
	Numeric Digit Min/Max:	265 (109h)
Executable from the PLC		No
Function with ON/OFF state		No
Type of return value		Number
Function available in standard screens		Yes
Function available with keyboards		No
Function available in script		No
Function available in application entry script		No

Numeric Min/Max



Numeric Digit Min/Max



The only difference between the Numeric Min/Max and Numeric Digit Min/Max keyboards is in the way decimal places are entered. With the Numeric and Numeric Min/Max keyboards, numbers are entered into the display from the right (like in a cash register), with the decimal point being fixed at the set position. With the Numeric Digit Min/Max keyboard, the decimal point can be entered as required with the "." key (like with a pocket calculator).

The following example explains this difference:

Entry range	: -32,768…32,767
Number of decimal places	: 3

The following input is required in order to enter a value of 12,300 with the keyboards:

Numeric, Numeric Min/Max	: 1 2 3 0 0
Numeric Digit Min/Max	: 1 2 . 3

If a value falls outside the input range when the RET key is tapped on, it will be set to the minimum or maximum (as appropriate) instead.

Following are the actions that will be triggered when a key is tapped on:

- RET transfers the value to the PLC.
- ESC discards the value and aborts the transfer to the PLC
- << deletes the last figure
- C zeros the value (not possible with Numeric Digit Min/Max)
- . Inserts a decimal point (only available for Numeric Digit Min/Max)
- +/- the value is negated (if signed)

→ The Numeric and Numeric Min/Max keyboards are different from each other in that the input range (minimum and maximum) is not displayed for Numeric.

2.6.8.7 Numeric Password

This keyboard is identical to the Numeric keyboard, with the exception that an asterisk (*) will be shown instead of the numbers entered.

This keyboard is identical to the Numeric keyboard, with the exception that an asterisk (<*>) will be shown instead of the numbers entered.

Number of the Function	263 (107h)
Executable from the PLC	No
Function with ON/OFF state	No
Type of return value	Number
Function available in standard screens	Yes
Function available with keyboards	No
Function available in script	No
Function available in application entry script	No

2.6.8.8 Select Numeric, Select Numeric Digit

If the Select Numeric or Select Numeric Digit keyboard is selected for a value display/entry, bargraph, or PLC text visualization object,
the value for the object will be selected when the latter is tapped on.

It will then be possible to change this selected value with function key visualization objects with a function from the Keyboards or Characters/StreamCharacters group. The buttons required to make the changes (Inc/Dec, individual buttons or keyboard) must be found directly on the screen.

Tapping on the object again will undo the selection.

The only difference between the Select Numeric and Select Numeric Digit system keyboards is how decimal places are entered with them.
In the case of the Select Numeric function, the numbers entered will be inserted into the display from the right (like in a cash register).
The decimal point will be at a fixed position.
In contrast, when using the Select Numeric Digit function, the decimal point can be inserted with the . key (like with a pocket calculator).

Number of the Function	Select Numeric:	266 (10Ah)
	Select Numeric Digit:	268 (10ch)
Executable from the PLC		No
Function with ON/OFF state		No
Function available in standard screens		Yes
Function available with keyboards		No
Function available in script		No
Function available in application entry script		No

The Call user defined Keyboard function can be called to use a custom keyboard for input purposes.

2.6.9 Using confirmation prompts to confirm whether functions should be run

The purpose of confirmation prompts is to make sure that functions are not run immediately. Instead, they require a confirmation from the relevant device operator before the corresponding function is executed. This also means that confirmation prompts are only available for certain functions used in the recipe and graph visualization objects. If you want to set up a confirmation prompt, you will need to set it up in Home\Recipes\Confirmation tab for *recipe* tag objects and in Home\Graph Blocks\Archive tab\Confirmation on use of function for *graph* tag objects.

The confirmation prompt itself GALILEOneeds to be set up on a screen of type Keyboards.

→ Please note that confirmation prompts can only be set up on keyboards of type Confirmation.

Keyboards can be designed according to your specific needs, with the only limitation being that only certain visualization objects are allowed.

For a detailed description of the keyboard screens, please refer to the → "Keyboards", page 307 section

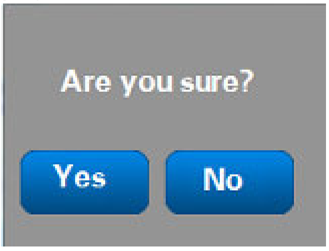


Figure 215: Keyboard screen being displayed on device screen. Yes will run the function, while No will keep the function from being run.

Example Confirmation prompt Off DemoApplication2.prj

Say you have a graph called PlcData, and you want a confirmation prompt to be shown before the graph loads and displays an archive file called <PlcDataArchive>.

Prerequisites
At least one archive file must be available, e.g., test2015.GDB.

To set up a confirmation prompt to be shown before a function is run, follow the steps below:

- Go to *Project Tree\Screens\Keyboards*.
- Open the context menu and select the Add new screen option, then enter a name for the new keyboard screen, e.g., <confirm>.
- Double-click on the keyboard screen to open it.
- Select the Confirmation option in the Type drop-down menu found in the properties for the <screen>.
- Click on *Controls\Text*.
- Position the object
  Move the cursor to the screen and draw the area for the visualization object.

- Write the text you want to use for the confirmation prompt into the *Properties 'Text'\Text field*.

You have now created the required screen, which now contains the text for the confirmation prompt.

Say you want the available confirmation prompt answers to be <Yes> and <No>. To do this, you will need to add two function keys to the keyboard screen so that these keys can be used to call the necessary functions, e.g., Keyboard Mask OK and Keyboard Mask CANCEL. Please note that these functions will only be available for keyboards of type Confirmation.

- Click on *Controls\Dynamic Controls\Function Key*.
- Position the object
  Move the cursor to the screen and draw the area for the visualization object.
- In the *Properties 'Function Key'\Group field*, select the group you want, e.g., <Characters>.
- In the *Properties 'Function Key'\Function field*, select the Keyboard Mask OK function.
- Click on *Controls\Dynamic Controls\Function Key*.
- Position the object
  Move the cursor to the screen and draw the area for the visualization object.
- In the *Properties 'Function Key'\Group field*, select the group you want, e.g., <Characters>.
- In the *Properties 'Function Key'\Function field*, select the Keyboard Mask CANCEL function.

To link the keyboard screen with the confirmation prompt to the function you want, follow the steps below:

- Select the tag object, e.g., *Home\Graph Blocks\Graph Blocks\<graph>\Archive tab\Confirmation on use of function*.
- Open the drop-down menu for the field corresponding to the function for which you want there to be a confirmation prompt.
- In the field corresponding to the function for which you want there to be a confirmation prompt, select the keyboard screen you created previously. For example, select the confirm keyboard screen in the <Load Archive File> field.
- Click on OK.
- Start the Galileo Simulator with function key F5 or
  by clicking on Home\ . Build.

Testing the confirmation prompt example at runtime

To test the confirmation prompt in the DemoApplication2.prj project, follow the steps below:

- ▶ Run the DemoApplication2.prj and switch from the start screen to the GraphDisplay screen.

The graph visualization object will display the current tag values.

- ▶ Scroll through the bar containing the Show marker line button until you reach the Load button.
- ▶ Click on the Stop button.

Please note that archive files can only be loaded when graph plotting is stopped. Otherwise, GALILEO will show an error.

- ▶ Click on the Load button.

A list of archive files available for selection will be displayed.

- ▶ Select the archive file you want, e.g., <test2015.GDB>.
- ▶ Confirm with button RET.
- ▶ The confirmation prompt you set up will appear. Click on No.

The graph visualization object will continue to display the current tag values.

- ▶ Run the test again, except click on YES this time.

The graph visualization object will load the test.GDB archive file and display the corresponding tag values.

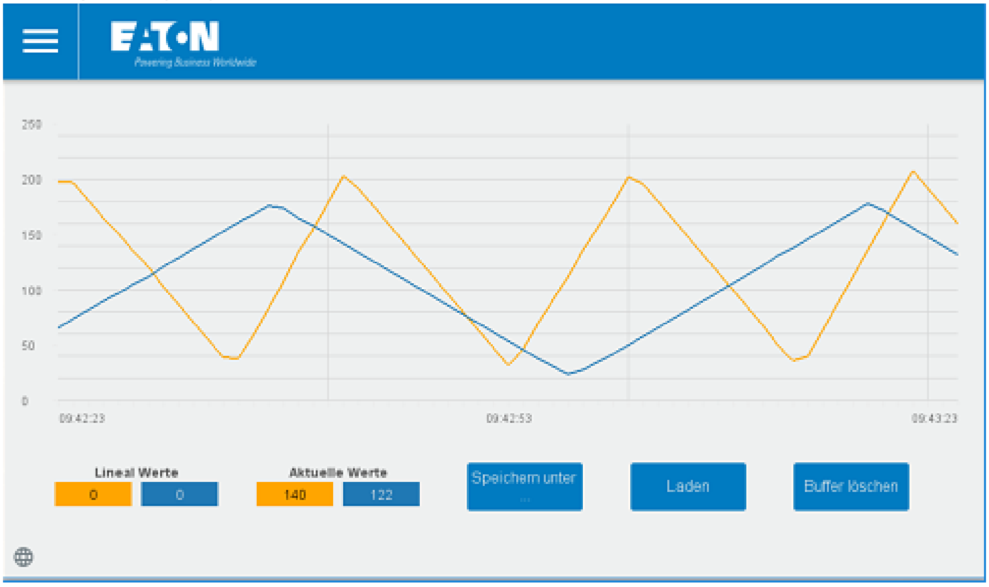
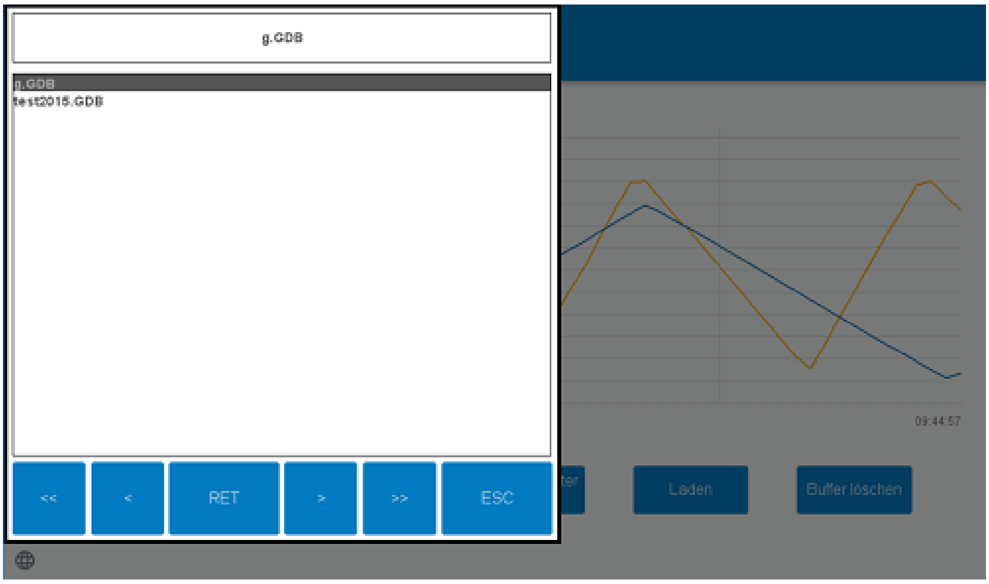


Figure 216: DemoApplication2.prj simulation with GraphDisplay screen and Load button.



g.GDB

g.GDB
test2015.GDB

Figure 217: Screen showing archive files available for selection

2.7 Dialog boxes for interactive communication

Usually, an application will have several screens in order to provide the user with information regarding the application that is currently running.
However, a dialog box that involves actual interaction with the user is required in order to:

- Display specific information for the user
- Detect user confirmations
- Both display and acquire information

Two different types can be designed in GALILEO: modal and modeless dialog boxes.

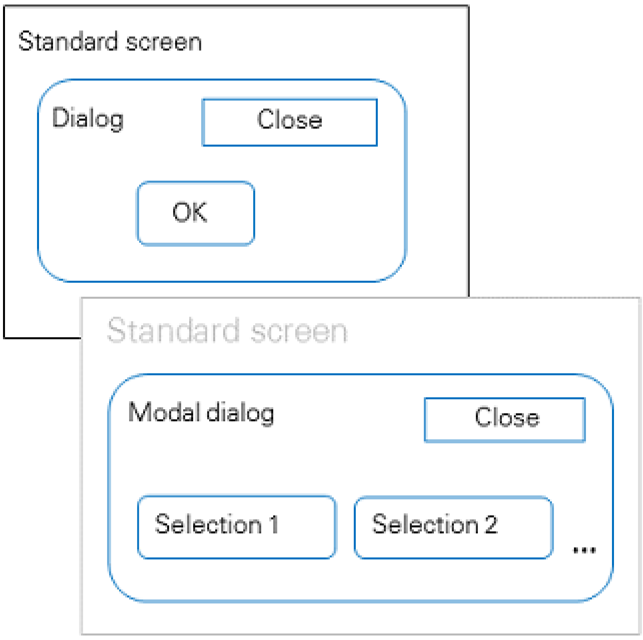
A modal dialog box is shown when input is required from a user in order to continue with an operation. Since this confirmation is required from the user, it is possible to disable the other screens in the application and gray them out in order to indicate this as long as the dialog box screen is open.

In most cases, a modal confirmation dialog box will ask the user to tap either the OK or the Cancel button. In this case, tapping the OK button means that the user – after entering data or making a selection if applicable – wants the application to continue. Meanwhile, tapping the Cancel button means that the user wants to cancel the function completely.

In contrast, modeless dialog boxes do not prevent a user from activating other application views while the dialog box is open, and usually contain at least a Close button that can be used to close the dialog box. However, this type of dialog box can also be used to offer additional functions.
For example, a modeless dialog box could consist of a user control that would be hidden and shown on the screen itself (please refer to the figure below). While the user control is being shown on the screen, it would continue to be possible to use other objects on the screen.

These dialog boxes are made up of several elements:

- A standard screen that requires the dialog box
- User controls as dialog box screens containing the dialog box's content
- Scripts with the functions that the dialog box should provide access to (not absolutely necessary)



A dialog box can be started from the screen itself or when a script signals an event, such as a specific value being reached.

If necessary, a script can be used to log the acquisition of the user's selection.

To add a dialog box to a standard screen, the dialog button visualization object needs to be used.

A dialog box can contain the same visualization objects and special functions as a standard screen. In addition, the "Close dialog" special function, which closes the current dialog box and returns to the standard screen that called it, is available as well.

The "Open dialog" and "Close dialog" functions are also available in event scripts, in which case the settings for parameters, position, graying out, help screens, entry script, exit script, moving/fixed, etc. need to be defined with parameters.

Nesting dialog boxes is allowed, although this is limited to a maximum of five hierarchy levels.
This means that you can design free (i.e., guided by the user) hierarchically structured dialog boxes.

2.7.1 Creating and using dialog boxes

To integrate a dialog box into the application, follow the steps below:

1. Design the dialog box as a user control;
   configure the actions you want (e.g., "Yes," "No," "Continue," etc.)
2. Call the dialog box
   Either with the dialog button visualization object or with a script

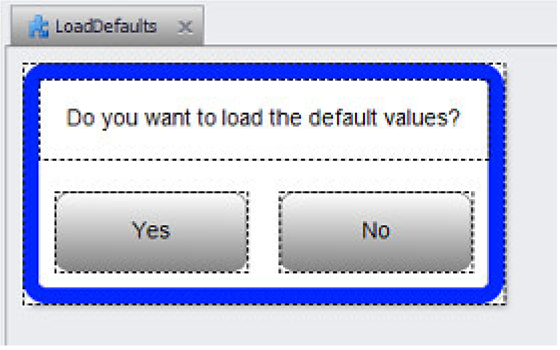
The following two examples illustrate possible use cases for dialog boxes.

2.7.1.1 Confirming an action with a dialog button

Use case: Say that you want the user to explicitly confirm that they want to load a recipe entry.

Procedure:

1. Create a user control named **<LoadDefaults>**. This user control will be used as a dialog box:

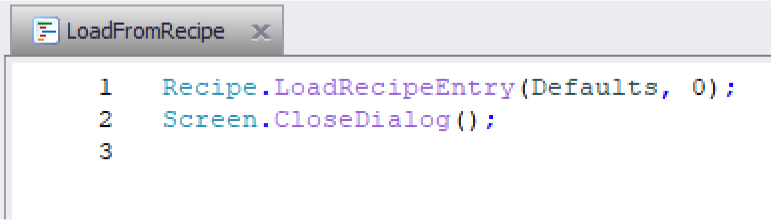


Do you want to load the default values?

Figure 218: User control named **<LoadDefaults>**

The Yes and No buttons are both of the function key type.

2. Create a script named **<LoadFromRecipe>** that runs the action after the confirmation:

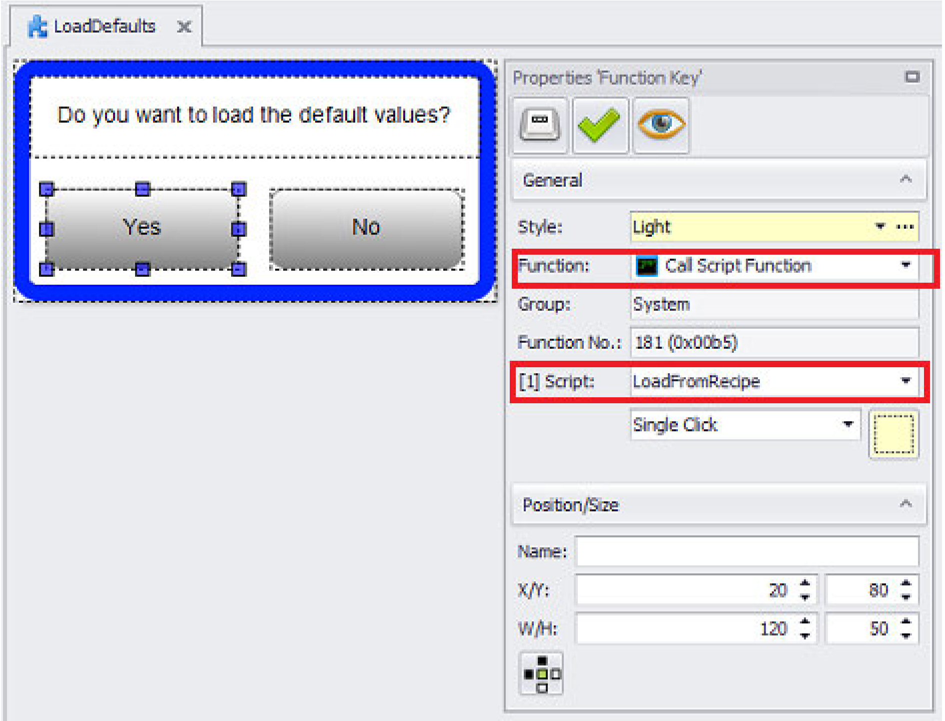


```
Recipe.LoadRecipeEntry(Defaults, 0);
Screen.CloseDialog();
```

Figure 219: Content of **<LoadFromRecipe>** script

The sequence at runtime will be as follows:

3. Switch back to the user control.
   Configure the Yes function key in such a way that the script you just created will be called:


LoadDefaults
Do you want to load the default values?
Yes
No
Properties 'Function Key'
General
Position/Size

Style: Light

Function: Call Script Function

Group: System

Function No.: 181 (0x00b5)

[1] Script: LoadFromRecipe

Single Click

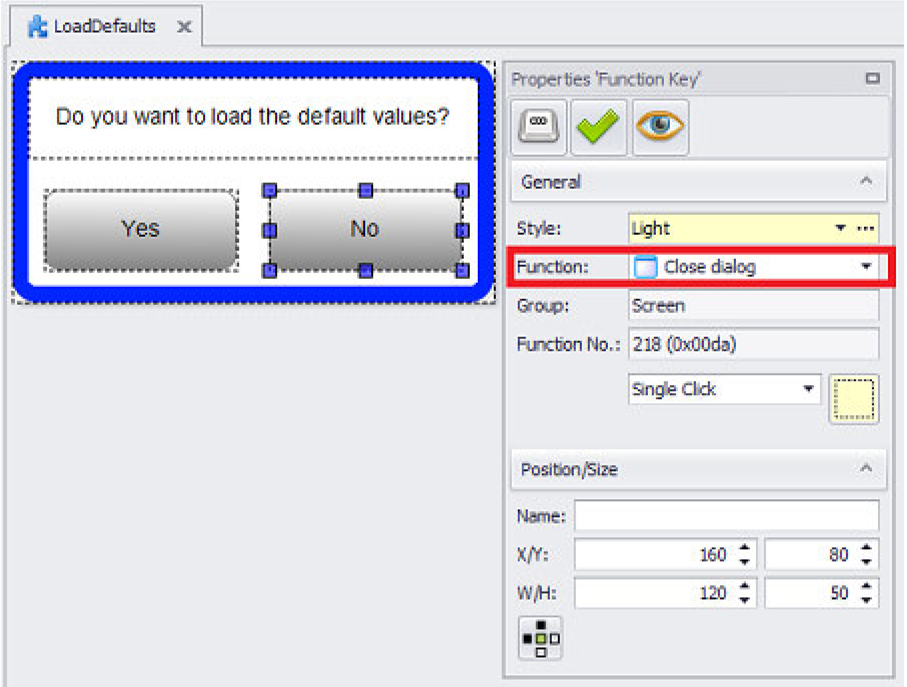
Name:

X/Y: 20 80

W/H: 120 50

Figure 220: **<LoadFromRecipe>** script call

4. Configure the No function key in such a way that the Close dialog function will be called:


LoadDefaults
Do you want to load the default values?
Yes
No
Properties 'Function Key'
General
Position/Size

Style: Light

Function: Close dialog

Group: Screen

Function No.: 218 (0x00da)

Single Click

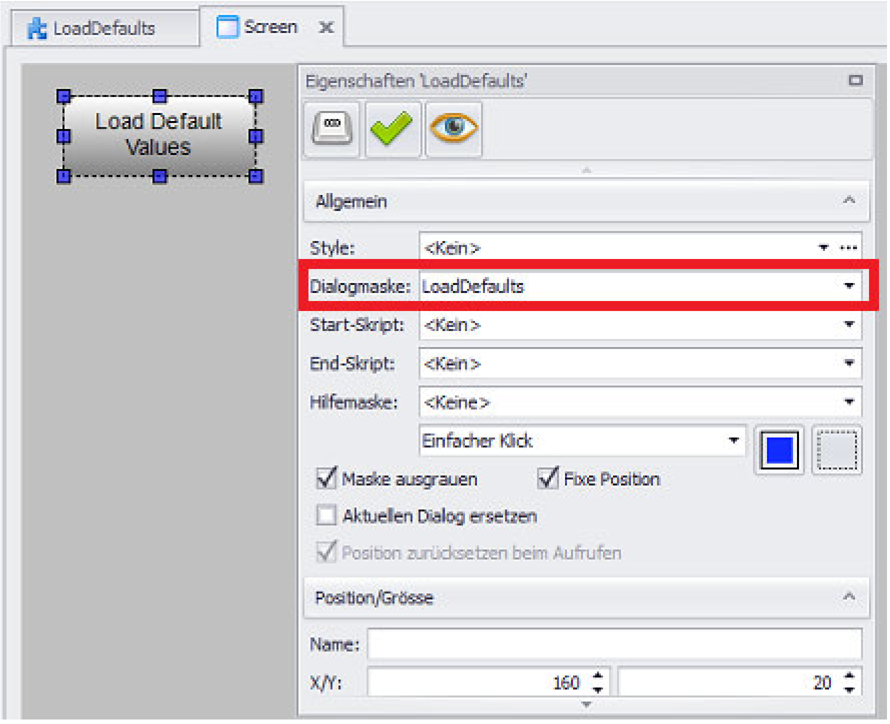
Name:

X/Y: 160 80

W/H: 120 50

Figure 221: Call Close Dialog

5. Place a dialog button object on a standard screen and assign the dialog box you just created:



LoadDefaults | Screen ×

Load Default Values

Eigenschaften 'LoadDefaults'

Allgemein

Style: <Kein>

Dialogmaske: LoadDefaults

Start-Skript: <Kein>

End-Skript: <Kein>

Hilfemaske: <Keine>

Einfacher Klick

☑ Maske ausgrauen ☑ Fixe Position

☐ Aktuellen Dialog ersetzen

☑ Position zurücksetzen beim Aufrufen

Position/Grösse

Name:

X/Y: 160 20

Figure 222: Call LoadDefaults

The sequence at runtime will be as follows:

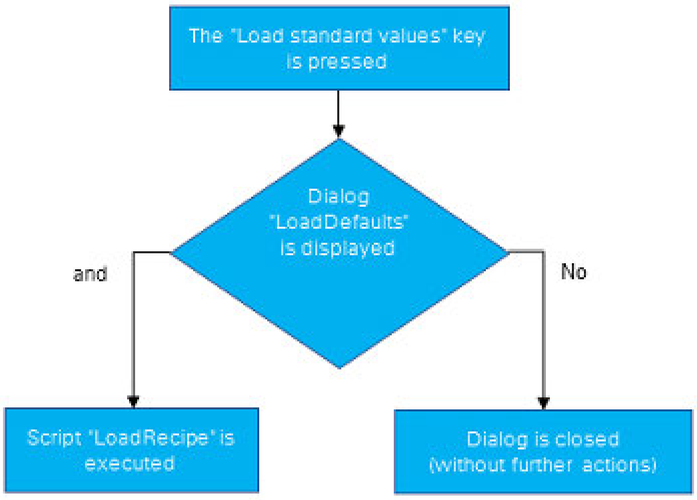

The "Load standard values" key is pressed
Dialog "LoadDefaults" is displayed
and
No
Script "LoadRecipe" is executed
Dialog is closed (without further actions)

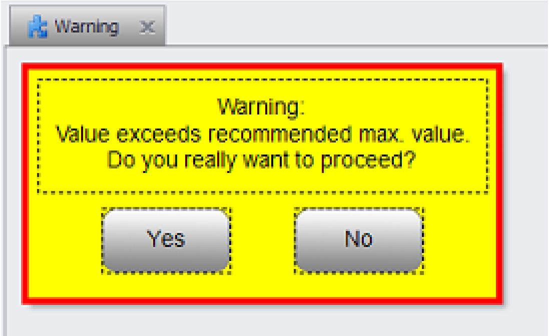
Figure 223: Sequence for confirming an action with the dialog button

2.7.1.2 Conditional confirmation of an action with a script

Use case: Say that your want for an action to be explicitly confirmed depending on defined conditions.

Procedure:

1. Create a user control named **<Warning>**. This user control will be used as a dialog box:

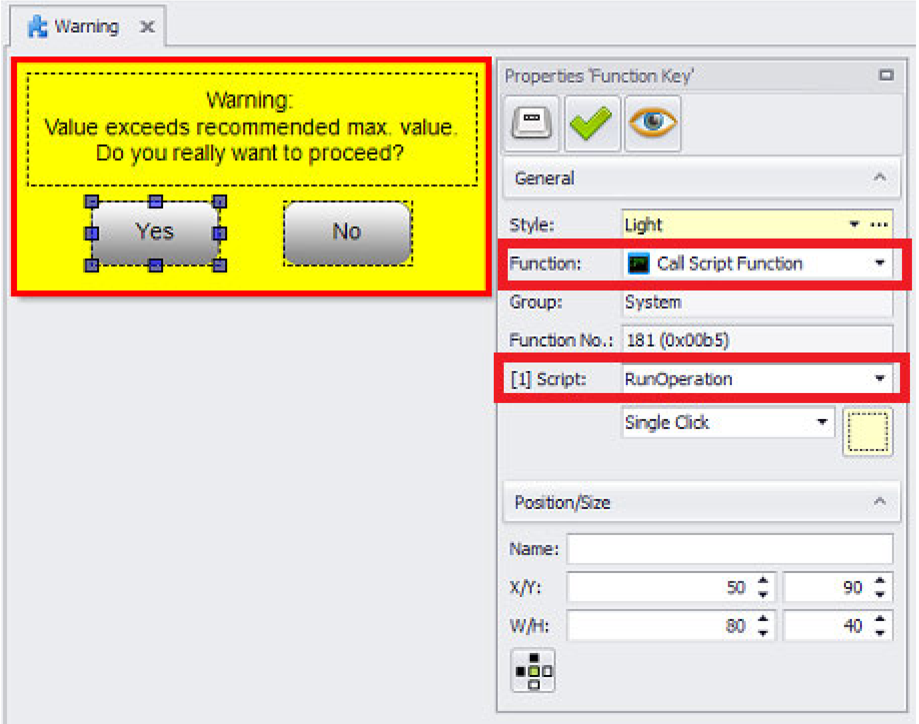


Warning:
Value exceeds recommended max. value.
Do you really want to proceed?

Figure 224: User control named **<Warning>**

The Yes and No buttons are both of the function key type.

2. Configure the Yes function key in such a way that it will run a previously defined script named **<RunOperation>**.
   This script could then, for example, run machine-specific actions:

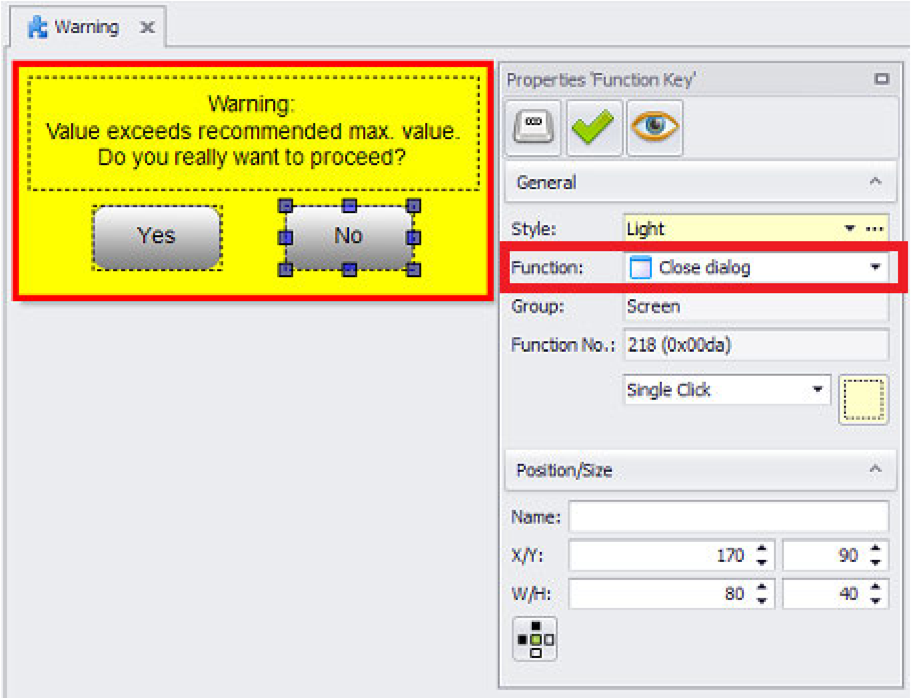


Warning:
Value exceeds recommended max. value.
Do you really want to proceed?

Style: Light
Function: Call Script Function
Group: System
Function No.: 181 (0x00b5)
[1] Script: RunOperation
Single Click
Name:
X/Y: 50 90
W/H: 80 40

Figure 225: Call **<RunOperation>**

3. Configure the No function key in such a way that the Close dialog function will be called:

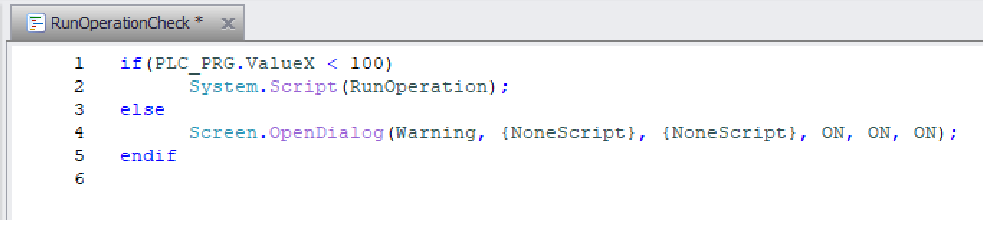


Warning:
Value exceeds recommended max. value.
Do you really want to proceed?

Style: Light
Function: Close dialog
Group: Screen
Function No.: 218 (0x00da)
Single Click
Name:
X/Y: 170 90
W/H: 80 40

Figure 226: Call Close Dialog

4. Create a script called **<RunOperationCheck>** with the following content:

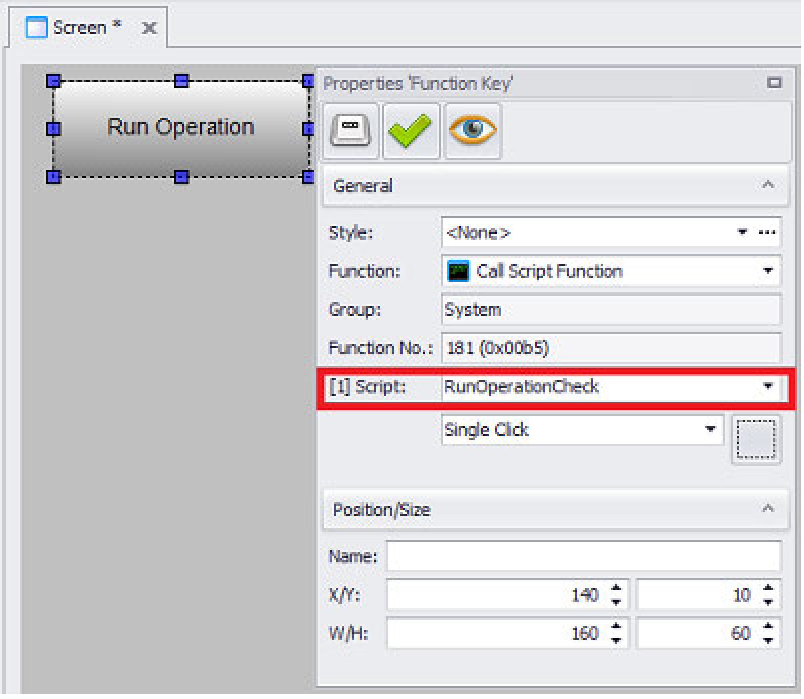


```
if(PLC_PRG.ValueX < 100)
        System.Script(RunOperation);
else
        Screen.OpenDialog(Warning, {NoneScript}, {NoneScript}, ON, ON, ON);
endif
```

Figure 227: Content of **<RunOperationCheck>** script

If the value of the **<PLC_PRG.ValueX>** <tag is less than 100, the **<RunOperation>** script will be run and will accordingly be able to carry out machine-specific actions. If, on the other hand, the value is >= 100, the **<Warning>** dialog box you created will be called.

5. Place a function key object on a standard screen and assign the **<RunOperationCheck>** script you created:



Style: <None>

Function: Call Script Function

Group: System

Function No.: 181 (0x00b5)

[1] Script: RunOperationCheck

Single Click

Name:

X/Y: 140 10

W/H: 160 60

Figure 228: **<RunOperationCheck>** script call

The sequence at runtime will be as follows:

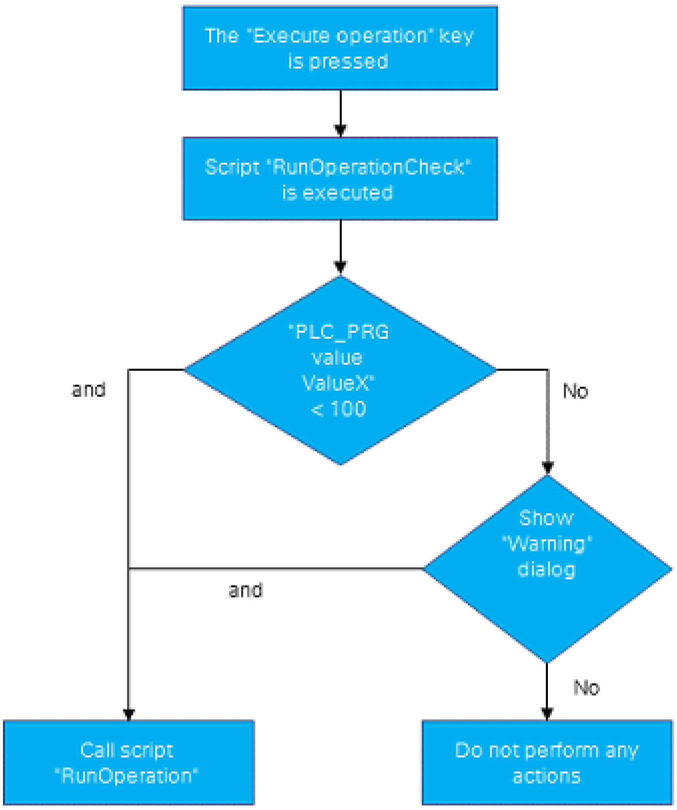

The "Execute operation" key is pressed
Script "RunOperationCheck" is executed
"PLC_PRG value ValueX" < 100
and
No
Show "Warning" dialog
and
No
Call script "RunOperation"
Do not perform any actions

Figure 229: Sequence for confirming an action with a script

2.8 Navigating with the Screen Graph

The Screen Graph dialog box shows how the individual standard screens are laid out in the project, as well as potential navigation paths, in an easily understandable manner.

To access the screen graph, go to *Project Tree\Screens\Standard screens*. Select the screen that should be used as the startup screen. Now open the context menu and select the *Show screen graph* option.

You can also access it by going to Home tab\Project Configuration and selecting it under Information.

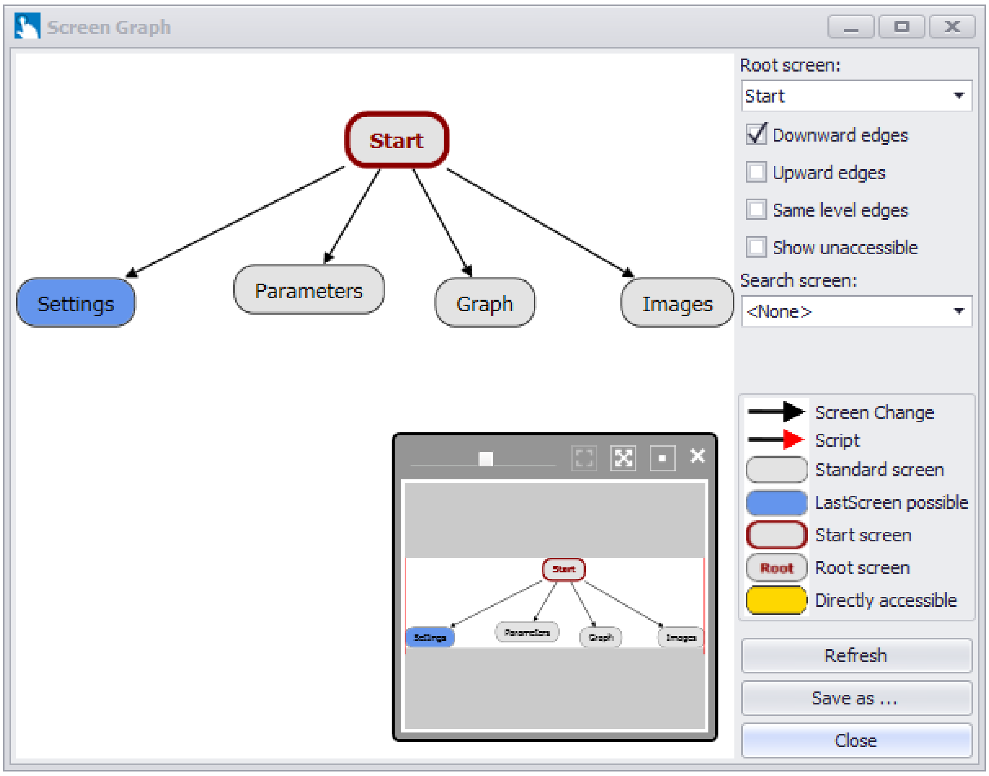
You can use the screen graph to select a start screen from which all other screens can be accessed To enable the various views, simply enable the corresponding checkboxes. Enabling the "Downward edges" option will show all the screens that can be reached from the startup screen by means of a screen change. Meanwhile, enabling the "Upward edges" option will show all the screens from which the startup screen can be reached by means of a screen change.

For additional assistance with screen graphs that have extensive branching, you can use the View Finder.

You can move the screen graph within the pane by dragging and dropping it as necessary. In order to do this, you will first need to place the cursor in an empty location.

You can also move individual objects in the screen graph within the pane by dragging and dropping them as necessary. In order to do this, you will first need to place the cursor on the object you want to move.

Home\Project Configuration\Information\ Show screen graph tab



Root screen: Start

☑ Downward edges

□ Upward edges

□ Same level edges

□ Show unaccessible

Search screen: <None>

→	Screen Change
→	Script
▭	Standard screen
🟦	LastScreen possible
▭	Start screen
Root	Root screen
🟨	Directly accessible

Figure 230: Screen Graph dialog box

View Finder

If the View Finder is not being displayed in the screen graph already, you can open it in the screen graph by clicking on the ✣ arrow.

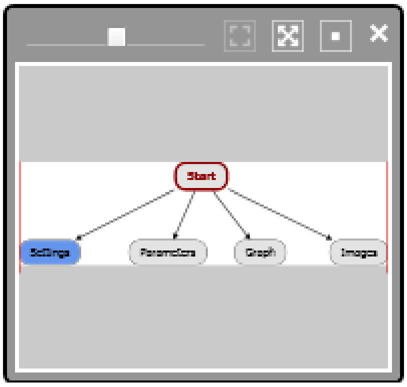


Figure 231: View Finder with Start root screen

The View Finder features the following options:

- Zooming in and out of the screen graph from -2 to +2
- Fitting the screen graph to the pane
- Displaying the screen graph in such a way that the startup screen is at the center of the pane

3. Controls for visualization

In Galileo, controls of various types are available for visualizing data and displaying information.

Basically, the controls can be divided into three categories:
Drawing controls

Drawing controls are primarily used for the graphical design of individual masks. Individual properties of the drawing controls can be influenced via tags (e.g. visibility, line width, line color).

Drawing controls are not operable. Therefore, the configuration of accessibilities is never possible.

Drawing controls include:

- → " Shapes", page 468 (Line, rectangle, Ellipse, Polygon)
- → " Text", page 492
- → " Image", page 439
- → " HTML viewer", page 435
- → " Webcam", page 515
- → " Video Player", page 498

Tag visualization controls

Variable visualization controls are used to visualize and, in some cases, change individual variable values.

Variable visualization controls for the pure display of a variable value include:

- → " Bargraph", page 356
- → " Gauge", page 408
- → " Flag Display", page 379

To the operable variable visualization controls, which make a change of the variable value possible, belong:

- → " List box", page 401
- → " Slider", page 453
- → " Button", page 365
- → " Switch", page 485
- → " Value Display/Entry", page 506

Only one variable visualization object can be operated on the device at a time. It is therefore not possible, for example, to press several objects of the type "key" simultaneously in order to change several tags. If such a functionality is desired, then a realization in a script is suitable for it, which is then executed over a keystroke.

If required, operable variable visualization objects can be made non-operable via the accessibility settings.

Where to link?

Data visualization controls

In addition to objects for visualizing single variable values, there are more complex objects that allow the display of multiple variable values. These so-called data visualization objects include:

- → " Alarm list", page 867
- → " Graph", page 690
- → " Parameter List", page 804
- → " Recipe", page 748

The configuration of the visualization properties takes place at the data visualization object itself. The configuration of the data to be visualized can be done via the "Start" tab in the corresponding category.

3.1 Bargraph

The bargraph visualization object can be used to display a tag of data type BYTE, WORT, DWORT, QWORD or FLOAT as a bar. Moreover, the display range for the bargraph can be defined independently of the corresponding tag's value range. This makes it possible to have the bargraph show small changes in the tag value much more prominently even when a tag has a large value range.

Bargraphs can be subdivided into three subranges. If the value of the tag being represented changes and this means a change of subrange, the bar's color will change accordingly. In order for this to occur, however, different colors need to be defined for the different subranges.

The actual visualization object will be purely an output object unless it is configured with a keyboard. If a keyboard is specified, it will be possible for device operators to enter the value that should be displayed, This keyboard will be displayed on the PC after clicking on the visualization object or on the device after tapping the object.

There are no functions that can be applied to this visualization object.

States cannot be defined for this visualization object.

If this visualization object is used multiple times to display a large number of system components with the same parameter sets, it is recommended to use parameters. Using parameters will optimize your work when it comes to assigning tags to the visualization object.
For detailed information on how to work with parameters, please refer to → section "Using user controls to optimize work", page 279.

Controls *tab*

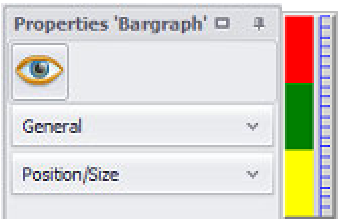


Figure 232: Right: Device screen bar graph

'Bargraph' properties

👁 → section "Visibility", page 259

✓ → section "Accessibility", page 258

→ "General", page 357

→ section "Position/Size", page 242

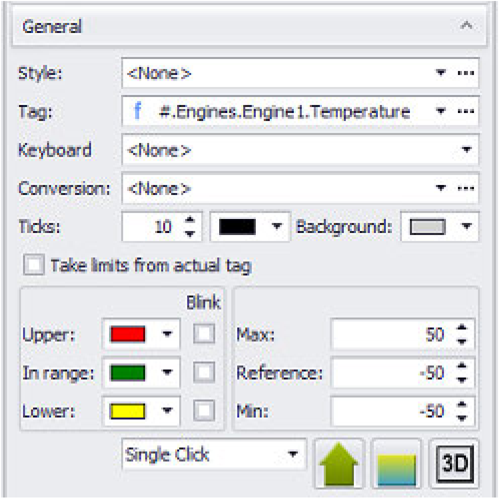
The ✓ accessibility button will only be shown in the Properties pane if a tag and a keyboard have been assigned to the visualization object, as it would not make sense to define accessibility conditions otherwise.

3. Controls for visualization

3.1 Bargraph

3.1.1 General

Controls tab *Bargraph*



General

Style: <None>

Tag: f #.Engines.Engine1.Temperature

Keyboard <None>

Conversion: <None>

Ticks: 10 ■ Background:

□ Take limits from actual tag

		Blink		
Upper:	🟥	□	Max:	50
In range:	🟩	□	Reference:	-50
Lower:	🟨	□	Min:	-50

Single Click

Figure 233: Properties 'Bargraph' pane

Style

Settings that affect the appearance of visualization objects can be bundled and stored as a style. After this, it will be possible to use the style's name to use these settings at any time and apply them to new visualization objects of the same type throughout the entire project. In addition, when changes are made to a style, the changes will automatically be applied to all visualization objects using the style, ensuring that all visualization objects using the same style will have a consistent appearance. For detailed information on how to create a style, please refer to → section "Style management", page 458.

Variable

The visualization object will show the value of the selected tag on the device screen at runtime. Clicking on the drop-down menu arrow will open a list of available tags (BIT, BYTE, WORD, DWORD, QWORD, FLOAT). Please note that in order for the tag you want to be shown in this list, the tag must first be imported or created in the Tags tab.

A parameter can also be specified instead of a tag. In this case, the Tag field will turn into a Type field. In order to be able to work with parameters, the visualization object must be placed on a UserControl. For detailed information on how to work with parameters, please refer to → "Using input parameters", page 279.

Values do not necessarily have to be displayed or entered the way the PLC needs them or the way they are stored in the tags. The Value Conversion Manager can be used to convert tag values and assign a variety of units to them before they are shown by visualization objects. The way value conversion works is described in greater detail in a different section – please refer to → "Conversions", page 921.

Button ...

Limits

Clicking on the ... icon will show the limits for the value tag in a separate dialog box. You can also reach this dialog box via *Project Tree\Tags\<Value tag>\Properties*. When a tag is created, it will already be assigned limits based on the value range defined for it. This value range can be signed or unsigned and will depend on the relevant tag's data type.

The limits can be adjusted as necessary within the value range. In the case of bit tags, you will only be able to specify a default value.

For a description of the tag properties and limits, please refer to: → section "Numeric tag limits", page 563.

Tag Address

This field will only appear if you select a PLC tag, i.e., it will not appear for internal tags.
It will show the PLC address for the selected tag. The properties for PLC tags are described in greater detail in a different section – please refer to → section "Tag address", page 569.

If a single tag is transmitted to more than one communication, the corresponding settings will need to be configured in *Home\Gateway*. For detailed information on how to work with a gateway, please refer to → section "Using the gateway functionality to communicate with multiple PLCs", page 596.

Keyboard

Clicking on the drop-down menu arrow will open a list of available keyboards. If a keyboard is selected, the visualization object will turn into a touch zone at runtime and the person using the device will be able to enter or change a value. If a keyboard is not selected, the visualization object will be a display object without a touch zone.

A numeric system keyboard can be assigned to numeric tags.

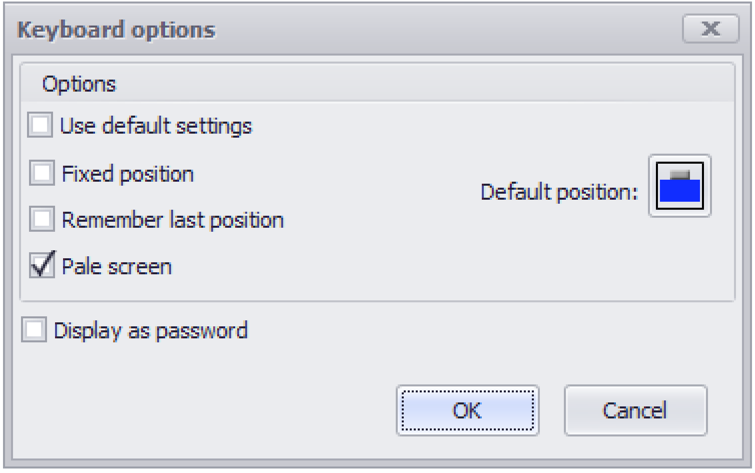
The following are the available numeric system keyboards:

- inc/dec 1
- inc/dec 10
- Numeric
- Numeric Digit Min/Max
- Numeric Min/Max
- Numeric Password
- Select Numeric
- Select Numeric Digit

If you select Select Numeric or Select Numeric Digit as a keyboard, it will be possible to adjust the value with a function from the Keyboards or Characters/Stream characters group, e.g., with the Inc/dec 1 function. For a description of how to use functions, please refer to the → "Controlling functions with the PLC", page 602 section.

To configure the display options for the keyboard, click on the ⚙ icon:

Controls\...\Properties\Keyboard ⚙



Keyboard options

Options

☐ Use default settings

☐ Fixed position

☐ Remember last position

☑ Pale screen

Default position:

☐ Display as password

Figure 234: Keyboard options dialog box

Use default settings: If this option is enabled, the other options will be hidden and the project settings will be applied instead.

Fixed Position: If this option is enabled, an active keyboard cannot be moved around

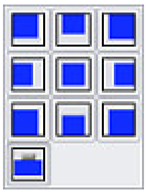
Remember last position: If this option is enabled, the keyboard's position will be saved when closed.
The next time the same keyboard is opened, it will be automatically moved to this saved position.

Gray out screen: If this option is enabled, the view behind the keyboard will be dimmed.

Default position: Used to specify the keyboard's default position relative to the object calling it.
The alignments available for selection include "top left," "center center," "bottom center," etc. plus "docked."
If you select the "docked" option, the Galileo runtime system will decide where the keyboard can best be displayed. If the "Remember last position" option is enabled, this setting will only be applied the very first time the keyboard is opened.

▀ Default position

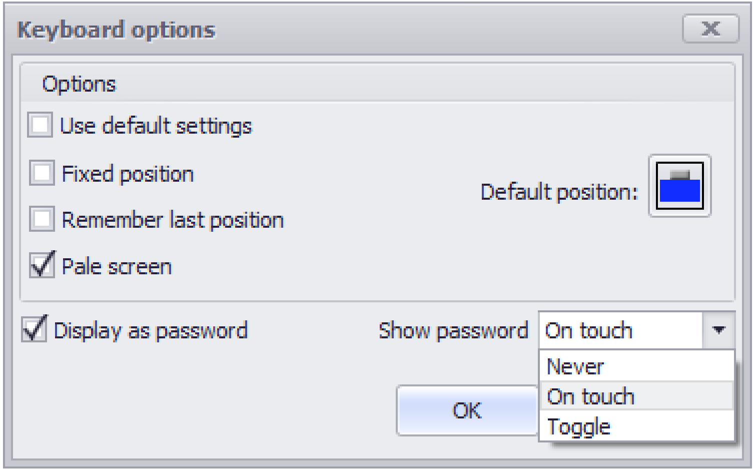
The keyboard will be positioned based on the option selected here. The default position on the screen can be defined by clicking inside the selection box as follows:



Top left ■, top center ■ or top right ■ Center left ■, center center ■ or center right ■ Bottom left ■, bottom center ■ or bottom right ■ or ■ Docked on the button

Display as password: If this option is enabled, only "****" will be displayed instead of the actual value. The following additional options will be available when the option is enabled:

Controls\...\Properties\Keyboard



Keyboard options ✕

Options

□ Use default settings

□ Fixed position

□ Remember last position

☑ Pale screen

Default position:

☑ Display as password

Show password: On touch ▾

- Never
- On touch
- Toggle

OK

Figure 235: Keyboard options dialog box

Never There will be no way to show the entered password in cleartext at runtime

On touch As soon as a number or letter is entered, an 👁 icon will appear in the input field. If this icon is tapped, the password will be shown in cleartext.

As soon as the 👁 icon is no longer being touched, the password will be shown as "****" again.

Toggle As soon as a number or letter is entered, an 👁 icon will appear in the input field. If this icon is tapped, the password will be shown in cleartext.
The password will continue to be shown in cleartext until the icon is tapped again.

→ A previously entered value cannot be shown in cleartext and will remain secret. Only current value input can be shown as cleartext.

For detailed information on how to put together keyboards made up of multiple individual function key visualization objects, please refer to the → section "User Defined Keyboards", page 316 section.

→ In order for this field to be enabled, a tag needs to be selected in the Tag field first.

Conversion

Values do not necessarily have to be displayed or entered the way the PLC needs them or the way they are stored in the tags. The Value Conversion Manager can be used to convert tag values and assign a variety of units to them before they are shown by visualization objects. The way value conversion works is described in greater detail in a different section – please refer to → "Conversions", page 921.

Ticks

The value in the Ticks field is used to specify how the scale should be subdivided.

→ For example, a scale subdivided into 10 increments (Ticks set to <10>) for a 0...100% display will technically have 11 dashes, as the dash for the origin line is not counted.

The color of the scale and its dashes can be set using the drop-down menu on the right.

Background

You can change the color if you want. The following options are available for this purpose:

- Transparent
- Standard Colors
- More Colors...

The most recently used colors will be shown (user-specific).

For detailed information on the available colors, please refer to the following section: → section "Color choice - Object's Color Palette", page 234

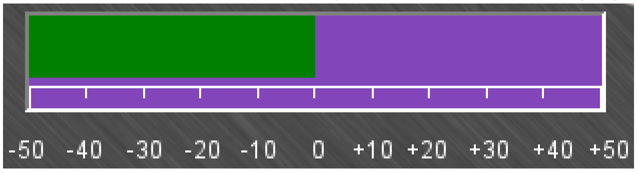


Figure 236: Bargraph with a purple background

Take limits from actual tag

This property will stop being grayed out as soon as a tag is assigned. If the checkbox is enabled, the minimum and maximum for the tag shown in the Tag field will be used as the minimum and maximum limits for the visualization object. This means that the corresponding scaling can be defined freely. For example, a speed indicator can show a scale of up to 300 km/h even if the tag will not reach a value of more than 200 km/h. This also means that cases such as the following are possible: it is possible for a tag of data type BYTE not to have a minimum or maximum, i.e., for the tag to assume unsigned values of 0 to 255 or signed values of -128 to +127. However, say that you want the visualization object to only show the range between 20 and 30 with a fine scale. In this case, you would need to enable the Open Limit checkbox for the tag and set a minimum of 20 and a maximum of 30 for the visualization object.

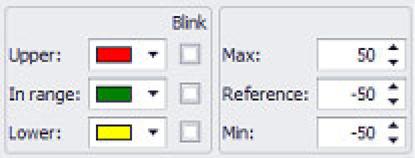
If the checkbox is disabled, the range shown will be defined based on the values in theMin and Max fields, i.e., the range will be a fixed range.

Ranges Color Blinking Limit value

The display range for a bargraph visualization object goes from its lower limit to the upper limit. This means that this display range will be independent from the limits for the tag being represented, except in cases in which the Take limits from actual tag option is enabled. As for the bargraph itself, it will display the current value of the tag being represented by extending from the reference value to the current value.

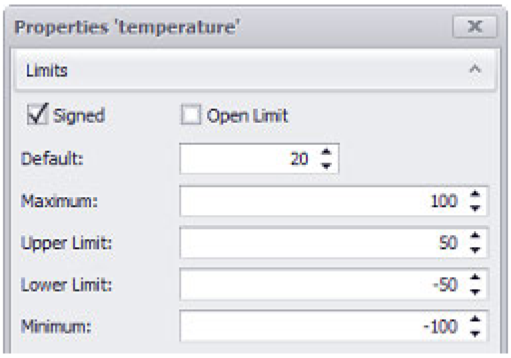
The display range can feature a maximum of three different colors, with each color corresponding to a subrange within which the current value of the tag being represented can fall. This means that up to three subranges can be defined and that a color can be selected for each one of them. These subranges are defined as follows:

- Upper: Subrange between the tag's upper limit and the scale's maximum as defined in the Max field
- In range: Subrange between the tag's upper limit and lower limit
- Lower: Subrange between the tag's lower limit and the scale's minimum as defined in the Min field



		Blink		
Upper:	🟥	☐	Max:	50
In range:	🟩	☐	Reference:	-50
Lower:	🟨	☐	Min:	-50

Figure 237: *Properties 'Bargraph'\General*



Properties 'temperature'

Limits

☑ Signed ☐ Open Limit

Default:	20
Maximum:	100
Upper Limit:	50
Lower Limit:	-50
Minimum:	-100

Figure 238: *Properties for 'Temperature' tag\General*

Flashing

If this checkbox is enabled, the visualization object's display subrange will flash. This flashing effect will be achieved by switching between the display subrange's color and the configured background color.

Example showing a bargraph with configured limits on a device screen

The screenshots below are based on the following settings:

Max = 50; maximum value of the Scale

Reference = -50; this is the value from which the bargraph starts

Min = -50; the scale's minimum value

Lower Limit of Tags = 0

Upper Limit of Tags = 30

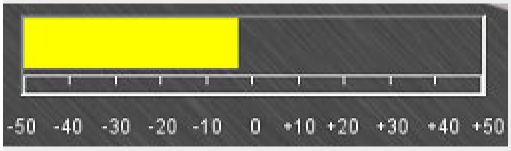


Figure 239: Bargraph at runtime after a value of <-4> is entered

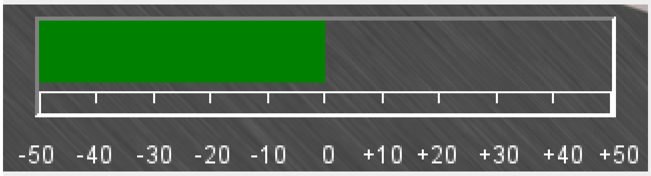


Figure 240: Bargraph at runtime after a value of <0> is entered

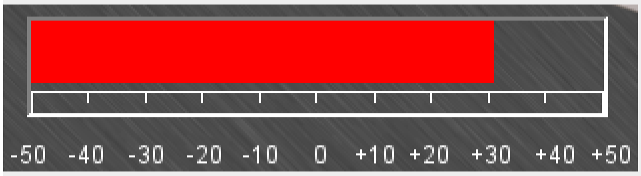


Figure 241: Bargraph at runtime after a value of <+31> is entered

Selection mode on panel Single Click

The following modes can be adjusted for visualization objects with a keyboard:

- Single Click
  Tapping once on a list item or a button on the device screen will activate the selection.
- Delayed
  This option is ideal for implementing a safety function designed to protect against accidental actions. In order for a button on the device screen to be activated, it will first have to be pressed and held (long press) for the configured delay time. By default, this delay time will be 0.5 s, but it can be changed in *Home\Settings\Runtime tab\Touch Settings*.

- Double Click
  Tapping twice on a list item or button on the device screen will select the item / activate the button. By default, the double-click speed will be 0.5 s for a double-click, but it can be changed in*Home\Settings\Runtime tab\Touch Setting*.

Direction

The arrows shown below can be used to define the direction of the bars. This setting will also affect the Scale property, i.e. , if the bars are horizontal, the available scale label options will be Top/Bottom. If they are vertical, the available options will be Left/Right.

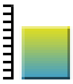
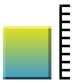
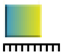
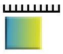
Horizontal Standing

Scale

This property can be used to specify the side of the bar on which a scale should be shown. The available options will be Top/Bottom for horizontal bars and Left/Right for vertical bars.

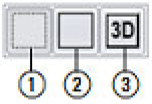
If a scale is being shown, the bar's width will be reduced by about 18 pixels. The reason for this is that these pixels will be used to display the scale.

The scale will be made up of scale dashes and will not include any labels or units. If you want to add a label, you will need to add it using static visualization objects of type "Text".

Without Scale	Scale on the left	Scale on the right
		
	Scale at the below	Scale on top
		

Frame

The visualization object can be displayed with any one of three different designs.



①	no frame
②	with frame
③	3D frame

3.2 Button

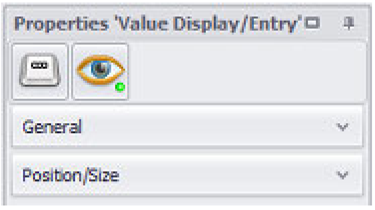
This visualization object can be used to set a single bit in a tag of data type BIT, BYTE, WORD, DWORD, QWORD, FLOAT in the PLC to a value of <0> or <1>. The other bits in the tag will remain unaffected. In real-life applications, buttons are used to switch equipment on and off and are also used as pushbuttons.

By default, this visualization object is an input object with a touch zone. It can also be used as an output object by configuring the appearance of the button's two states differently.

If this visualization object is used multiple times to display a large number of system components with the same parameter sets, it is recommended to use parameters. Using parameters will optimize your work when it comes to assigning tags to the visualization object.
For detailed information on how to work with parameters, please refer to → section "Using user controls to optimize work", page 279.

A detailed example showing how to use this visualization object is provided in section: → section "Your own first project", page 141.

Controls\ tab



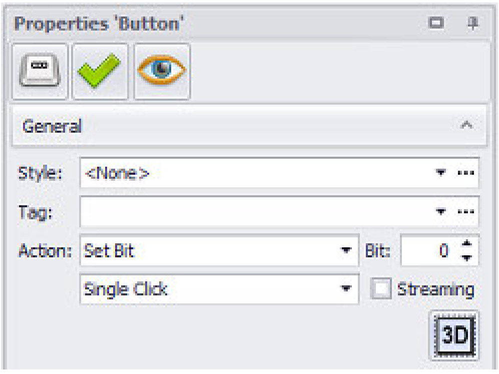
'Button' properties

▭	→ " States", page 371
✓	
👁	→ "General", page 366

The ✓ accessibility button will only be shown in the Properties pane if a tag and a keyboard have been assigned to the visualization object, as it would not make sense to define accessibility conditions otherwise.

3.2.1 General

Controls\ Button



Style: <None>
Tag:
Action: Set Bit
Bit: 0
Single Click
□ Streaming
3D

Figure 242: Window Properties 'Button'

Style

Settings that affect the appearance of visualization objects can be bundled and stored as a style. After this, it will be possible to use the style's name to use these settings at any time and apply them to new visualization objects of the same type throughout the entire project. In addition, when changes are made to a style, the changes will automatically be applied to all visualization objects using the style, ensuring that all visualization objects using the same style will have a consistent appearance. For detailed information on how to create a style, please refer to → section "Style management", page 458.

Tag

Clicking on the drop-down menu arrow will open a list of available tags of data type BIT, BYTE, WORD, DWORD, QWORD, FLOAT. Please note that in order for the tag you want to be shown in this list, the tag must first be imported or created in the Tags tab.

The value of the selected tag will be changed at runtime by the device operator when they tap on the visualization object's button. The settings in the Action and Bit fields will determine whether the selected bit will be set to 1 or 0. If the Streaming checkbox is not enabled, the action will be maintained until the device operator taps on the button again. If the Streaming checkbox is enabled, the action will only be maintained as long as the device operator keeps pressing the button. When the button is released, the tag will assume its original value again.

→ The tag can be set to a predefined value. In order to access the tag's properties, click on the ... .

Using parameters instead of tags is not possible with this visualization object.

Button ...

Limits

Clicking on the ... icon will show the limits for the value tag in a separate dialog box. You can also reach this dialog box via *Project Tree\Tags\<Value tag>\Properties*.

When a tag is created, it will already be assigned limits based on the value range defined for it. This value range can be signed or unsigned and will depend on the relevant tag's data type.

The limits can be adjusted as necessary within the value range. In the case of bit tags, you will only be able to specify a default value.

For a description of the tag properties and limits, please refer to: → section "Numeric tag limits", page 563.

Tag Address

This field will only appear if you select a PLC tag, i.e., it will not appear for internal tags.
It will show the PLC address for the selected tag. The properties for PLC tags are described in greater detail in a different section – please refer to → section "Tag address", page 569.

If a single tag is transmitted to more than one communication, the corresponding settings will need to be configured in *Home\Gateway*. For detailed information on how to work with a gateway, please refer to → section "Using the gateway functionality to communicate with multiple PLCs", page 596.

Action

→ If a device operator presses a button by mistake, they can move their finger sideways while still pressing on the button. This will enable them to "leave" the button without triggering the corresponding action.

As soon as the bit assumes a state of <1>, the visualization object will be shown as configured for the *ON* state in *Extended properties\States* tab.

As soon as the bit assumes a state of <0>, the visualization object will be shown as configured for the *OFF* state in *Extended properties\States tab*.

SET Bit

The bit is set to 1 after pressing when the key is released. If the bit was already set to 1, it remains at 1.

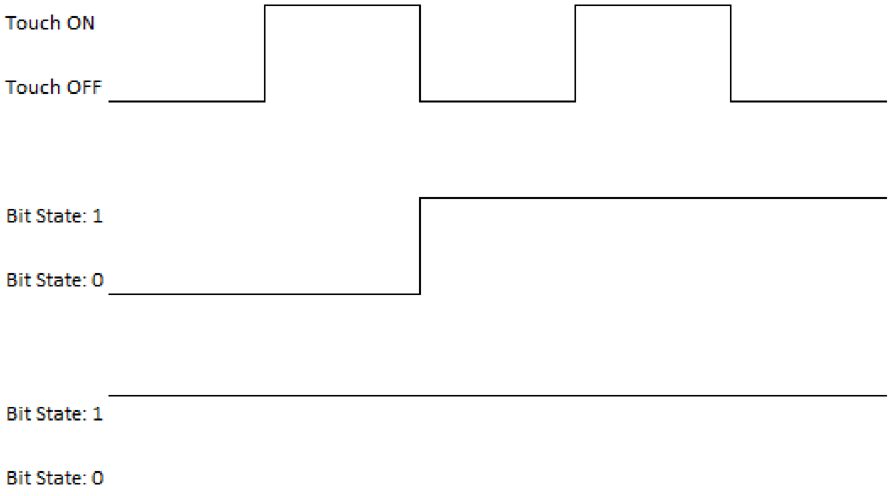


Figure 243: Behavior at "Set bit", depending on the initial value of the respective bit

DELETE Bit

The bit is set to 0 after pressing when the key is released. If the bit was already set to 0, it remains at 0.

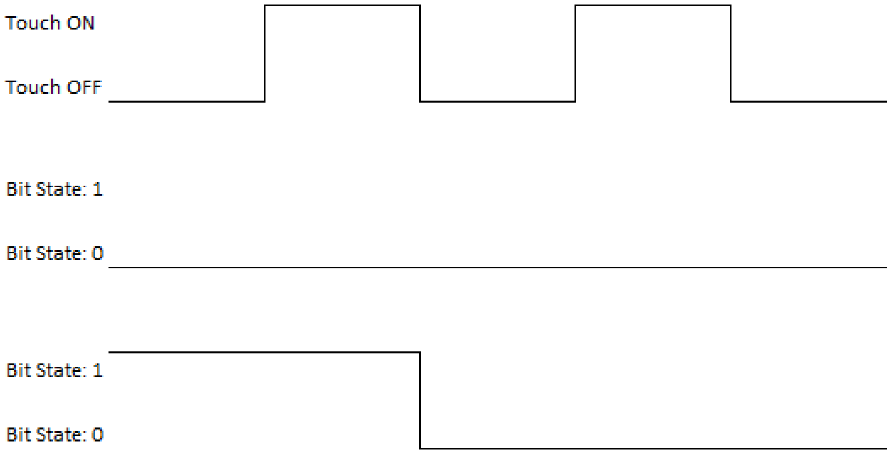


Figure 244: Behavior at "Delete bit", depending on the initial value of the respective bit

Toggle Bit

The bit is inverted after pressing when the key is released.

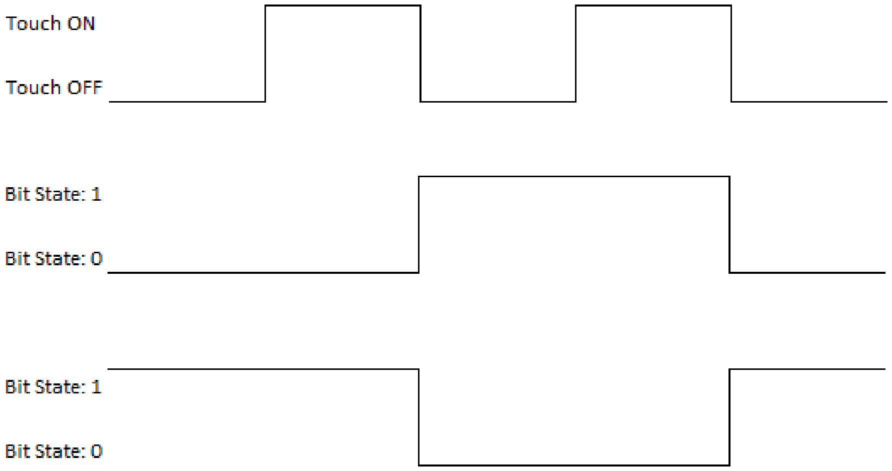


Figure 245: Behavior at "bit toggle", depending on the initial value of the respective bit

Button

Ist dieses Kontrollkästchen durch Häkchen aktiviert, wird das Bit entsprechend der ausgewählten Aktion solange manipuliert, wie die Schaltfläche des Visualisierungsobjektes Taste gedrückt bleibt (Touch EIN). Beim Loslassen wird die Manipulation aufgehoben und das Bit zurückgesetzt.

Die Aktion "Bit umschalten" ist nicht verfügbar, wenn dieses Kontrollkästchen aktiviert ist.

Streaming

If this checkbox is activated by a check mark, the bit is manipulated according to the selected action for as long as the button of the visualization object Button remains pressed (Touch ON). When released, the manipulation is cancelled and the bit is reset.

The "Toggle bit" action is not available if this check box is activated.

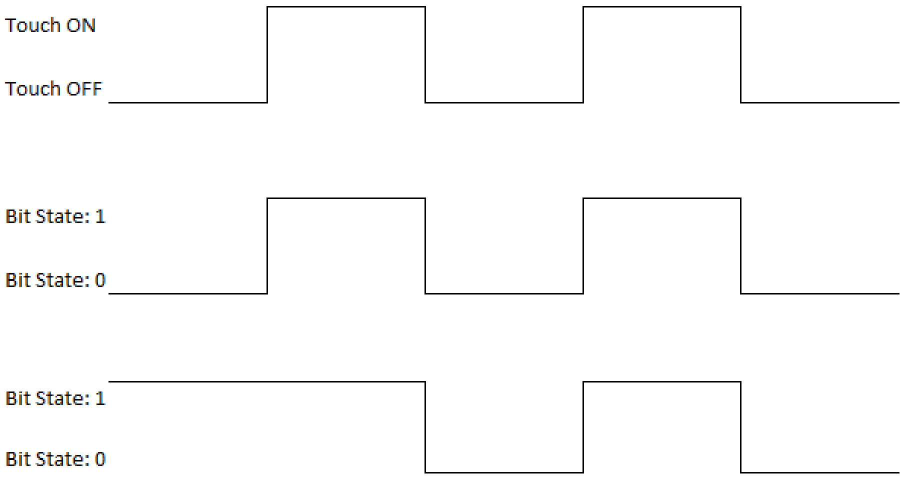


Figure 246: Behavior with "Set bit" in combination with "Pushbutton", depending on the initial value of the respective bit

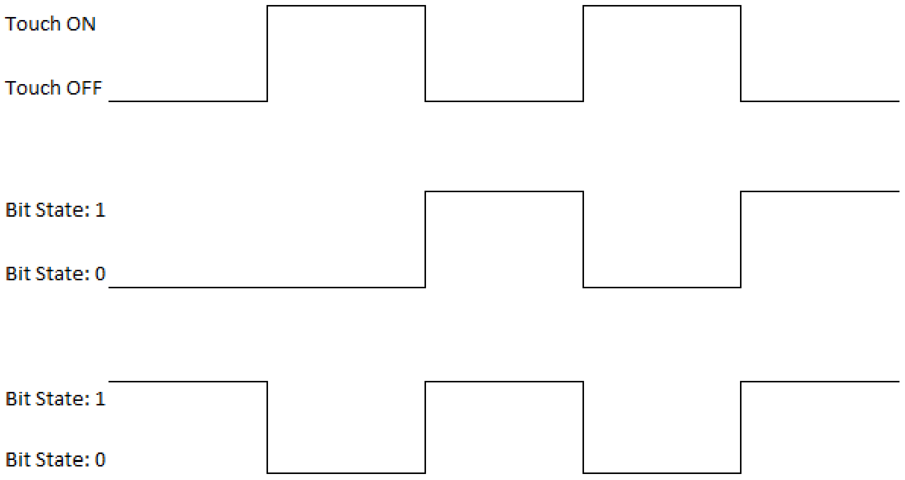


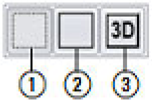
Figure 247: Behavior for "Delete bit" in combination with "Pushbutton", depending on the initial value of the relevant bit

Bit

Used to define the bit in the tag that should be changed. The bit numbers start from 0. The maximum value for the Bit field will be determined by the selected tag, i.e., BYTE = 7, WORD = 15, DWORD = 31. This field will always have a value of 0 in the case of tags of data type BIT.

Frame 3D

The visualization object can be displayed with any one of three different designs.



① no frame

② with frame

③ 3D frame

Selection mode on panel Single Click

The following modes can be adjusted for visualization objects with a keyboard:

- Single Click
  Tapping once on a list item or a button on the device screen will activate the selection.
- Delayed
  This option is ideal for implementing a safety function designed to protect against accidental actions. In order for a button on the device screen to be activated, it will first have to be pressed and held (long press) for the configured delay time. By default, this delay time will be 0.5 s, but it can be changed in *Home\Settings\Runtime tab\Touch Settings*.
- Double Click
  Tapping twice on a list item or button on the device screen will select the item / activate the button. By default, the double-click speed will be 0.5 s for a double-click, but it can be changed in*Home\Settings\Runtime tab\Touch Setting*.

3.2.2 States

Button visualization objects have two states: *Off* and *On*.

The basic configuration parameters for the visualization object's foreground and background are described in detail in the following section: Defining text series

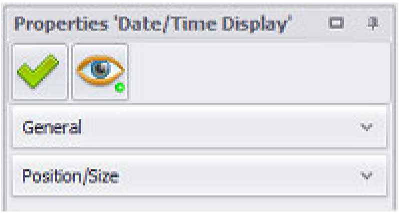
3.3 Date Properties/Time Display

This visualization object can be used to show the date and/or time. In real-life applications, this visualization object is used in a user control that in turn is used as a part of multiple (or all) screens in an application. The object uses the device's integrated clock, which can be synchronized with the PLC. For a description of how to synchronize them, please refer to the → "Synchronizing the time and date", page 612.

This visualization object is purely an output object.

There are several functions that can be applied to this visualization object in order to configure the date and time. Two functions that are particularly worth mentioning within this context is the function for switching to and from daylight savings time and the US date and time On/Off function, which is used to switch between a 24-hour format and a 12-hour US format. When using the US time format, the day and months will be reversed as well, and am/pm will be additionally shown. The functions for date/time display objects are described in detail in the internal functions group – please refer to → "Internal functions (Internals)", page 1109.

Controls\ tab



Properties 'Date/Time Display'

General

Position/Size

Properties of the 'Date/Time Display'

✓ → section "Visibility", page 259Visibility

👁 → section "Visibility", page 259Visibility

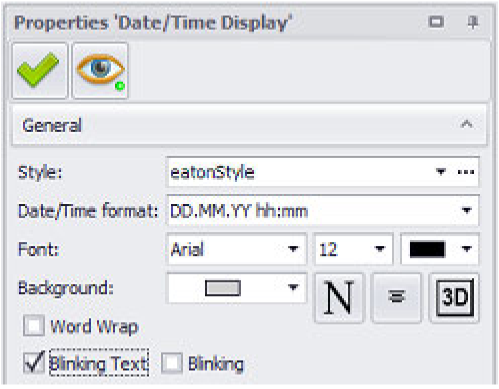
→ section "General", page 373

→ section "Position/Size", page 242

The ✓ accessibility button will only be shown in the Properties pane if a tag and a keyboard have been assigned to the visualization object, as it would not make sense to define accessibility conditions otherwise.

3.3.1 General

Controls\ Date/Time Display



Properties 'Date/Time Display'

General

Style: eatonStyle

Date/Time format: DD.MM.YY hh:mm

Font: Arial 12

Background:

☐ Word Wrap

☑ Blinking Text ☐ Blinking

Figure 248: Window Properties 'Date/Time Display'

Style

Settings that affect the appearance of visualization objects can be bundled and stored as a style. After this, it will be possible to use the style's name to use these settings at any time and apply them to new visualization objects of the same type throughout the entire project. In addition, when changes are made to a style, the changes will automatically be applied to all visualization objects using the style, ensuring that all visualization objects using the same style will have a consistent appearance. For detailed information on how to create a style, please refer to → section "Style management", page 458.

Date/Time format

This display format field is used to define how the time and/or date will be shown.

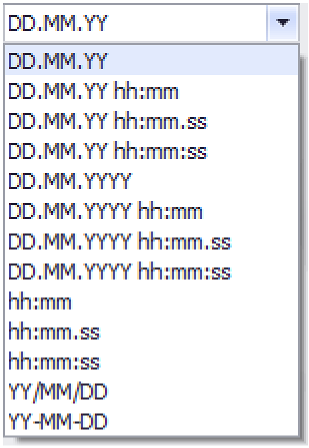


Figure 249: Date/Time format drop-down menu

DD	Day (1...31)
MM	Month (1...12)

YY	Year two digit (e.g. 15)
YYYY	Year four digit (e.g. 2015)
hh	Hour (0...23)
mm	Minute (00...59)
ss	Second (0...59)

For a description of the font, orientation, and background color, please refer to section:.

3.4 Dialog button

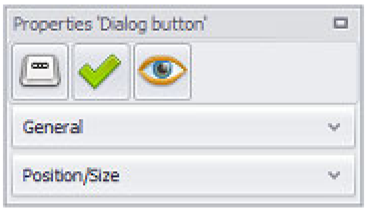
This visualization object can be used to add a specific dialog box to the application.

This dialog box must first be configured as a user control.

The Dialog screen field in the Properties pane for the dialog button can be used to define which user control will be shown. You can also use these properties to set the corresponding orientation and dialog box behavior.

By default, this visualization object is an input object with a touch zone.

Controls\ tab



'Dialog button' properties

→ "States", page 377

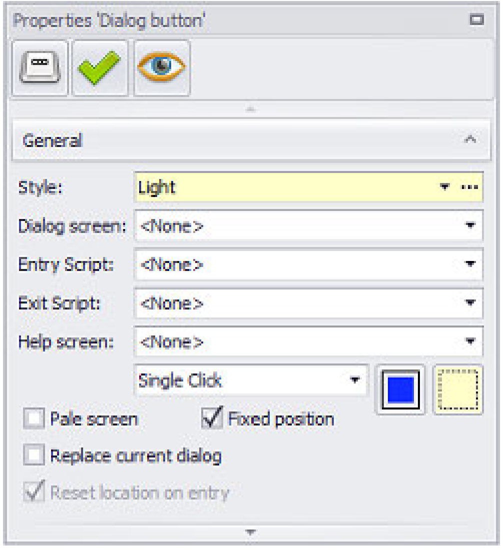
✓

Visibility

→ "General", page 375

3.4.1 General

Controls\ Dialog button tab


Properties 'Dialog button'

General

Style: Light

Dialog screen: <None>

Entry Script: <None>

Exit Script: <None>

Help screen: <None>

Single Click

□ Pale screen ☑ Fixed position

□ Replace current dialog

☑ Reset location on entry

Figure 250: Properties 'Dialog Button' pane

Style: Settings that affect the appearance of visualization objects can be bundled and stored as a style. After this, it will be possible to use the style's name to use these settings at any time and apply them to new visualization objects of the same type throughout the entire project.

In addition, when changes are made to a style, the changes will automatically be applied to all visualization objects using the style, ensuring that all visualization objects using the same style will have a consistent appearance. For detailed information on how to create a style, please refer to → "Style management", page 458.

Dialog screen: Used to select a user control as the dialog box that should be shown when the dialog button is clicked or tapped on.

Entry Script: This function is used to run a script when the screen is opened (before the screen is displayed).
The script specified in the Entry Script field will be run before the screen is opened.

Exit Script: This function is used to run a script when the screen is exited.
The script specified in the Exit Script field will be run before the screen is closed.

Help screen: Used to assign a previously created help screen

For a detailed description of event scripts, please refer to the→ section " Scripts", page 952 section

☑ When the following options are enabled:

Gray out screen: The standard screen underneath the dialog box will be shown semi-transparent.
As soon as a dialog box is opened, the standard screen surrounding the dialog box will be disabled.

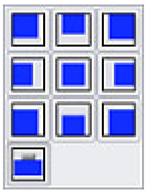
Fixed Position: It will not be possible to move the dialog box
.

Replace current dialog: When another dialog box is opened from an open dialog box, the first dialog box will be closed if this option is enabled.

Reset location on entry: If the dialog box has been moved, it will be moved back to its default position the next time it is called.

🟦 Default Position

The dialog box will be positioned based on the option selected here. The default position on the screen can be defined by clicking inside the selection box as follows:



Top left ▣, top center ▣, or top right ▣

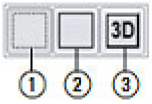
Center left ▣, center center ▣, or center right ▣

Bottom left ▣, bottom center ▣, or bottom right ▣

or ▣ Docked on the button

Frame 3D

The visualization object can be displayed with any one of three different designs.



① no frame

② with frame

③ 3D frame

3.4.2 States

Dialog button visualization objects have two states: *Inactive* and *Active*.

The basic configuration parameters for the visualization object's foreground and background are described in detail in the following section: Defining text series

3.4.3 Dialog button functions

Can be called with a function key:

→ section "CloseDialog", page 378
→ section "CloseAllDialogs", page 377

Can be called as functions in scripts:

→ section "OpenDialog", page 378
→ section "CloseDialog", page 378
→ section "CloseAllDialogs", page 377

→ section "Dialog boxes for interactive communication", page 344
→ section "'Dialog button' properties", page 375

3.4.3.1 CloseAllDialogs

This function closes all open dialog boxes in all instances.

Number of the Function	219 (DBh)
Executable from the PLC	Yes
Function with ON/OFF state	No
Function available in standard screens	Yes
Function available with keyboards	No

Function available in script	Yes
Function available in application entry script	Yes

3.4.3.2 CloseDialog

This function closes the frontmost dialog box.

Number of the Function	218 (DAh)
Executable from the PLC	Yes
Function with ON/OFF state	No
Function available in standard screens	Yes
Function available with keyboards	No
Function available in script	Yes
Function available in application entry script	Yes

3.4.3.3 OpenDialog

This script function corresponds to the → " Dialog button", page 375 object.

3.5 Flag Display

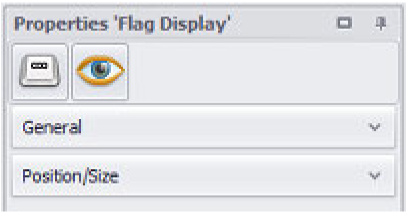
In real-life applications, this visualization object is used to signal the PLC's response to a device action. While it is true, for example, that a button's ON state can be configured in such a way that the state will be clearly displayed, this is not quite the same thing, as the ON state does not depend on the PLC's response, but rather on whether the button is pressed. In other words, this does not guarantee that the PLC has implemented the triggered action successfully.

This visualization object can be used to display an image, color, or text on the device either for each bit in a tag or for each value in a tag. If several bits have a value of "1", the least significant bit will be used to determine the visualization object's state.

This visualization object is purely an output object.

If this visualization object is used multiple times to display a large number of system components with the same parameter sets, it is recommended to use parameters. Using parameters will optimize your work when it comes to assigning tags to the visualization object.
For detailed information on how to work with parameters, please refer to → section "Using user controls to optimize work", page 279.

Controls\ tab



Properties 'Status display'

⌨ → "States", page 382

✓

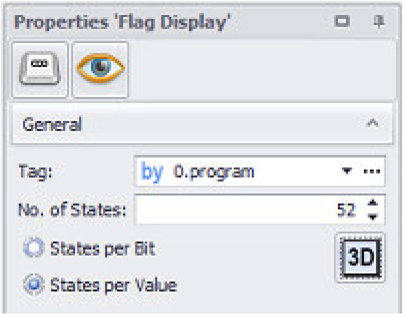
👁

→ "General", page 380

The ✓ accessibility button will only be shown in the Properties pane if a tag and a keyboard have been assigned to the visualization object, as it would not make sense to define accessibility conditions otherwise.

3.5.1 General

Controls\ Flag Display



Properties 'Flag Display'

General

Tag: by 0.program

No. of States: 52

○ States per Bit

● States per Value

Figure 251: Properties 'Flag Display' pane with the States per Value option selected

Tag

The value of the selected tag at runtime will determine what the visualization object will look like on the device screen. The States tab can be used to define a specific appearance for each possible visualization object value.

Clicking on the drop-down menu arrow will open a list of available tags (BIT, BYTE, WORD, DWORD, QWORD, ERROR). Please note that in order for the tag you want to be shown in this list, the tag must first be imported or created in the Tags tab.

Button ...

Limits

Clicking on the ... icon will show the limits for the value tag in a separate dialog box. You can also reach this dialog box via *Project Tree\Tags\<Value tag>\Properties*. When a tag is created, it will already be assigned limits based on the value range defined for it. This value range can be signed or unsigned and will depend on the relevant tag's data type.

The limits can be adjusted as necessary within the value range. In the case of bit tags, you will only be able to specify a default value.

For a description of the tag properties and limits, please refer to: → section "Numeric tag limits", page 563.

Tag Address

This field will only appear if you select a PLC tag, i.e., it will not appear for internal tags.
It will show the PLC address for the selected tag. The properties for PLC tags are described in greater detail in a different section – please refer to → section "Tag address", page 569.

If a single tag is transmitted to more than one communication, the corresponding settings will need to be configured in *Home\Gateway*. For detailed information on how to work with a gateway, please refer to → section "Using the gateway functionality to communicate with multiple PLCs", page 596.

No. of States, per Bit, per Value

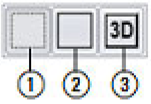
This input field can be used to set the number of states that will be used. The maximum number will depend on the value range of the tag being used. In addition, this number will depend on whether the States per Bit or States per Value option is selected.

→ For performance reasons it is recommended not to use more than 10.000 states.

- States per Bit
  When this option is selected, every bit in the tag will represent a different state. The number of states will start with <1>. By default, this means that no bit is set. Counting is done in ascending order with a fixed bit sequence starting with bit 0. Accordingly, a number of states of <2> will mean that bit 1 can be set or that no bit is set. The maximum number of states will be the number of bits plus the state in which no bit has been set. In other words, the maximum number of states for one bit is 2, with other maximum numbers as follows: one byte = 9, one word = 17, one double word = 33.
  If more than one bit is set within the tag, the least significant bit will be the one that determines the state.
- States per Value
  If this option is selected, each tag value will represent a different state.
  The first default state will be defined with a value of <0> for the tag. Additional states will be counted in continuous ascending order. By default, the last state must be <invalid state>. For a total of <4> states, this will mean that it will be possible to define the states for tag values 0, 1, 2 and <invalid state>.
  For a bit, the maximum number of states is 3, with other maximum numbers of states as follows: one byte = 257, one word / one double word = 65,537. While it is true that the data range for a double word is significantly larger at 4,294,967,297, it is not possible to take full advantage of it due to the limit on the number of colors available.

Frame 3D

The visualization object can be displayed with any one of three different designs.



1. no frame
2. with frame
3. 3D frame

3.5.2 States

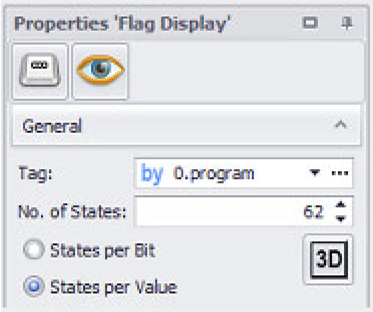
The number of possible states for a flag display is defined in the No. of States field in the Properties 'Flag Display' pane.

The basic configuration parameters for the visualization object's foreground and background are described in detail in the following section: Defining text series

Example for a flag display object that uses the States per Value property

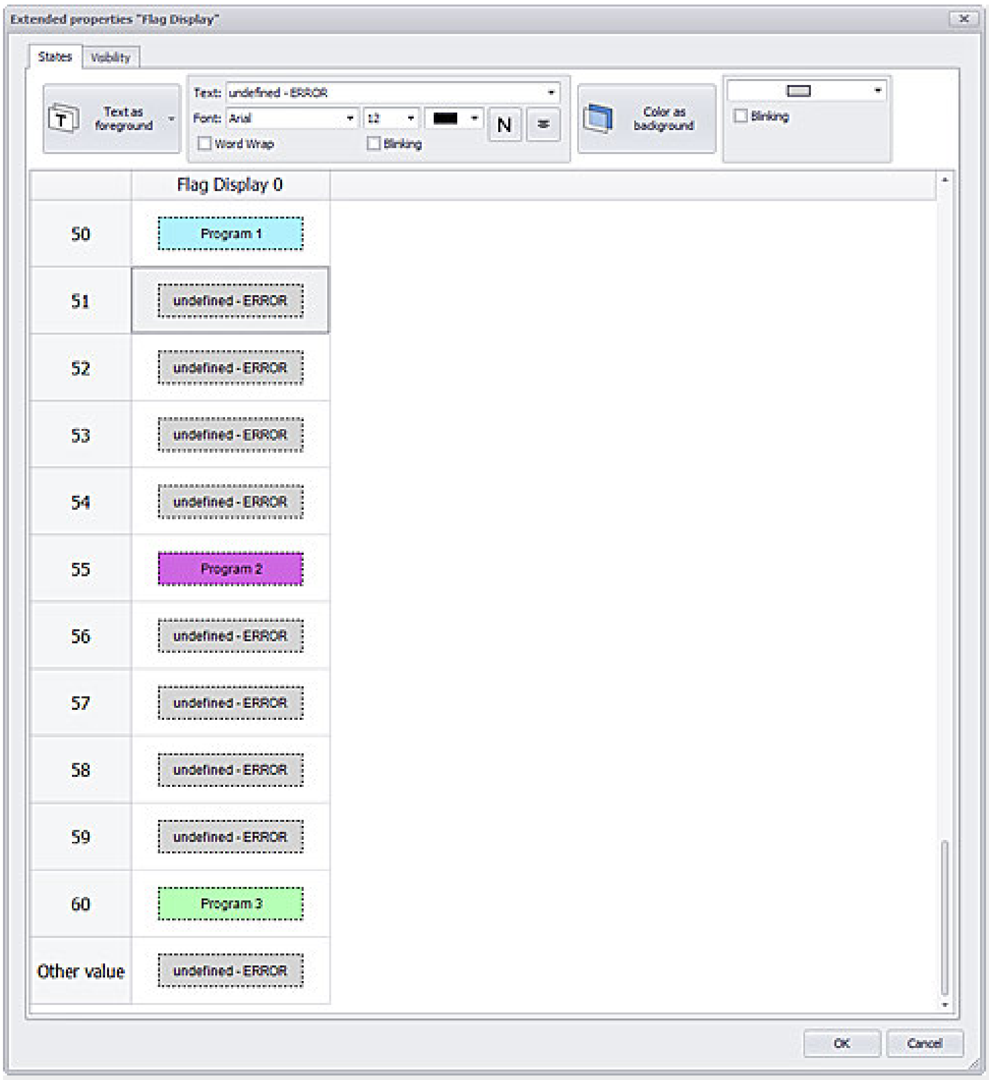
Say you have defined three programs for your machine controller. The value in the program tag determines which program will be selected. If program = 50, Program1 will be selected; if program = 55, Program2 will be selected; if program = 60, Program3 will be selected. The flag display object will show which program has been selected using different colors and labels.

The exercise is implemented with a "States per Value" flag display object in this case. A program tag of data type BYTE is created. The umber of states for the flag display is set at 62.



Tag: by 0.program
No. of States: 62
○ States per Bit
● States per Value

Figure 252: Window Properties 'Status display'



Extended properties "Flag Display"

Text: undefined - ERROR
Font: Arial 12
□ Word Wrap □ Blinking
□ Blinking

	Flag Display 0
50	Program 1
51	undefined - ERROR
52	undefined - ERROR
53	undefined - ERROR
54	undefined - ERROR
55	Program 2
56	undefined - ERROR
57	undefined - ERROR
58	undefined - ERROR
59	undefined - ERROR
60	Program 3
Other value	undefined - ERROR

Figure 253: States for a flag display object with the States per Value property and a No. of States: <62>

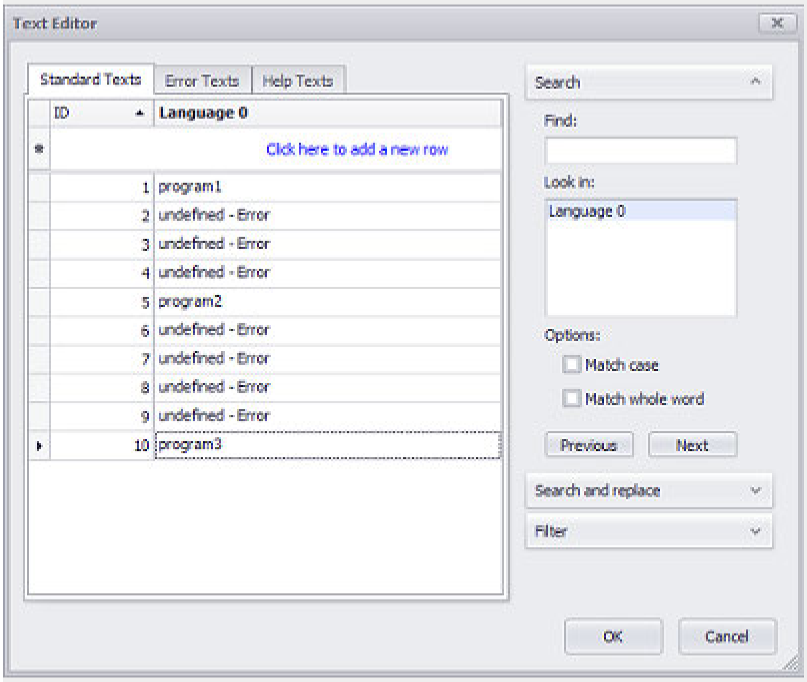
Example: Define series

Prerequisites

- A flag display with 62 possible states must have already been created as in the previous example.
- You must have already created 11 standard texts in the Text Editor.

You will now be using the Define series option in the context menu in order to assign the texts from the Text Editor to states 50 through 60.

Home\Texts\Texts tab



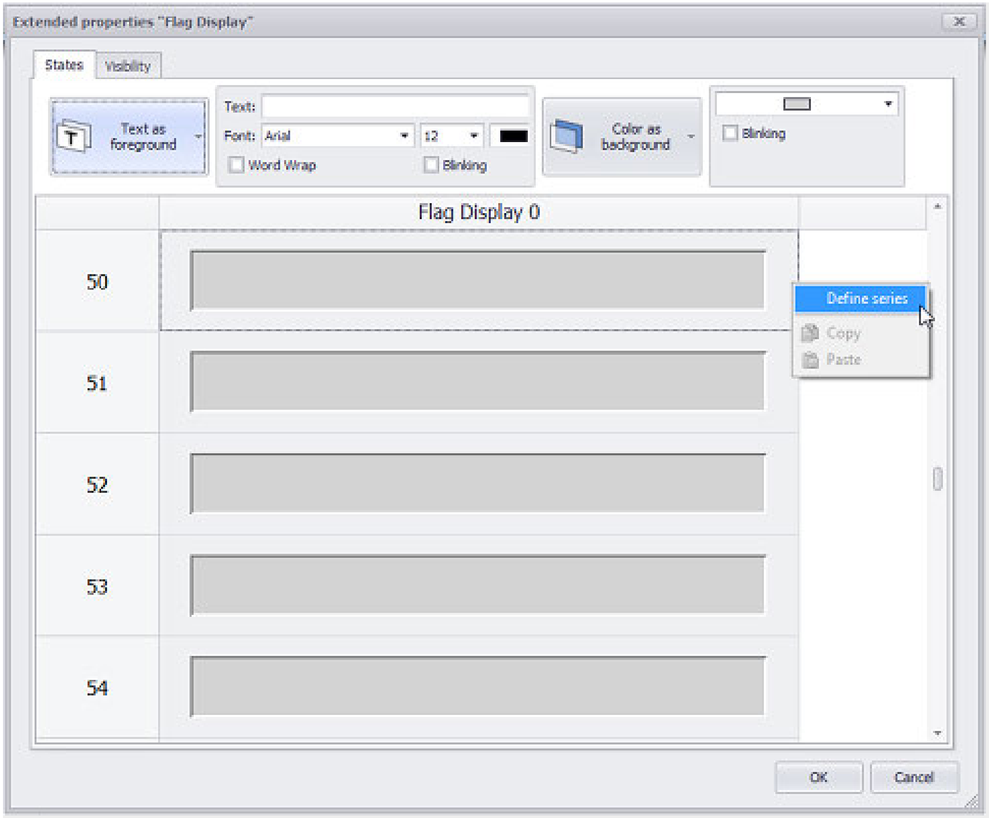
ID	Language 0
	Click here to add a new row
1	program1
2	undefined - Error
3	undefined - Error
4	undefined - Error
5	program2
6	undefined - Error
7	undefined - Error
8	undefined - Error
9	undefined - Error
10	program3

Figure 254: Text list: Window 'Texts'

To do so, follow the steps below:

- Open the extended properties of the Status display by clicking .
- Select <Text as a foreground> .
- Select the states to which you want to assign the texts from the Text Editor, e.g., states 50 through 60. To do this, use the usual Windows method for selecting multiple items.
- Right-click to open the context menu.
- Select <Define series>.
- For the Start value, select the ID from the Text Editor from which the series should start, e.g., <0>.
- For the Increment by field, select the value by which the Text Editor ID should be incremented for every additional entry in the series, e.g., <1>.

The extended properties for the flag display will look as shown in the previous example.



Extended properties "Flag Display"

Text:

Font: Arial 12

□ Word Wrap □ Blinking

□ Blinking

Figure 255: States 50 through 60 are selected and the context menu is open

3.6 Function keys

Function key visualization objects are a key element in working with GALILEO. They are used to call functions.

In real-life applications, function key visualization objects are used for two reasons:

- Say you want to enable device operators to interact with and modify objects, visualization objects, and/or device settings on the device screen at runtime.
- Say you want to enable PLCs to interact with and modify objects and/or device settings at runtime.

Function key visualization objects are usually shown in the form of a button on the device screen. When a device operator taps on the button on the device screen, the function linked to it is triggered. The action performed by each function is pre-defined, as is the type of object or visualization object on which the action will be performed. If there are multiple visualization objects of a single type on a screen or multiple objects of a single type in a project, it will be necessary to enter parameters using the properties for the relevant function keys so as to uniquely specify the object to which each function key applies. Parameters may also be required in order to specify the action that the function being triggered should carry out with the required detail. These parameters also need to be entered using the properties for the relevant function keys.

Function key visualization objects can be visually configured as a button resembling a physical button or a as a button containing an icon or simply text. The *Controls\Function Key\Properties\States tab* contains a variety of configuration options for a function key's appearance.

Function key visualization objects are used to link an event to a function and to parameters. This event can be triggered by a device operator by tapping on the corresponding button on the device screen or by the PLC by setting a specific bit in the Control system tag.

→ Not all functions can be triggered the same way with a function key visualization object and with a PLC.

For more information on how a function can be triggered, please refer to the corresponding function's description. Functions are grouped based on the objects and visualization objects to which they apply, as well as the topics they fall under. For example, the Internals group contains all the functions related to the device, e.g., adjusting the device's backlight, etc.

Function key visualization objects are purely output objects.

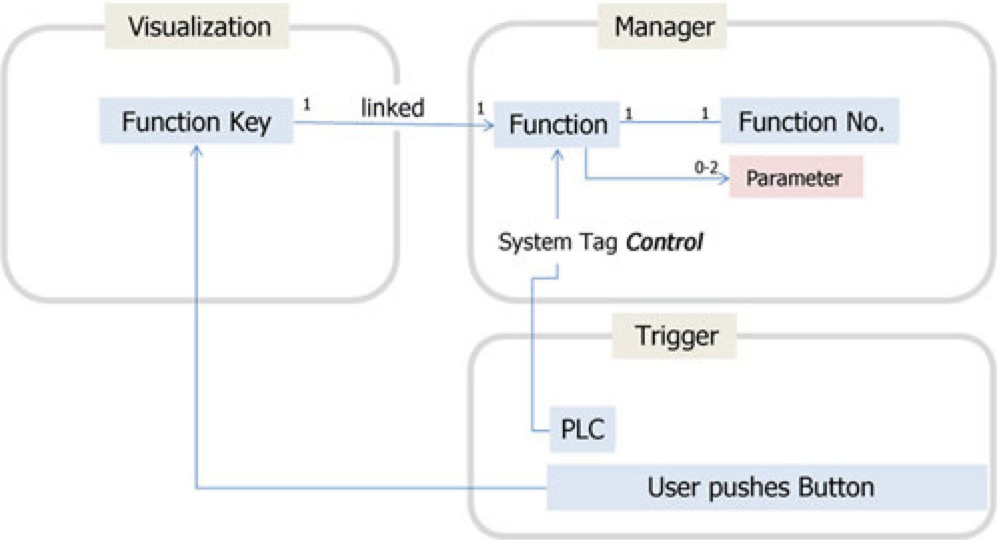


Figure 256: How function keys work: how the function key visualization object and functions interact with each other

3.6.1 Function key and functions

Since the purpose of function key visualization objects is to call objects, this section will be used to explain what functions are, how they are grouped, and where the descriptions for the various individual functions can be found.

The purpose of functions is to make it possible to modify and manage objects, visualization objects, and device settings at runtime.
For example, GALILEO has a function that can be used to reduce the backlight current and, as a result, the device's brightness to 40%. This function is called Brightness 40%,

An event is needed in order to trigger it.

- Say you want to enable device operators to interact with and modify objects, visualization objects, and/or device settings on the device screen at runtime. In this case, the event will be when the function key visualization object's button is tapped on.
- Say you want to enable PLCs to interact with and modify objects and/or device settings at runtime. In this case, the event will be when a specific bit in the Control system tag is set.
- Event scripts can also be used to call functions.

Certain functions can only be called with scripts, while others can only be called with a function key visualization object. In addition, certain functions can be called either way. For more information, please refer to the description for the relevant function.

If you want to trigger functions with the PLC at runtime, make sure to familiarize yourself with the topic of data communications. For a description of what happens when data is being communicated, please refer to the → section "Data communication principles", page 580 and → section "Controlling functions with the PLC", page 602 sections.

Difference between function key and function

A function is not the same as a function key. A function key is a visualization object that is displayed on the device screen. Thesefunction keys are used to trigger events. Meanwhile, a function is responsible for carrying out a specific action. This action will be carried out when the corresponding event is triggered.

If, for example, you wanted to enable device operators to set the brightness on the device screen to 40%, you would put a function key visualization object on a screen and then link this object to the Brightness 40% function.

For a description of how to use the function key visualization object, please refer to the → section "Using function keys to trigger actions", page 389 section.

How functions are grouped and where the corresponding descriptions are found

All the functions available in GALILEO are organized by groups in order to make them more manageable. These groups, in turn, are based on the objects and visualization objects to which the functions they contain apply. This documentation covers the various functions under the specific topic they fall under. The table below contains all the function groups, as well as the topics under which you can find the descriptions for the corresponding functions.

Alarm	Alarm management functions
Characters	Entering characters
Communication	Communication Functions
Conversion	Conversion Functions
Display	Functions relevant to the device display
Event	Event query for an IF condition in a loop script
Graph Block	Graph Block functionsGraph Block function overview
Keyboards	Keyboard call
Language	Functions for changing the language
ListBox	Functions for navigating through a list
Math	Advanced mathematical operations
Parameter List	Functions for navigating through a parameter list
Source file, print resolution (EPS, TIF)	Printer Functions
Recipe	Various recipe functions
Screen	Functions relevant to the screen
StreamCharacters	Entering characters
String	Text functions
Entry	UserControl
System	Runtime Functions
SystemTime	Functions relevant to the device's or PLC's time
UserLogger	User Logger Functions
UserManagement	User management functions
→	You can also find all functions in the online help, grouped in the Topics tab. The Topics tab is found un the upper left corner, next to the Contents tab.

How to read the Description of functions

Each function is described in its own section. Accordingly, the name for each function is used as the title for its section, followed by a description of the corresponding functionality. This description is then followed by information on possible limitations (e.g., whether a function can be triggered by a PLC) in table format. .

→ Not all functions can be triggered the same way with a function key visualization object and with a PLC.

Number of the Function	Which function number does the PLC need to write to the Control system tag in order to trigger the function?
Executable from the PLC	Is it allowed for the function to be triggered by the PLC?
Function with ON/OFF state	Does an ON/OFF parameter need to be passed to the function?
Function available in Standard screens	Is it allowed to trigger the function with a function screen visualization object?
Function available in keyboards	Is it allowed to trigger the function with a function screen visualization object if the visualization object is on a keyboard screen?
Function available in script	Is a function call by a script allowed?
Function available in application entry script	Can the script function be called in the application entry script as well?

Example showing how to read the description

Function → section "Brightness 40%", page 1111

3.6.2 Using function keys to trigger actions

In real-life applications, function keys are used to provide device operators with a way to interact with and modify the objects on the device screen at runtime. If an object needs to be changed, this will require an action on behalf of the device operator, who must accordingly tap on the relevant function key visualization object. Function key visualization objects can be visually configured as a button resembling a physical button or a as a button containing an icon or simply text. When a device operator taps on a function key visualization object, the function associated with that object will be triggered. The association between a function key and the function that should be triggered can be defined in the properties for the function key. Please note that certain functions require parameters in order to be run. These parameters also need to be entered using the properties for the relevant function keys. In addition to this, it is important to point out that the functions for many function keys can also be triggered by a PLC.

The following steps show how to create a function key visualization object and link it to a function. The use case that follows describes a typical scenario, as well as the results expected on the device screen or computer.

Prerequisites

- A device must already be selected, e.g., XV-102-D6-70TWRC-10
- A communication system must already be selected, e.g., XSOFT-CODESYS-2
- A screen must have already been created, e.g., **Start screen**.
- You must have already imported the tags you want to display and edit from the PLC.

To create a function key, follow the steps below:

1. **Creating a function key visualization object and visually configuring the corresponding button**

▶ Click on *Controls\Function Key*.

▶ Position the object
Move the cursor to the screen and draw the area for the visualization object.

▶ To have the function key look like an actual physical button, click on ⬚ and then select the 3D style.

▶ Click on *Controls\Function Key*, click on the button that says Text as foreground, and select the Use an icon as foreground option.

▶ In the drop-down menu to the right, select the image you want, e.g., <Playbutton>. Make sure that the image is found in *Project Tree\Images tab\Project images folder*.

→ Make sure to click on the actual arrow for the Project images folder. If you click anywhere else, the folder will not expand.

▶ Select a blue color as the background color.

▶ Click on OK.

The changes you make to the visual properties for the function key visualization object will be shown directly on the screen.

Configure a function key

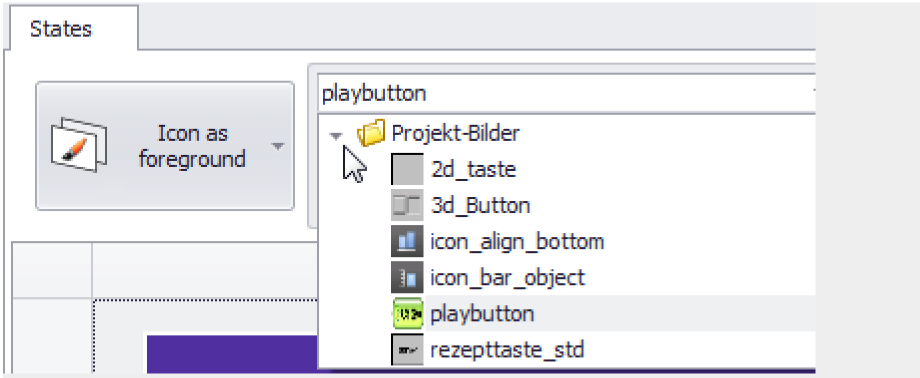


Figure 257: <playbutton> icon selected as foreground after clicking on the arrow for the Project images folder

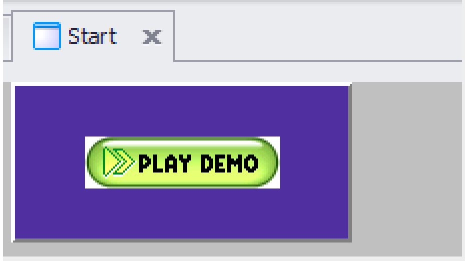


Figure 258: Start screen with function key visualization object configured with an icon in the foreground and a blue color as the background

2. Linking a function key visualization object to a function

In order to make it possible for device operators to run a function on the device screen at runtime, the function first needs to be assigned to the visualization object.

→ If a function key visualization object is used to trigger functions that affect graph, parameter list, image, video player, webcam, HTML Viewer, list box, recipe, or container visualization objects, all the visualization objects must be on the same screen.

Prerequisites
A function key visualization object must have already been created and must be on the screen.

- ▶ Make sure that the visualization object is selected.
- ▶ In the Group field, select the group that contains the function you want, e.g., <Graph>.
  For a description of which groups contain which functions, please refer to: → "How functions are grouped and where the corresponding descriptions are found", page 388.
- ▶ Click on the drop-down menu arrow for the Function field.
- ▶ Select the function you want, e.g., <Zoom In>. If the function you want is not found in the list, you may have to select a different function group.

1. When you select a function, the corresponding decimal number (hexadecimal number) will be automatically shown in the Function No. field. This will be the value that you will need to use if you want to trigger the function with the corresponding PLC. This value cannot be edited.

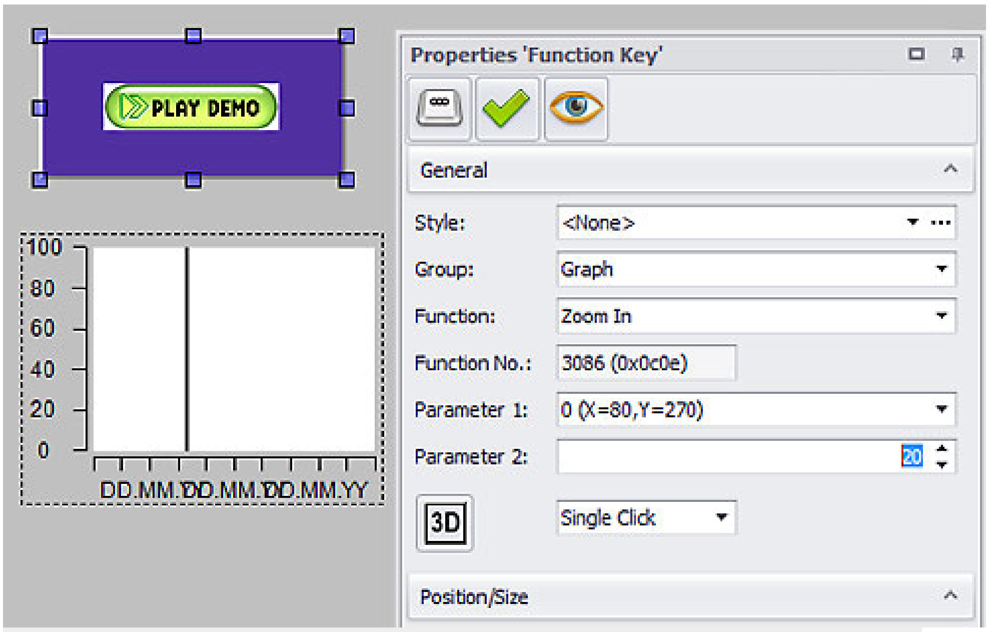
   If the function requires parameters, one or both of the following fields will appear in the properties pane: Parameter 1, Parameter 2.

   - ▶ If you need to enter additional information concerning parameters, enter it into the Parameter 1, Parameter 2 fields, e.g., <0 (X=50, Y=140)> and <20>.

- Instead of Single Click, select the Double Click option.
- Start the Inspector by pressing the F5 key.

The button will be displayed on the PC. Tapping twice on this button will run the selected function.

Linking a function key to the <Zoom In> function



Properties 'Function Key'

General

Style: <None>

Group: Graph

Function: Zoom In

Function No.: 3086 (0x0c0e)

Parameter 1: 0 (X=80,Y=270)

Parameter 2: 20

3D

Single Click

Position/Size

Figure 259: Screen with function key and graph visualization objects; to the right is the Properties 'Function Key' pane; Parameter 1 is being used to select the visualization object, while Parameter 2 is being used to enter a zoom factor of 20x.

See also

→ section "Function keys", page 386
→ section " Function key", page 392
→ section "User Defined Keyboards", page 316
→ section "Controlling functions with the PLC", page 602

3.6.3 Function key

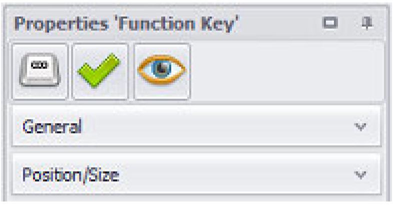
Function key visualization objects are a key element in working with Galileo. They are used whenever a function needs to be applied to another visualization object. Function key visualization objects are usually shown in the form of a button on the device screen, and the Style property can be used to define the appearance of this button. The following settings are available:

If a function key visualization object is clicked on or tapped on on the device screen by a device operator, the corresponding function will be triggered. Here is an example: The graph being displayed on a device screen will be zoomed into when the device operator taps on a function key that was linked to the **Zoom In** function when the application was created. The properties for a function key can be used to link it to the function that needs to be triggered. In addition, it may be necessary to define parameters in the General section of the Properties 'Function Key' pane depending on the specific function that has been selected. The properties for a function key can be subdivided into the following groups:

- General and Position/Size; used to configure a variety of parameters that define how the function key will be displayed on the device screen, as well as to assign a function.
- 👁 Function key visibility; used to set the conditions that define when the function key should not be visible.
- ✓ Function key accessibility; used to set the conditions that define when the function key should not be editable.
- ⌨ Function key states; used to configure the function key's appearance based on its state.

Prerequisites
In order to be able to follow the descriptions below, you must first have clicked on a function key on the current screen

Controls\ tab



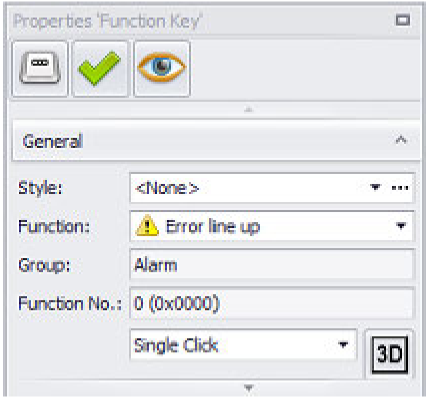
'Function Key' properties	
⌨	→ " States", page 396
✓	
👁	
	→ section "General", page 393

3.6.3.1 General

The General section can be used to configure all the properties that affect how the visualization object will be displayed, with the exception of its size and position.

A context-sensitive filter in the Function field makes it possible to quickly find the function you want.

Controls\ Function Key


Properties 'Function Key'

General

Style: <None>

Function: Error line up

Group: Alarm

Function No.: 0 (0x0000)

Single Click

Figure 260: Properties 'Function Key'\General pane

Style

Settings that affect the appearance of visualization objects can be bundled and stored as a style. After this, it will be possible to use the style's name to use these settings at any time and apply them to new visualization objects of the same type throughout the entire project. In addition, when changes are made to a style, the changes will automatically be applied to all visualization objects using the style, ensuring that all visualization objects using the same style will have a consistent appearance. For detailed information on how to create a style, please refer to → section "Style management", page 458.

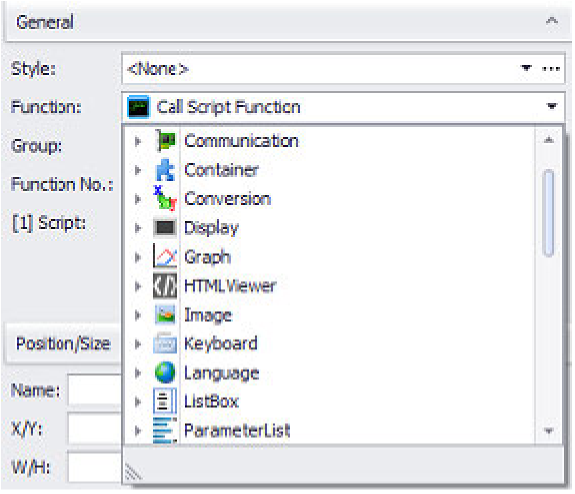
Group

All the functions that can be triggered by tapping on a button on the device screen are organized by groups. These groups, in turn, are based on the objects to which the functions they contain apply. For example, the functions for the parameter list object will be found in the ParameterList group. To figure out which group you should select, first determine the object to which the function should apply.

Function

The function that will be triggered when a device operator taps on the button on the device screen. Only the functions belonging to the selected function group will be available for selection. For a description of what the triggered function will do with the corresponding object or visualization object, please refer to the description for the specific function you want, e.g., → "Error line up", page 886

The Function field features a context-sensitive input functionality that makes it possible to intuitively find and select functions with a search that uses functional groups.



General

Style: <None>

Function: Call Script Function

Group:

Function No.:

[1] Script:

Position/Size

Name:

X/Y:

W/H:

Function No.

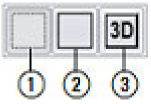
This function number is the number than needs to be entered into the Control data block's 4th data word if you want the function to be triggered by the PLC. For a description of how to trigger functions using the PLC, please refer to the → "Controlling functions with the PLC", page 602 section.

Parameters 1 / 2

Whether or not the Parameter fields appear in the Properties pane will depend on the specific function selected in the Function field. Certain functions require for additional parameters to be specified. For a description of the specific parameters required, please refer to the description for the specific function you want, e.g., → "Page up", page 837

Frame 3D

The visualization object can be displayed with any one of three different designs.



1 no frame

2 with frame

3 3D frame

Selection mode on panel Single Click

The following modes can be adjusted for visualization objects with a keyboard:

- Single Click
  Tapping once on a list item or a button on the device screen will activate the selection.
- Delayed
  This option is ideal for implementing a safety function designed to protect against accidental actions. In order for a button on the device screen to be activated, it will first have to be pressed and held (long press) for the configured delay time.

By default, this delay time will be 0.5 s, but it can be changed in *Home\Settings\Runtime tab\Touch Settings*.

- Double Click
  Tapping twice on a list item or button on the device screen will select the item / activate the button. By default, the double-click speed will be 0.5 s for a double-click, but it can be changed in*Home\Settings\Runtime tab\Touch Setting*.

3.6.3.2 States

The number of states for a function key visualization object will depend on the function selected for it.
In most cases, the states will be *Off* and *On*.

The basic configuration parameters for the visualization object's foreground and background are described in detail in the following section: Defining text series

3.7 List box - used to enable users to select options

The list box visualization object makes it possible for users to quickly and directly select an option from a list box or a drop-down menu.

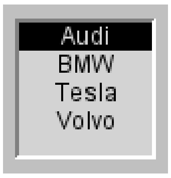


Figure 261: List box example

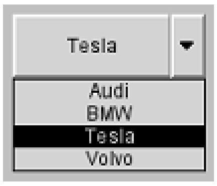


Figure 262: Drop-down menu example

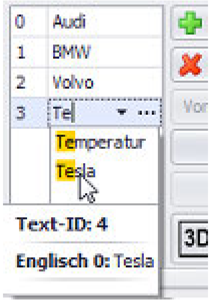
As soon as you add the object to a screen, the selection list will be generated in the Properties pane for the visualization object.

The properties for this visualization object can be used to customize how the object will be displayed, including features such as its position and size. A list box consists of one or more list box options.

3.7.1 How list boxes work

Once you add a list box object to a screen, you will need to define whether the values for the corresponding options will consist of texts or numbers.

Once you do this, you can use the ✚ Add button to add the options you want (you can either enter an option directly into the corresponding field or use the drop-down menu arrow at the end of the field in order to select an option from existing standard texts).



0	Audi
1	BMW
2	Volvo
3	Te

Text-ID: 4
Englisch 0: Tesla

Figure 263: Drop-down menu example

Once you have completed the steps above, you will need to assign at least one tag as an index or storage location:

- Index - Numbers the list and controls how values are assigned

The order index for the selected option will be stored in this tag, starting with 0 (e.g., 1 if "bar"is selected and 0 if "0.5" is selected). The tag type can be integer, byte, or larger.

- Storage location tag - stores the selection made by the user and provides it for further processing, e.g., for the PLC.

The input entered will be checked to see if it is valid, and a corresponding prompt will be output.

3.7.2 Adding a list box

- ▶ Create a screen or open the screen where you want to place the list box.
- ▶ Go to *Controls\List box*.
- ▶ Position the object
  Move the cursor to the screen and draw the area for the visualization object.

A pre-formatted list box will be shown on the screen.

To configure the gauge's appearance, follow the steps below:

- ▶ Click on the visualization object.

  The General section in the Properties pane will show all of the list box's properties.
- ▶ Change the list box's size as needed either by entering the list box's coordinates and width/height in Properties\Position/Size or by moving the cursor over a corner of the visualization object and dragging the corner in order to resize the list box.

3.7.3 Using list boxes Display values

List boxes are normally used to display values so that a user can select one of them.

The following steps show how to create a simple list box. The use case that follows describes a typical scenario, as well as the results expected on the device screen.

To create a list box, follow the steps below:

1. → "Create a tag and set limits for it", page 399
2. → "Position the visualization object and assign the tag to it", page 399
3. → "Assigning a name", page 399Assigning a name
4. → "Define the list options", page 399Define the list options

List box example

Say you want to create a drop-down menu list box with four options and you want the list box to be called <Car>.
In addition, you want all the list options to be visible so that users can choose any of them.

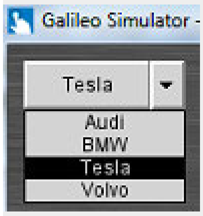


Figure 264: Device screen with list box containing four options

To implement the list box from this example, follow the steps below:

Create a tag and set limits for it

- Create an internal tag of data type ab String called <Dropdown_Content>.
- In the Properties 'Dropdown_Content' pane, enter the following limits for the tag:
  Length <10>,
  Default< Auto>,
  Encoding <ISO-8859-1>
  Minimum <0>

The tag should only be available locally – WebVisu □ disabled

Position the visualization object and assign the tag to it

- Open the <Start> screen or
  create a screen called <Listbox>.
- If applicable, add a screen change from the <Start> screen to the <Listbox> screen.
  Described in section:
  → "Step 8: Adding screen changes", page 189
- Go to *Controls\List box*.
- Position the object
  Move the cursor to the screen and draw the area for the visualization object. A pre-formatted and enabled list box will be shown on the screen.
- In the *General* section of the Properties pane, assign the following values:

Values:	<Text>
Selected index:	<Dropdown_Content>
Save selected value to:	<None>
Is Drop-Down	☑
Maximum displayed rows:	<4>
Selection style:	<Invert colors>

Assigning a name

- In the Controls Manager pane, enter <Car> as the name for the list box.

Define the list options

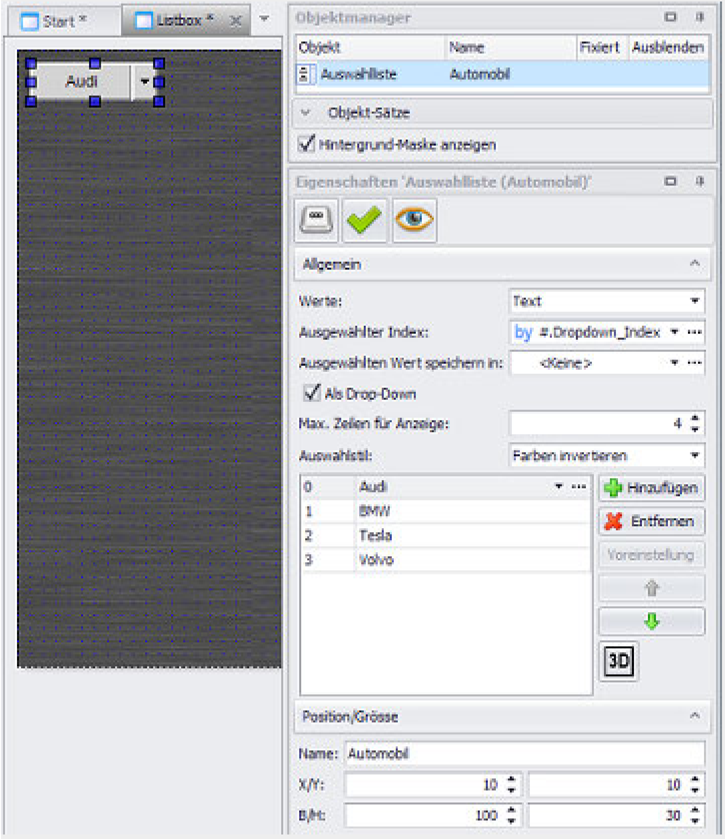
- Use ➕ Add to add four options.
  (IDs 0–3)
- Click on the input row with ID 0 and enter <Audi> as the text.

- ▶ Click on the input row with ID 1 and enter <BMW> as the text.
- ▶ Click on the input row with ID 2 and enter <Tesla> as the text.
- ▶ Click on the input row with ID 3 and enter <Volvo> as the text.

Define the position and name

- ▶ In Properties\Position/Size, enter the following name: <Car>.
- ▶ Adjust the list box's size and position
  X/Y: <10/10>
  W/H: <120/40>

Controls\List box\Properties



Objekt	Name	Fixiert	Ausblenden
Auswahlliste	Automobil		

☑ Hintergrund-Maske anzeigen

Werte: Text
Ausgewählter Index: by #.Dropdown_Index
Ausgewählten Wert speichern in: <Keine>
☑ Als Drop-Down
Max. Zeilen für Anzeige: 4
Auswahlstil: Farben invertieren

0	Audi
1	BMW
2	Tesla
3	Volvo

Name: Automobil
X/Y: 10 10
B/H: 100 30

Figure 265: Properties 'List box'; settings for this example

- ▶ Start the Galileo Simulator with function key F5 or by clicking on Home\ . Build.

The list box will get the following options at runtime:

- Audi
- BMW
- Tesla
- Volvo

3.7.4 List box

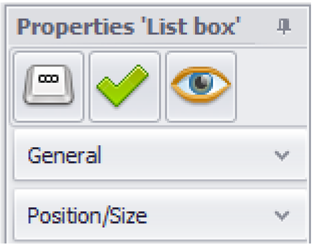
A list box will display its options on the device screen
and is purely an input/output object.

There are multiple navigation functions that can be applied to this visualization object.

It is possible to define two different states for this visualization object. This makes it possible to change the visualization object's appearance as soon as the object is enabled or disabled.

Prerequisites
In order to be able to follow the descriptions below, you must first have clicked on a list box visualization object on the current screen.

Controls\ tab



'List box' properties

→ "States", page 403

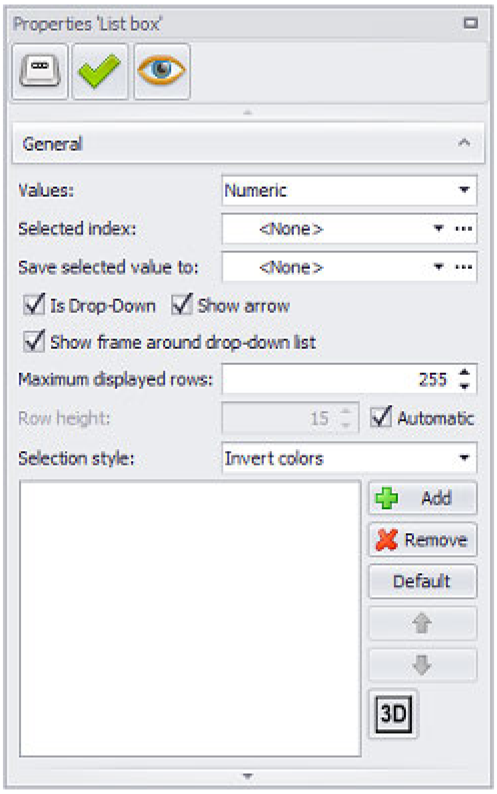
✓

→ section "General", page 401

3.7.4.1 General

The General section can be used to configure all the properties that affect how the visualization object will be displayed, with the exception of its size and position.

Controls\ List box tab



Properties 'List box'

General

Values: Numeric

Selected index: <None>

Save selected value to: <None>

☑ Is Drop-Down ☑ Show arrow

☑ Show frame around drop-down list

Maximum displayed rows: 255

Row height: 15 ☑ Automatic

Selection style: Invert colors

Figure 266: Properties 'List box' pane

Values:
Used to define the type of list items. The following options will be available for selection:

- Numeric
- Text

Selected index:
Index tag
Numbers the list and controls how values are assigned.
The order index for the selected option will be stored in this tag, starting with 0.
The tag type can be Integer, Byte, or larger.

Save selected value to:
Storage location tag
Stores the selection made by the user and provides it for further processing, e.g., for the PLC.

Is Drop-Down
If ☑, the list will be output as a drop-down menu.

If □ the checkbox is disabled, it will be output as a list box.

The option selected will determine whether Maximum displayed rows:or Background will be shown as an option

Maximum displayed rows:
Depends on the option selected for Is Drop-Down
Used to define the number of lines that will be displayed for a drop-down menu.
Valid values: 1–255

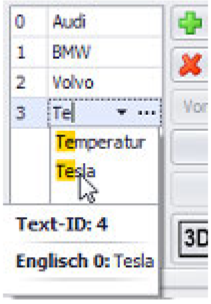
Background
Used to set the background color for the list box as a whole (not for the individual options).

For detailed information on the available colors, please refer to the following section: → section "Color choice - Object's Color Palette", page 234

Selection style:

Shows how the currently selected option will be highlighted:

- Frame - A box enclosing the current selection will be shown
- Invert colors - The background color of the row that is currently selected will be changed in order to highlight the corresponding option



List box pane

Shows the options in the list box.

Each option can be assigned a text or a number - as a name - depending on the option selected for "Values." This name will then be displayed for the corresponding option in the list box.

What the input row does

This row can be used to enter a text or number directly or to select from a list of existing standard texts.

✚ Add

Clicking on this button will add an option.

Each option consists of an index number and a placeholder for the corresponding value, and will be displayed in the list box pane, next to the buttons, for editing.

You can add up to 65,535 list options.

✖ Remove

Deletes the selected option from the list

Default settings

If no index has been assigned, this button can be used to define which list option will be preset and highlighted when the list box is opened.

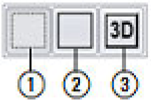
Up/Down ⬆ ⬇

This button will be enabled if an option is first selected in the list box pane. The selected option will be moved one place up every time you click on ⬆ and one place down every time you click on ⬇ . The options' place in the list determines the order in which they will ultimately be shown in the list box, as well as their index.

3D

Display style

The visualization object can be displayed with any one of three different designs.



①	No frame
②	with frame
③	3D frame

3.7.4.2 States

List box visualization objects have two states: Collapsed and Row.

The basic configuration parameters for the visualization object's foreground and background are described in detail in the following section: Defining text series

3.7.4.3 List box functions

The functions for List box focus mostly on ways to scroll entries.

To run List box functions you will need either a Function Key or a index variable.

The call can also be triggered by a Function Key visualization object.

→ The Function Key and List box visualization objects must be found on the same screen.

You can design the button for this Function Key according to your specific needs, e.g., by adding a background color or image to it.

Furthermore, an entry can also be selected directly from the list, if the entry is currently visible.

→ To quickly insert the object, use a function key with a special function on the screen 💡

List box: First Line

Calling the *First line* function for a list box visualization object will make the highlight bar jump to the first entry in the list box.

To create a function key that can be used to call the function, follow the steps below:

- Click on *Controls\Function Key*.
- Position the object
  Move the cursor to the screen and draw the area for the visualization object.
- Under properties of the Function field, select the visualization object List box as function *first line*
- Start the Galileo Simulator with function key F5 or
  by clicking on Home\ . Build.

Regardless of which list box entry is selected on the device screen at runtime, the highlight bar will jump to the first entry when the button for the function key visualization object is clicked or tapped on.

Number of the Function	5632 (0x1600)
Executable from the PLC	No
Function with ON/OFF state	No
Parameter 1 List box	List box on current screen
Function available in standard screens	Yes
Function available with keyboards	No
Function available in script	Yes, with the name of the list box
Function available in application entry script	Yes, with the name of the list box

List box: Last Line

Calling the Last line function for a list box visualization object will make the highlight bar jump to the last entry in the list box.

To create a function key that can be used to call the function, follow the steps below:

- Click on *Controls\Function Key*.
- Position the object
  Move the cursor to the screen and draw the area for the visualization object.
- Under properties of the Function field, select the visualization object List box as function *Last line*
- Start the Galileo Simulator with function key F5 or
  by clicking on Home\ . Build.

Regardless of which list box entry is selected on the device screen at runtime, the highlight bar will jump to the last entry when the button for the function key visualization object is clicked or tapped on.

Number of the Function	5633 (0x1601)
Executable from the PLC	No
Function with ON/OFF state	No
Parameter 1 List box	List box on current screen
Function available in standard screens	Yes
Function available with keyboards	No
Function available in script	Yes, with the name of the list box
Function available in application entry script	Yes, with the name of the list box

List box: Line down

Calling the Line down function for a list box visualization object
will make the highlight bar move one entry down in the list box.

To create a function key that can be used to call the function, follow the steps below:

- Click on *Controls\Function Key*.
- Position the object
  Move the cursor to the screen and draw the area for the visualization object.
- Under properties of the Function field, select the visualization object List box as function *Line down*
- Start the Galileo Simulator with function key F5 or
  by clicking on Home\ . Build.

Regardless of which list box entry is selected on the device screen at runtime, the highlight bar will jump one line further when the button for the function key visualization object is clicked or tapped on.

Number of the Function	5635 (0x1603)
Executable from the PLC	No

Function with ON/OFF state	No
Parameter 1 List box	List box on current screen
Function available in standard screens	Yes
Function available with keyboards	No
Function available in script	Yes, with the name of the list box
Function available in application entry script	Yes, with the name of the list box

List box: Line up

Calling the Line up function for a List box visualization object will make the highlight bar move one entry up in the parameter list.

To create a function key that can be used to call the function, follow the steps below:

- Click on *Controls\Function Key*.
- Position the object
  Move the cursor to the screen and draw the area for the visualization object.
- Under properties of the Function field, select the visualization object List box as function *Line up*
- Start the Galileo Simulator with function key F5 or
  by clicking on Home\ . Build.

Regardless of which list box entry is selected on the device screen at runtime, the highlight bar will jump back one line when the button for the function key visualization object is clicked or tapped on.

Number of the Function	5634 (0x1602)
Executable from the PLC	No
Function with ON/OFF state	No
Parameter 1 List box	List box on current screen
Function available in standard screens	Yes
Function available with keyboards	No
Function available in script	Yes, with the name of the list box
Function available in application entry script	Yes, with the name of the list box

List box: Page down

Calling the Page down function for a List box visualization object will scroll down one page in the list box.

To create a function key that can be used to call the function, follow the steps below:

- Click on *Controls\Function Key*.
- Position the object
  Move the cursor to the screen and draw the area for the visualization object.
- Under properties of the Function field, select the visualization object List box as function *Page down*
- Start the Galileo Simulator with function key F5 or
  by clicking on Home\ . Build.

Regardless of which list box entry is selected on the device screen at runtime, the highlight bar will jump one page further when the button for the function key visualization object is clicked or tapped on.

Number of the Function	5637 (0x1605)
Executable from the PLC	No
Function with ON/OFF state	No
Parameter 1 List box	List box on current screen
Function available in standard screens	Yes
Function available with keyboards	No
Function available in script	Yes, with the name of the list box
Function available in application entry script	Yes, with the name of the list box

List box: Page up

Calling the Page up function for a List box visualization object will scroll down one page in the list box.

To create a function key that can be used to call the function, follow the steps below:

- Click on *Controls\Function Key*.
- Position the object
  Move the cursor to the screen and draw the area for the visualization object.
- Under properties of the Function field, select the visualization object List box as function *Page up*
- Start the Galileo Simulator with function key F5 or
  by clicking on Home\ . Build.

Regardless of which list box entry is selected on the device screen at runtime, the highlight bar will jump one page back when the button for the function key visualization object is clicked or tapped on.

Number of the Function	5636 (0x1604)
Executable from the PLC	No
Function with ON/OFF state	No
Parameter 1 List box	List box on current screen
Function available in standard screens	Yes
Function available with keyboards	No
Function available in script	Yes, with the name of the list box
Function available in application entry script	Yes, with the name of the list box

List box: Select

→ Not all device types support this function

Number of the Function	5638 (0x1606)
Executable from the PLC	No
Function with ON/OFF state	No
Parameter 1 List box	List box on current screen
Parameter 2	Value within WORD range (0 - 65535)

Function available in standard screens	Yes
Function available with keyboards	No
Function available in script	Yes, with the name of the list box
Function available in application entry script	Yes, with the name of the list box

List box: SelectVar

→ Not all device types support this function

Number of the Function	5639 (0x1607)
Executable from the PLC	No
Function with ON/OFF state	No
Parameter 1 List box	List box on current screen
Parameter 2	Tag of data type STRING The tag should contain the directory where the video files are stored. Example: `"C:/temp/"`
Function available in standard screens	Yes
Function available with keyboards	No
Function available in script	Yes, with the name of the list box
Function available in application entry script	Yes, with the name of the list box

3.8 Gauge

In real-life applications, gauges are used when you want the person looking at them to be able to notice tendencies and value changes quickly and easily. Accordingly, high-precision readings are not a key aspect when using gauges.

The classic example of a gauge is a speedometer.

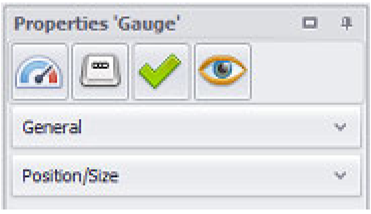
A gauge spans a scale between a minimum and a maximum. These two values can be entered as fixed values, or tags can be assigned to them so that the minimum and maximum will be changeable. In addition, the scale is subdivided into segments. Moreover, the exact tag value can be displayed as a value in addition to the pointer. If an upper and a lower limit are defined for the tag, the gauge can highlight these ranges on the scale in color.

There are no functions that can be applied to a gauge object.

If this visualization object is used multiple times to display a large number of system components with the same parameter sets, it is recommended to use parameters. Using parameters will optimize your work when it comes to assigning tags to the visualization object.
For detailed information on how to work with parameters, please refer to → section "Using user controls to optimize work", page 279.

Following is a description of all the configuration options for the gauge object. Please note that the Properties 'Gauge' pane only contains a subset of the properties found in the Extended properties "Gauge" dialog box.

Controls\ tab



Properties 'Gauge'

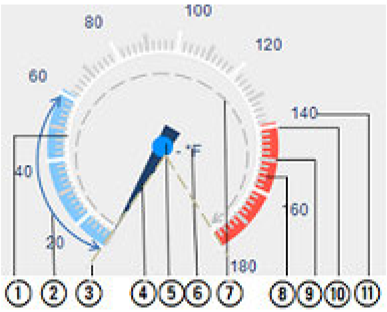
General

Position/Size

	Properties 'Gauge'
	→ " Extended properties "Gauge"", page 410
	Defining text series
	→ "General", page 409

The ✓ accessibility button will only be shown in the Properties pane if a tag and a keyboard have been assigned to the visualization object, as it would not make sense to define accessibility conditions otherwise.

Gauge definitions



①	Circular arc
②	Context
③	Start angle
④	Arrow
⑤	Center point
⑥	Value display
⑦	Open angle
⑧	Fine scale
⑨	Medium scale
⑩	Coarse scale
⑪	Scale text

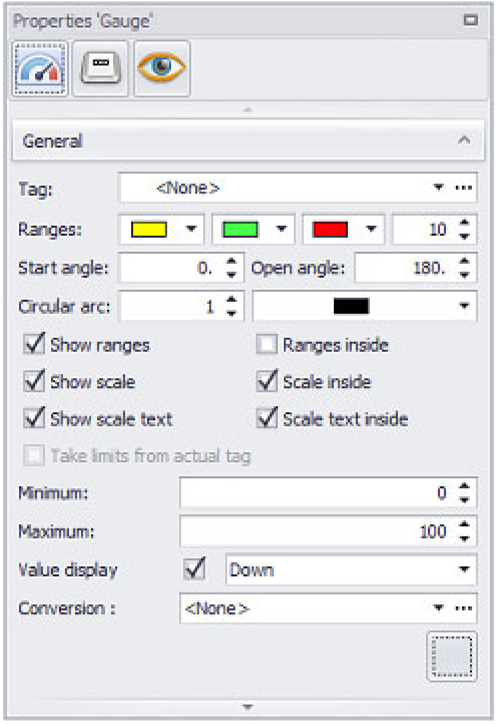
Figure 267: Gauge on device screen with its elements

→ The gauge object will not be shown in a project until it is drawn to a certain specific size that will depend on the selected font sizes and other factors. Whenever the gauge object is smaller than this required size, it will be shown as a an X instead.

3.8.1 General

The General section contains the most important properties that affect how the visualization object will be displayed and that can be configured directly.
As a subset of Extended properties 'Gauge', the description is under the → "General tab", page 411.

Controls 📈 *Graph*



Properties 'Gauge'

General

Tag: <None>

Ranges: 🟨 🟩 🟥 10

Start angle: 0. Open angle: 180.

Circular arc: 1 ■

☑ Show ranges □ Ranges inside

☑ Show scale ☑ Scale inside

☑ Show scale text ☑ Scale text inside

□ Take limits from actual tag

Minimum: 0

Maximum: 100

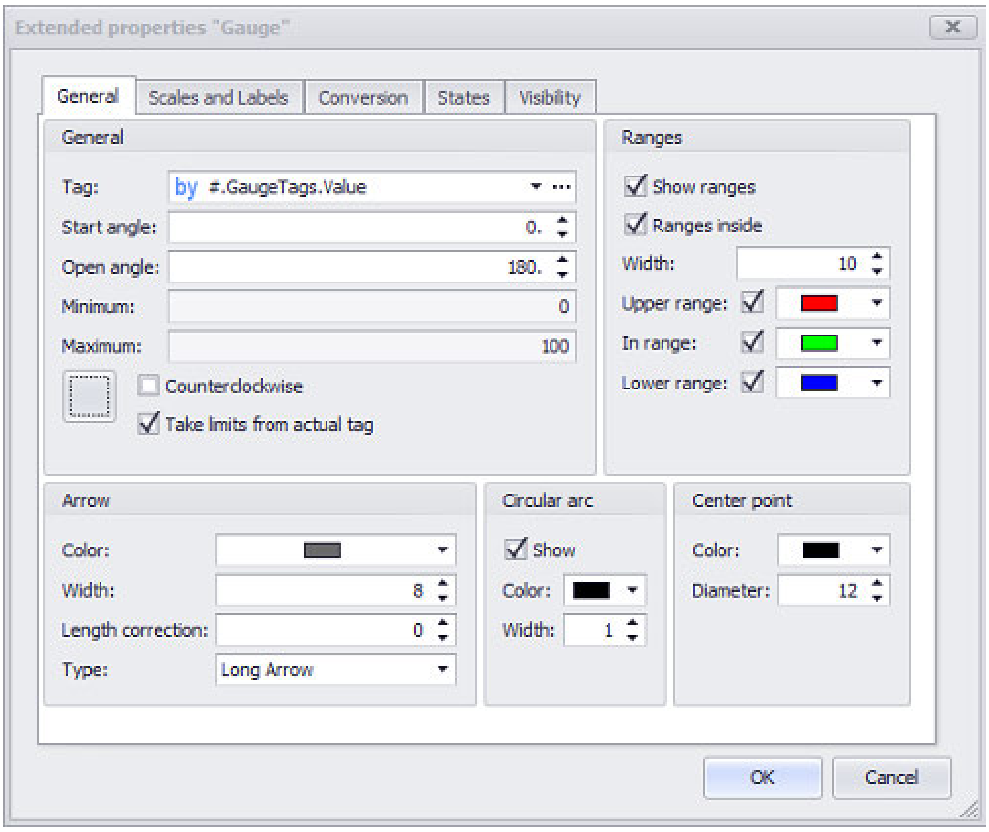
Value display ☑ Down

Conversion : <None>

Figure 268: Window Properties Gauge General

3.8.2 Extended properties "Gauge"

Following is a description of all the properties of the gauge visualization object.



Extended properties "Gauge"

General | Scales and Labels | Conversion | States | Visibility

General

Tag: by #.GaugeTags.Value

Start angle: 0.

Open angle: 180.

Minimum: 0

Maximum: 100

☐ Counterclockwise

☑ Take limits from actual tag

Ranges

☑ Show ranges

☑ Ranges inside

Width: 10

Upper range: ☑ 🟥

In range: ☑ 🟩

Lower range: ☑ 🟦

Arrow

Color: ■

Width: 8

Length correction: 0

Type: Long Arrow

Circular arc

☑ Show

Color: ■

Width: 1

Center point

Color: ■

Diameter: 12

Figure 269: Window Extended properties 'Gauge'\General tab

3.8.2.1 General tab

3.8.2.2 General

Variable

The visualization object will show the value of the selected tag on the device screen at runtime. Clicking on the drop-down menu arrow will open a list of available tags (BYTE, WORD, DWORD, FLOAT). Please note that in order for the tag you want to be shown in this list, the tag must first be imported or created in *Project Tree\Tags*.

A parameter can also be specified instead of a tag. In this case, the Tag field will turn into a Type field. In order to be able to work with parameters, the visualization object must be placed on a subscreen. For detailed information on how to work with parameters, please refer to → "Using input parameters", page 279.

Values do not necessarily have to be displayed or entered the way the PLC needs them or the way they are stored in the tags. The Value Conversion Manager can be used to convert tag values and assign a variety of units to them before they are shown by visualization objects. The way value conversion works is described in greater detail in a different section – please refer to → "Conversions", page 921.

Button ...

Limits

Clicking on the ... icon will show the limits for the value tag in a separate dialog box. You can also reach this dialog box via *Project Tree\Tags\<Value tag>\Properties*. When a tag is created, it will already be assigned limits based on the value range defined for it. This value range can be signed or unsigned and will depend on the relevant tag's data type.

The limits can be adjusted as necessary within the value range. In the case of bit tags, you will only be able to specify a default value.

For a description of the tag properties and limits, please refer to: → section "Numeric tag limits", page 563.

Tag Address

This field will only appear if you select a PLC tag, i.e., it will not appear for internal tags.
It will show the PLC address for the selected tag. The properties for PLC tags are described in greater detail in a different section – please refer to → section "Tag address", page 569.

If a single tag is transmitted to more than one communication, the corresponding settings will need to be configured in *Home\Gateway*. For detailed information on how to work with a gateway, please refer to → section "Using the gateway functionality to communicate with multiple PLCs", page 596.

Start angle / Open angle

Start angle is the angle at which the scale will start. The scale will span an Open angle starting from this start angle. The Counterclockwise checkbox should be pointed out within this context. This checkbox is disabled by default, meaning that the scale will extend clockwise, starting at the left end of the open angle and ending at its right end. The Start angle and Open angle settings can be freely defined within a range of 0° to 360°, accurate to two decimal places. When the application is built using the visualization software, the pointer will point at the position defined in Start angle.

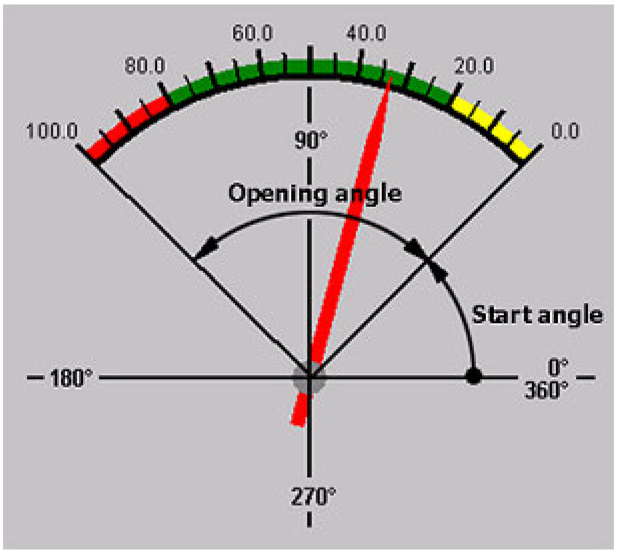


Figure 270: Scale with a start angle of 45°, an open angle of 90°, and the Counterclockwise checkbox enabled

Minimum

The display range for the gauge will start from the minimum defined in this field. Please note that even if the value for the corresponding tag is lower than this minimum, the gauge pointer will not move beyond this minimum. The Minimum field will only be enabled for input if the Take limits from actual tag checkbox is disabled.

→ Please note that the gauge's maximum is not the same as the tag's maximum!

The minimum can be set as a constant value or as a dynamic tag-based value. To set it as a constant value, simply enter the number you want into the corresponding field. To set it as a dynamic limit instead, enter the tag you want into the Minimum field. For detailed information on how to set the minimum as a dynamic value, please refer to → "Numeric tag limits", page 563

Maximum

The display range for the gauge will end at the maximum defined in this field. Please note that even if the value for the corresponding tag is greater than this maximum, the gauge pointer will not move beyond this maximum. The Maximum field will only be enabled for input if the Take limits from actual tag checkbox is disabled. The value entered into the Maximum field will be interpreted as a fixed value.

→ Please note that the gauge's maximum is not the same as the tag's maximum!

The maximum can be set as a constant value or as a dynamic tag-based value. To set it as a constant value, simply enter the number you want into the corresponding field. To set it as a dynamic limit instead, enter the tag you want into the Maximum field. For detailed information on how to set the maximum as a dynamic value, please refer to → "Numeric tag limits", page 563

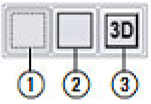
Take limits from actual tag

This property will stop being grayed out as soon as a tag is assigned. If the checkbox is enabled, the minimum and maximum for the tag shown in the Tag field will be used as the minimum and maximum limits for the visualization object. This means that the corresponding scaling can be defined freely. For example, a speed indicator can show a scale of up to 300 km/h even if the tag will not reach a value of more than 200 km/h. This also means that cases such as the following are possible: it is possible for a tag of data type BYTE not to have a minimum or maximum, i.e., for the tag to assume unsigned values of 0 to 255 or signed values of -128 to +127. However, say that you want the visualization object to only show the range between 20 and 30 with a fine scale. In this case, you would need to enable the Open Limit checkbox for the tag and set a minimum of 20 and a maximum of 30 for the visualization object.

If the checkbox is disabled, the range shown will be defined based on the values in theMin and Max fields, i.e., the range will be a fixed range.

Frame 3D

The visualization object can be displayed with any one of three different designs.



1. no frame
2. with frame
3. 3D frame

Counterclockwise

□ Counterclockwise

The pointer will move clockwise if the checkbox is disabled.
The Scale ends at the Start angle.

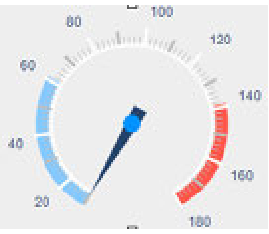


Figure 271: Scale with Start angle 300° and Open angle 300°

☑ Counterclockwise

The pointer will move counterclockwise if the checkbox is enabled.
The Scale begins at the Start angle.

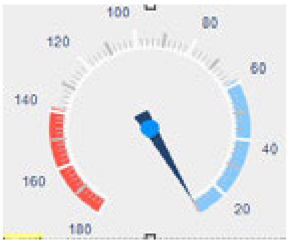
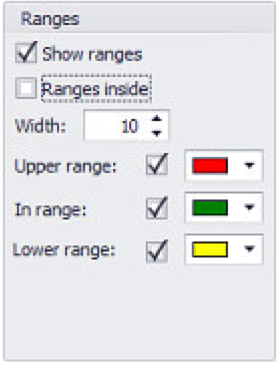


Figure 272: Scale with Start angle 300° and Open angle 300°

3.8.2.3 Ranges

The properties in this section can be used to subdivide the scale into up to three ranges, which are not the same as the scale's segments. which are not the same as the scale's segments. These three ranges will be shown as colored bars along the gauge's arc and can be used as visual aids for identifying value ranges.

Controls\Gauge\Extended properties "Gauge"\General tab\General



Ranges

☑ Show ranges

☐ Ranges inside

Width: 10

Upper range: ☑ 🟥

In range: ☑ 🟩

Lower range: ☑ 🟨

Figure 273: Window Ranges

Show Ranges

If this checkbox is enabled, the defined subranges will be shown with the corresponding colors. If it is disabled, they will not.

The subranges that can be defined are as follows: Upper range, In range, and Lower range. These subranges will be used as defined subranges only if the corresponding checkboxes are enabled.

Ranges inside

The subrange color bars can be displayed either within or outside the scale. This checkbox is used to specify which option you want.

☐ Ranges inside

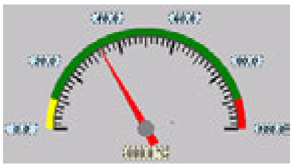


Figure 274: The subrange color bars being displayed outside the scale

☑ Ranges inside

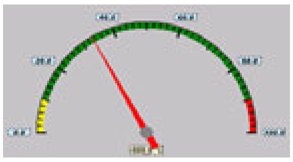


Figure 275: The subrange color bars being displayed within the scale

Width

The width of the subrange color bars displayed without or outside the scale can be defined with this settings.

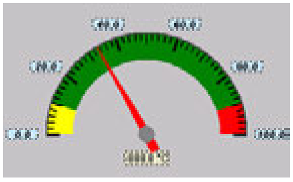


Figure 276: Three subrange color bars with a width of 40 pixels, within the scale

Upper range

The Upper range of the display scale is the range between the Upper Limit of the tag in the Tag field and the Maximum of the gauge as specified in *Extended properties "Gauge"\General*. In order for this lower range to be displayed, the corresponding checkbox needs to be enabled. A specific color can be assigned to this upper range by clicking on the corresponding drop-down menu arrow. The default color will be red.

In Range

The In range subrange of the display scale is the range between the Upper Limit and Lower Limit of the tag in the Tag field. In order for this lower range to be displayed, the corresponding checkbox needs to be enabled. A specific color can be assigned to the In range subrange by clicking on the corresponding drop-down menu arrow. The default color will be green.

Lower range

The Lower range of the display scale is the range between the Lower Limit of the tag in the Tag field and the Minimum of the gauge as specified in *Extended properties "Gauge"\General*. In order for this lower range to be displayed, the corresponding checkbox needs to be enabled.

A specific color can be assigned to this lower range by clicking on the corresponding drop-down menu arrow. The default color will be yellow.

Color selection for the ranges

You can assign specific colors to the scale's three subranges by clicking on the corresponding color drop-down menu arrows.
The default colors will be as follows: Upper range = red; In range = green; Lower range = yellow.

You can change the color if you want. The following options are available for this purpose:

- Transparent
- Standard Colors
- More Colors...

The most recently used colors will be shown (user-specific).

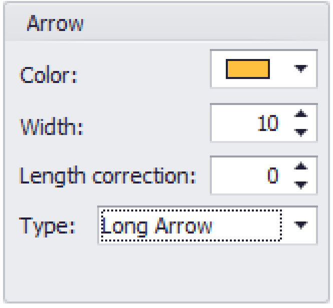
For detailed information on the available colors, please refer to the following section:
→ section "Color choice - Object's Color Palette", page 234

3.8.2.4 Arrow

Changes made to the settings for the pointer will be applied to the gauge object immediately.

When the application is built using the visualization software, the pointer will point at the position defined in Start angle. At runtime, the pointer o the device screen will point at the default value for the tag.

Controls\Gauge\Extended Properties 'Gauge'



Arrow

Color:

Width: 10

Length correction: 0

Type: Long Arrow

Figure 277: Window Arrow

Color

You can change the color if you want. The following options are available for this purpose:

- Transparent
- Standard Colors
- More Colors...

The most recently used colors will be shown (user-specific).

For detailed information on the available colors, please refer to the following section: → section "Color choice - Object's Color Palette", page 234

Width

Used to define the width of the pointer, in pixels, at its widest point.

Length correction

The length correction settings can be used to define the number of pixels by which the pointer should be lengthened so that it will project beyond the scale.

Entering negative values will shorten the pointer so that it is located within the display scale.

Type

This setting provides four different types of pointer.

The pointer is displayed with the pointer at the center point of the gauge. The display of types Long needle and Long arrow starts in front of the center point.

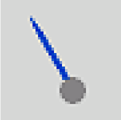


Figure 278: Type: Needle

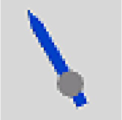


Figure 279: Type: Long

needle

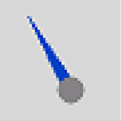


Figure 280: Type: Arrow

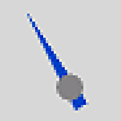
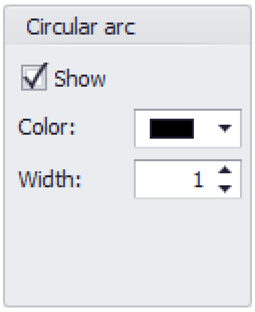


Figure 281: Type: Long arrow

3.8.2.5 Circular arc

The arc's width can be defined in General tab or in *\Properties pane\Extended properties\ \Tab: General\Circular arc*. In addition, the arc can be limited to a maximum size there by enabling the Fix maximum arc radius checkbox. The units for the values entered are pixels.

Controls\ Window Extended properties 'Gauge'



Circular arc

☑ Show

Color: ■

Width: 1

Figure 282: Window Circular arc

Show

If this checkbox is enabled, the arc will be shown.

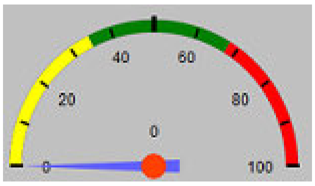


Figure 283: The Circular arc is not shown.

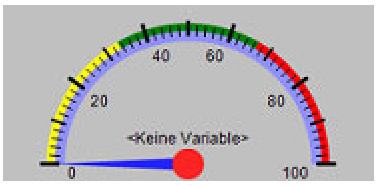


Figure 284: The Circular arc is shown.

Color

You can change the color if you want. The following options are available for this purpose:

- Transparent
- Standard Colors
- More Colors...

The most recently used colors will be shown (user-specific).

For detailed information on the available colors, please refer to the following section:
→ section "Color choice - Object's Color Palette", page 234

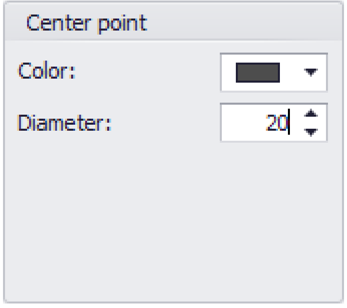
Width

Defines the width of the Circular arc in pixels.

3.8.2.6 Center point

The gauge's pointer is anchored to the center point, which can be configured using the following properties.

Controls\Gauge\Extended Properties 'Gauge'



Center point

Color:

Diameter: 20

Figure 285: Window Center point

Color

You can change the color if you want. The following options are available for this purpose:

- Transparent
- Standard Colors
- More Colors...

The most recently used colors will be shown (user-specific).

For detailed information on the available colors, please refer to the following section:
→ section "Color choice - Object's Color Palette", page 234

Diameter

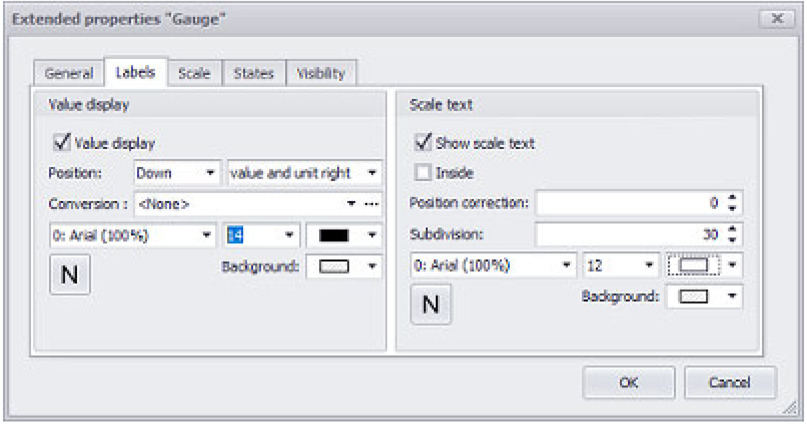
Used to define the width of the center point's diameter in pixels.

3.8.2.7 Scale and label tab

The Scale and labels tab is where the text and scaling are configured for display, i.e., where the settings that affect the scale's appearance are configured.

This "scale" consists of the object's arc and the actual scale dashes. The arc is configured in *\Extended properties dialog box\General\Circular arc*, while the scale subdivisions and dashes are configured in this tab, in the Scale lines section.

Controls\ Gauge\Extended Properties 'Gauge'



Extended properties "Gauge"

Value display

☑ Value display

Position: Down | value and unit right

Conversion : <None>

0: Arial (100%) | 14

N

Background:

Scale text

☑ Show scale text

□ Inside

Position correction: 0

Subdivision: 30

0: Arial (100%) | 12

N

Background:

Figure 286: Window Extended properties 'Gauge'\Scale and labels

3.8.2.8 Scale text

Show scale text

If this checkbox is enabled ☑, the scale label will be shown.

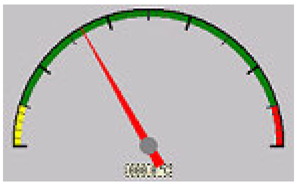


Figure 287: without scale text

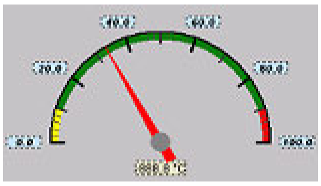


Figure 288: with scale text

→ The gauge display will always be shown with the largest possible size within the drawn object frame.
The size of the gauge object frame drawn in the project will not be changed.

Inside

If this checkbox is enabled ☑, the labels will be placed on the inside the scale (when seen from the pointer's center point).

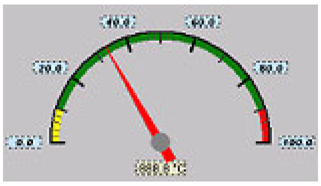


Figure 289: Label outside

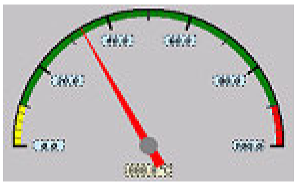


Figure 290: Label inside

→ The gauge display will always be shown with the largest possible size within the drawn object frame.
The size of the gauge object frame drawn in the project will not be changed.

Position correction

The position correction setting makes the scale labels' or arc's radius, relative to the pointer's center point, smaller by the specified number of pixels. The gauge's actual total size will remain the same. Any value within a range of -128 to +127 can be entered.

Positive values = The arc radius will be made smaller

Negative values = The scale labels' radius will be made smaller

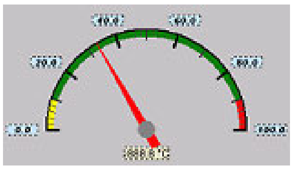


Figure 291: Position correction: 0 pixels - outside the Scale Text

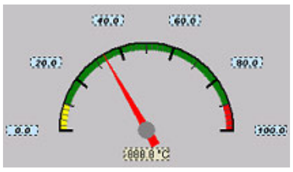


Figure 292: Position correction: 15 pixels - outside the Scale Text

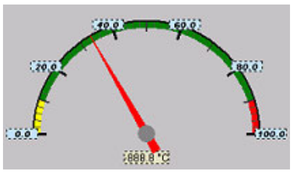


Figure 293: Position correction: -15 pixels - outside the Scale Text

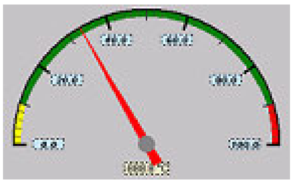


Figure 294: Position correction: 0 pixels - inside the Scale Text

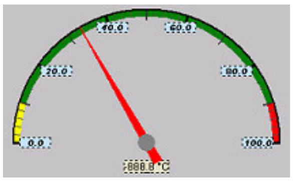


Figure 295: Position correction: 15 pixels - inside the Scale Text

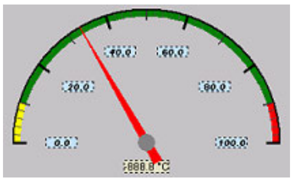


Figure 296: Position correction: -15 pixels - inside the Scale Text

Separation

This property is used to define the interval, measured in scale subdivisions, at which the scale labels should be shown.

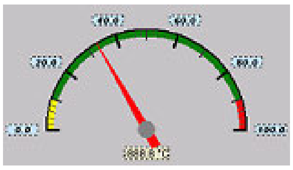


Figure 297: Scale labels with a subdivision value of <20>

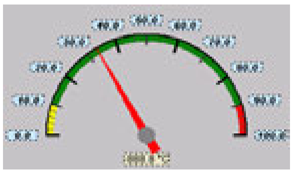


Figure 298: Scale labels with a subdivision value of <10>

For a description of the font, orientation, and background color, please refer to section:.

3.8.2.9 Value display

Value display

This settings can be used to enable and disable the functions in the Value display section.

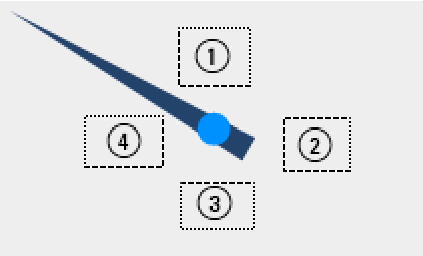
The gauge's value display is an element used to show the current tag value and the corresponding unit.

For a description of the font, orientation, and background color, please refer to section:.

Position

Position describes the value display's position relative to the center point. This property can be set in the first of the two drop-down menus. Please note that this drop-down menu will only be enabled if the Value display checkbox is enabled. In addition, the second of the two drop-down menus can be used to specify whether the value display should display the tag value only, the unit only, or both. This property can be set in the second of the two drop-down menus.

First Selection field



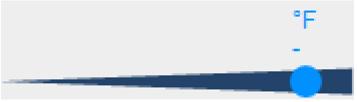
① Top - on top of the Center point

② Right - on the right of the Center point

③ Down - below the center point

④ Left - on the left of the Center point

Second Selection field

Value	Only the value will be displayed	
value and unit right	The value will be displayed followed by the unit on the right	
Value and Unit Top	The value will be displayed and the unit will be displayed above it	
Unit	Only the unit will be displayed	

Values do not necessarily have to be displayed or entered the way the PLC needs them or the way they are stored in the tags. The Value Conversion Manager can be used to convert tag values and assign a variety of units to them before they are shown by visualization objects. The way value conversion works is described in greater detail in a different section – please refer to → "Conversions", page 921.

3.8.2.10 Scale lines

There are three types of scale subdivisions: fine, medium, and coarse. The exact number of segments that each of these subdivisions contains can be defined by using the corresponding Subdivision fields. The display range between the minimum and maximum will first be subdivided into the smallest configurable unit of 1. Numbers smaller than 1 are not permitted. After this, the unit will be multiplied by the value in the Subdivision field in order to get the corresponding scale, i.e., number of equally spaced dashes. An example: if the display range spans from -50 °C to +50 °C, the smallest unit will be a hundredth of a °C. Using a subdivision value of <2> would mean that a scale dash would be added every two hundredths of a °C. The length, width, and color of the dashes can be defined separately for each subdivision.

Show Scale

This checkbox can be used to select whether the scale will be shown or not.

□ ☑ Show Scale

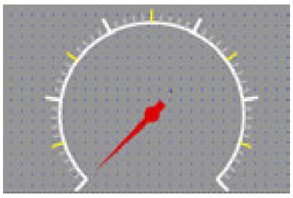


Figure 299: Scale is shown

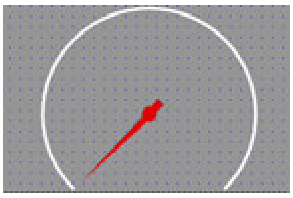


Figure 300: Scale is not shown

Scale Inside

The scale dashes can be displayed either within or outside the arc. This checkbox is used to specify which option you want.

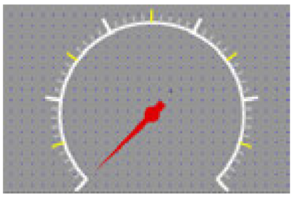


Figure 301: Dashes outside the arc

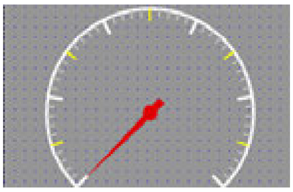
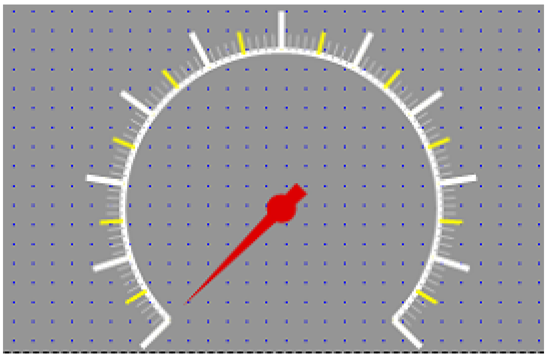


Figure 302: Dashes within the arc

Scale subdivisions

	Length	Width	Color	Subdivision
Fine	8	1	White	1
Medium	12	2	yellow	5
Coarse	20	3	White	10



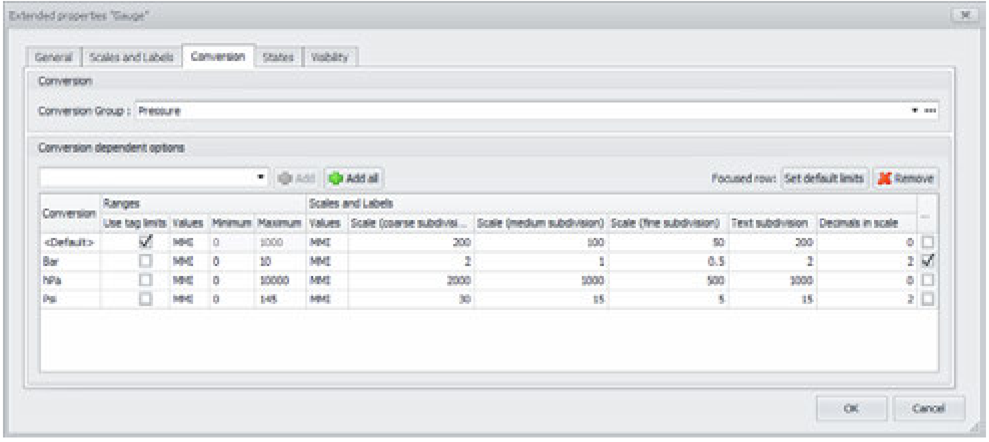
→ The coarse value should be divisible by the medium and fine values so that the scale dashes will be distributed uniformly.

3.8.2.11 Conversion tab

For the displaying of the gauge by using value conversions, the configuration for each value conversion can be performed here.

→ Settings configured here will override the corresponding settings under the Scale and labels tab.

Controls\ Gauge\Extended Properties 'Gauge'



Extended properties "Gauge"

Conversion

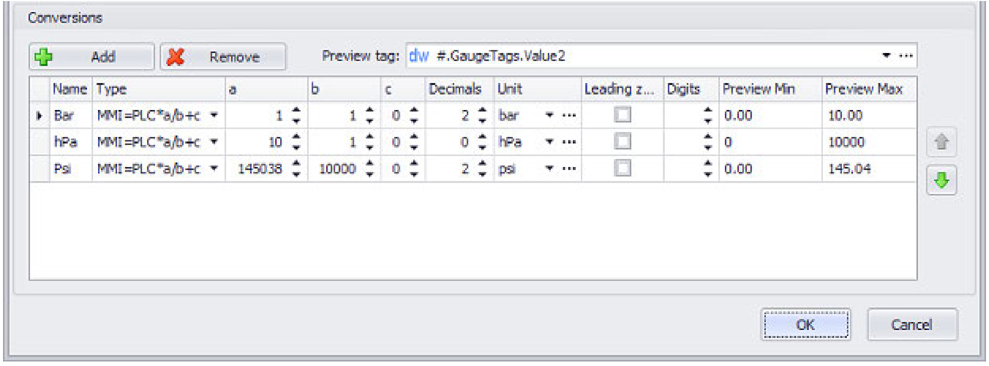
Conversion Group : Pressure

Conversion dependent options

Conversion	Ranges				Scales and Labels						
	Use tag limits	Values	Minimum	Maximum	Values	Scale (coarse subdivisi...	Scale (medium subdivision)	Scale (fine subdivision)	Text subdivision	Decimals in scale	
<Default>	☑	MMI	0	1000	MMI	200	100	50	200	0	□
Bar	□	MMI	0	10	MMI	2	1	0.5	2	2	☑
hPa	□	MMI	0	10000	MMI	2000	1000	500	1000	0	□
Psi	□	MMI	0	145	MMI	30	15	5	15	2	□

Figure 303: Window Extended properties Gauge\Conversions

Controls\ Gauge\Extended Properties 'Gauge'\Conversion



Conversions

Preview tag: dw #.GaugeTags.Value2

Name	Type	a	b	c	Decimals	Unit	Leading z...	Digits	Preview Min	Preview Max
Bar	MMI=PLC*a/b+c	1	1	0	2	bar	□		0.00	10.00
hPa	MMI=PLC*a/b+c	10	1	0	0	hPa	□		0	10000
Psi	MMI=PLC*a/b+c	145038	10000	0	2	psi	□		0.00	145.04

Figure 304: Window Extended properties Gauge\Conversions

- ▶ First select the conversion group in the Conversion section you want to apply.

or

- ▶ Click on ▾ ⋯ and add a value conversion group with the relevant conversions in the Value Conversion dialog box.
- ▶ Then add any number of conversions you want to the conversion group you just created.

If there are no conversion-specific options for the selected conversion group, the options from the <Default> row apply.
Please note that the <Default> row cannot be deleted.

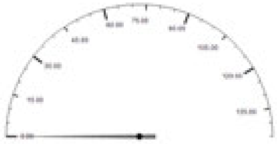
By default, all scale settings will be the same for all conversions, but they can be changed as necessary.

Ranges

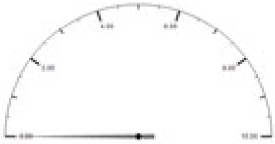
Use tag limits — By setting this option, the limits for minimum and maximum of the range will be taken from the tags own limits.

Value type

MMI — If "MMI" is selected as the Value Type it means that the values entered at Minimum and Maximum are already converted and represent the values to display. In the example above, this is the case for row "Psi": The Maximum value for displaying the values in Psi is exactly 145.



PLC(S) — If "PLC" is selected as the Value type it means that the values entered at Minimum and Maximum are given in the value scale of the PLC value and needs to be converted with the appropriate conversion. In the example above, this is the case for row "Bar": The Maximum value is entered as PLC value of 1000. The conversion "Bar" causes that the last two digits will be handled as decimals. Therefore, the gauge will, after applying the conversion, display the range from 0 - 10.



Minimum
Maximum

Scale minimum and maximum to display (considering the Value Type configuration).

Scales and Labels

Value type

MMI

If "MMI" is selected as the Value type it means that the scales subdivision and label subdivision are already converted and represent the values to use. In the example above, this is the case for row "Psi": The coarse scale subdivision will be performed in gaps of exactly 30 Psi (see gauge visualization above).

PLC(S)

If "PLC" is seleced as the Value type it means that the scales subdivision and label subdivision are given in the value scale of the PLC value and needs to be converted with the appropriate conversion. In the example above, this is the case for row "Bar": The coarse scale subdivision is defined as 200. The conversion "Bar" causes that the last two digits will be handled as decimals. Therefore, the coarse scale subdivision will not be 200 but 2 Bar (see gauge visualization above).

Scales
Coarse
Middle
Fine

Scale subdivision in coarse to fine increments (considering the Value Type configuration).

Label subdivision

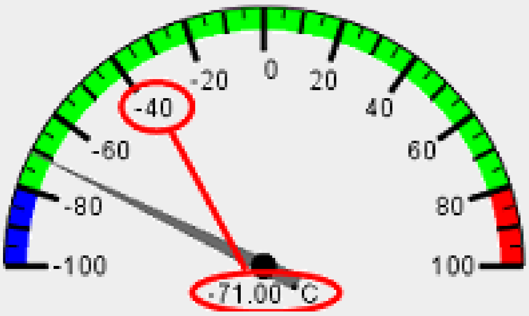
How the scale label will be subdivided (considering the Value Type configuration).

Decimal places in scale

Number of decimal places shown in the scale label
Can be used if you want to use a scale label format different from the value conversion display format, i.e., a whole number in contrast to the exact number.

Example

It would not make sense to display the scale in the same format as the value display, as the latter would always be ".00" in the scale.



Use for preview

Used to specify the conversion that should be used for the draft

For detailed information on the States and Visibility tabs, please refer to the:→ section " States", page 248 and → section "Visibility", page 259 sections

3.8.3 Position/Size

For detailed information on the object's positioning and size, please refer to: → section "Position/Size", page 242

Defining dynamic values with tags

Instead of entering a fixed value, you can link the field to a tag, so that the value can be changed dynamically. To link to a tag, follow the steps below:

- Place the cursor inside the field.
- Right-click to open the context menu and select the Use Tag option.

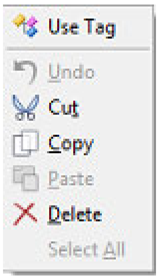
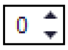
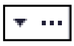


Figure 305: Context menu with Use Tag option

The field will go from looking like this:  to looking like this: .

- Click on the drop-down menu arrow and select the tag that should represent the value.
- Clicking on the ... icon will open the Properties <tag name> dialog box. Make sure that the settings for the tag are correct, e.g, Default, Maximum. If necessary, you can also use tags here in order to define limits. During compilation, the system will check whether there are recursive tag references and, if so, will output an error message.

→ All value tags are always declared in decimal format regardless of the selected display format (decimal, hex, binary, octal, BCD)!

→ All values are PLC values. They will be transmitted to the PLC the way they are, i.e., without being converted.

3.9 Help Button

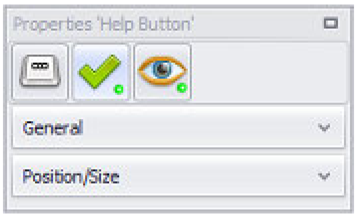
Help button visualization objects can be used to enable device operators to access help information in a help screen on the device. To do so, the operator must tap or click on the help button. Help button visualization objects can be placed on any screen type.

The properties for help buttons can be subdivided into the following groups:

- General and Position/Size; used to configure a variety of parameters that define how the help button will be displayed on the device screen.
- 👁 Help button visibility; used to set the conditions that define when the help button should be visible.

Prerequisites
To be able to follow the descriptions below, make sure you click on a help button on the current screen.

Controls\ tab



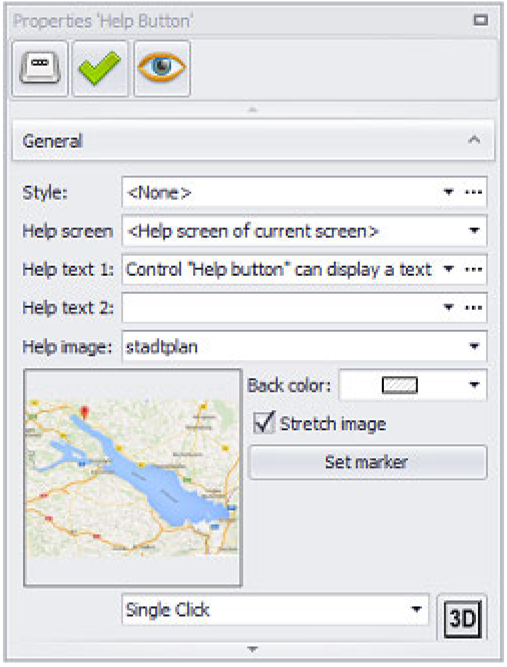
'Help button' properties

▭	→ " States", page 248
✓	
👁	
	→ "General", page 428

General

The Help text 1, Help text 2, and Help image fields in the General section can be used to define the relevant help information contents. In turn, these contents can be linked to the help screen that should display them. Finally, this section can also be used to configure the properties for the help button and the way the image will be shown.

Controls\ Help Button tab



Properties 'Help Button'

General

Style: <None>

Help screen <Help screen of current screen>

Help text 1: Control "Help button" can display a text

Help text 2:

Help image: stadtplan

Back color:

☑ Stretch image

Single Click

Figure 306: Window Help button properties\General

Help screen

This field can be used to select the help screen that will be opened when the operator taps on the help button. All the help screens in the project will be available for selection,

as will be the <Help screen of current screen> option. If you select the latter, the help screen used will be the one configured in the properties for the screen containing the help button.

Help text 1/ Help text 2

Used to enter or select the help texts that will be shown on the help screen when the help button is tapped on.

→ If the help text does not have enough space in the help information visualization object at runtime, a scrollbar will automatically appear on the right side of the help text field if the Display scrollbar when needed option is enabled for the help information visualization object.
If a line is too wide for the help information visualization object, the line will automatically wrap on the device at the last space in the line if the Word Wrap option is enabled for the help information visualization object. Please refer to → section " Help information", page 433

...

If the help text you want does not exist yet, clicking on this button will open the Text Editor, which can be used to define and select help texts as necessary. For detailed information on how to work with the Text Editor, please refer to → section "Working with the Text Editor ", page 640

Image

The selected image is associated with the relevant Help Button.

Selected images can be removed by clicking on the drop-down menu and selecting <None> or by selecting the field and pressing the Delete key.

Clicking on the drop-down menu arrow in the Image field will open a drop-down menu with all the images found in the Project images folder. This means that, before selecting an image here, you will need to add it to Project Tree\Images\Project images by importing it or by dragging& and dropping it there. Images will be shown in the preview window.

Copying an image to the Media Project images folder

To copy an image from Windows Explorer to Project Tree\Media tab\Project images, follow the steps below:

- Open Windows Explorer and locate the image you want
- In the project tree, open the Media\Project images folder.
- In Windows Explorer, click on the image you want and drag it specifically to the Project images folder.

After you release the left mouse button, the image will be added to the Project images folder.

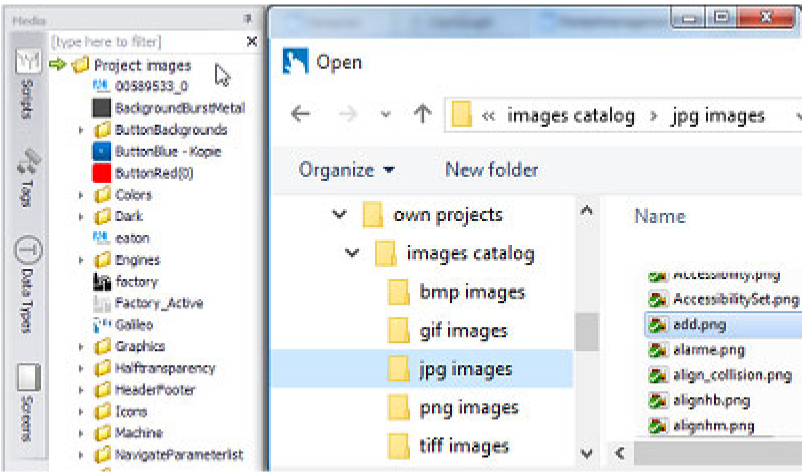


Figure 307: The add.png image is copied from Windows Explorer to the Project images folder by dragging and dropping it

Background color

You can use this field to define the background color for cases in which the selected image does not completely fill out the area of the help display object.

You can change the color if you want. The following options are available for this purpose:

- Transparent
- Standard Colors
- More Colors...

The most recently used colors will be shown (user-specific).

For detailed information on the available colors, please refer to the following section:
→ section "Color choice - Object's Color Palette", page 234

Stretch image

If the checkbox is enabled, the following will happen:

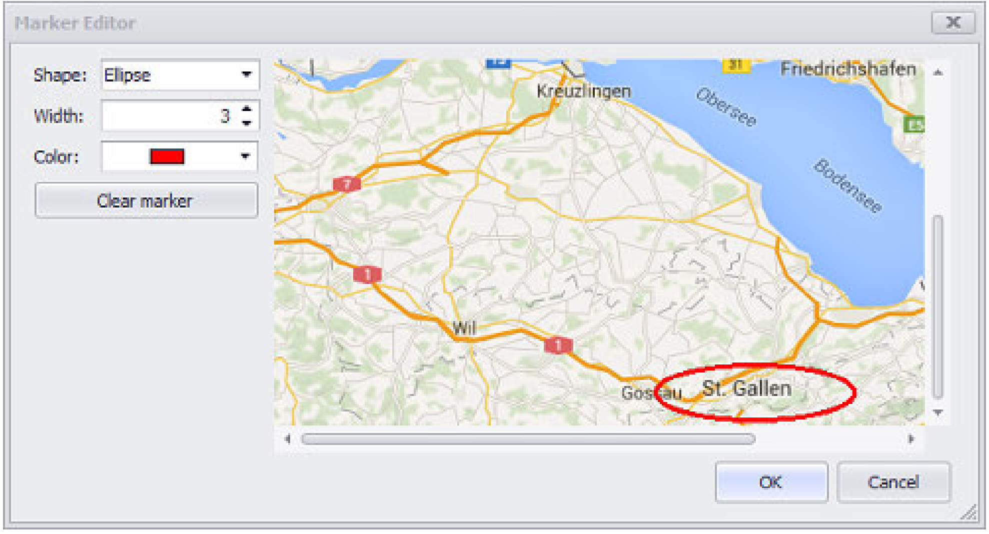
- If it is smaller than the selected display area, the image will be stretched to fit the visualization object Help Information.
- If it is larger than the selected display area, the image will be shrunk to fit the visualization object Help Information.

If the checkbox is disabled, only a section of the image will be shown in the help information visualization object if the image is larger. If it is smaller, it will be shown centered in the visualization object. In the latter case, the display area will be filled out with the color that was selected in the Background property.

Set marker

This option will only be enabled if an image is selected in the Image field. Clicking inside the field will open the Marker Editor dialog box.

Home\ Tag Help\Set marker



Marker Editor

Shape: Ellipse

Width: 3

Color: 🟥

Figure 308: Windows

Shapes

The "Ellipse" and "Rectangle" shapes are available in the drop-down menu for the type of marker.

Width

The line width for the shape is defined here, in pixels.

Color

This field can be used to select a color for the marker over the button.

You can change the color if you want. The following options are available for this purpose:

- Transparent
- Standard Colors
- More Colors...

The most recently used colors will be shown (user-specific).

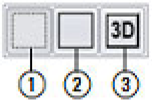
For detailed information on the available colors, please refer to the following section:
→ section "Color choice - Object's Color Palette", page 234

Clear marker

Clicking on this button will remove the marker from the image.

Frame 3D

The visualization object can be displayed with any one of three different designs.



1	no frame
2	with frame
3	3D frame

Selection mode on panel Single Click

The following modes can be adjusted for visualization objects with a keyboard:

- Single Click
  Tapping once on a list item or a button on the device screen will activate the selection.
- Delayed
  This option is ideal for implementing a safety function designed to protect against accidental actions. In order for a button on the device screen to be activated, it will first have to be pressed and held (long press) for the configured delay time. By default, this delay time will be 0.5 s, but it can be changed in *Home\Settings\Runtime tab\Touch Settings*.
- Double Click
  Tapping twice on a list item or button on the device screen will select the item / activate the button. By default, the double-click speed will be 0.5 s for a double-click, but it can be changed in*Home\Settings\Runtime tab\Touch Setting*.

3.10 Help information

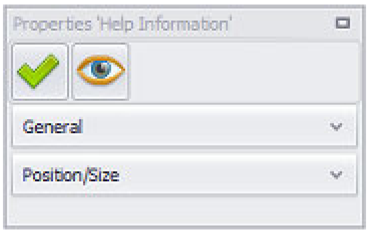
Help information visualization objects are used to define the display areas for help texts and help images on help screens. These visualization objects will not be available for any other screen type.

You can place as many help information visualization objects as you want on a help screen, but the number of contents will be limited to specific information types selected in Properties\General.

Prerequisites

In order to be able to follow the descriptions below, a help screen in *Project Tree\Screens\Help Screens* must be open. In addition, the help screen must contain a help information visualization object and the object must have been clicked on.

Controls\ tab



'Help information' properties

✓ → section "Accessibility", page 258

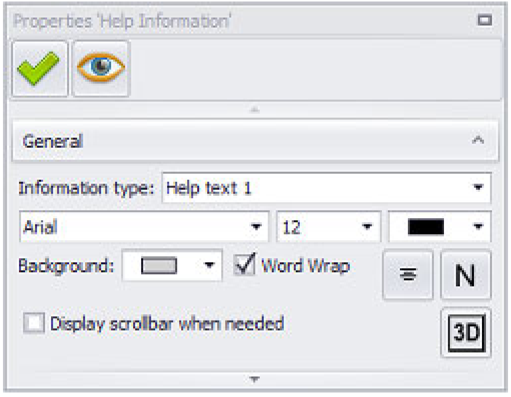
👁 → section "Visibility", page 259

→ section "General", page 433

→ section "Position/Size", page 242

General

Controls\ Help Information\General tab



Information type: Help text 1

Arial 12

Background: ☑ Word Wrap

□ Display scrollbar when needed

Figure 309: Window Help information properties

Information Type

Used to select which of the four information types should be displayed in the selected area:

- Help text 1
- Help text 2

- HTML URL
- Image
- Video
- Error text

The contents for the information types are defined either using the tag help object or in the properties for the help button. For a description of these two options, please refer to the → " Help Button", page 428 and → " Tag Help", page 935 sections.

Show scrollbar when needed

If this checkbox is enabled and the text that needs to be displayed is too long for it to be shown in the help information object, scrollbars will be shown within the object. These scrollbars can then be used to scroll through the text.

A scrollbar will be displayed only if the Help Information pane is at least 60 pixels tall.

If the checkbox is disabled, only the text that fits the display area for the help information object will be shown.

All other properties will only be available if a help text is selected as the information type.

For a description of the font, orientation, and background color, please refer to section:.

3.11 HTML viewer

Settings

→ " Text", page 492

→ " HTML viewer", page 435

The HTML Viewer adds an HTML box as a placeholder.

The Properties 'HTML Viewer' pane can be used to integrate a URL address.

Doing this makes it possible, for example, to link to documentation for your application.

Option:
The HTML Viewer visualization object is only available for devices with GALILEO OPEN, the touch panel in the XP500 and XV300 series, and in Galileo Simulator.

The HTML Viewer does not perform any functions itself. It simply supports the functionalities integrated into the corresponding HTML site based on the HTML browser (such as Internet Explorer) installed on the target device.

The HTML Viewer object can be added multiple times.
However, this may slow down the refresh rate in the application.

3.11.1 Integrating URL address contents into the HTML Viewer

The HTML Viewer visualization object is only available for devices with GALILEO OPEN, the touch panel in the XP500 and XV300 series, and in Galileo Simulator.

The following formats are supported:

- *.htm
- *.html

The HTML Viewer can be used to view formatted texts, images, and hyperlinks.

3.11.2 HTML Viewer

Prerequisites
In order to be able to follow the descriptions below, you must first have clicked on an HTML Viewer visualization object on the current screen.
You must have defined a string tag in the project tree's Tags tab.

The HTML Viewer object can be added multiple times.
However, make sure to avoid complex HTML pages in order to prevent slow initialization or sluggish page loading.

In the application,

scroll bars will be shown automatically in GALILEO OPEN and in Galileo Simulator (Win32 applications) if the HTML page is bigger than the HTML Viewer box you added.

On a touch panel (CE platforms) the page will scroll within the added HTML Viewer box.

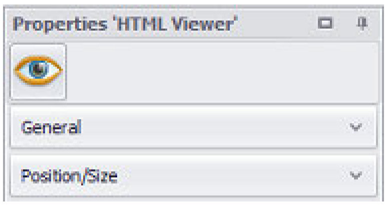
The HTML Viewer is always on the topmost screen layer.
Accordingly, make sure that other controls on the screen, if any, will remain accessible when using the viewer.

When another overlapping object or dialog box (e.g., keyboards, system messages, system icons, menus) is activated, the contents of the HTML Viewer box will be hidden.

The HTML Viewer does not require a browser cache.

Execution errors on the current HTML page will be suppressed, but critical errors and security alarms will be shown.

Controls\ tab



'HTML Viewer' properties

👁 → section "Visibility", page 259

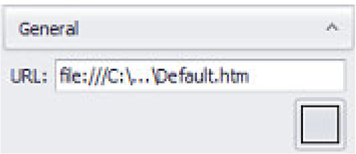
→ section "General", page 436

→ section "Position/Size", page 242

3.11.3 General

The General section can be used to configure all the properties that affect how the visualization object will be displayed, with the exception of its size and position.

Controls\ HTML Viewer tab



URL: file:///C:\...\Default.htm

Figure 310: Properties pane HTML Viewer\General

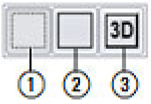
URL:

Used to enter a URL address as a parameter.
This page will later be shown in the HTML Viewer box
You can enter the address as a hyperlink with a path for local systems or as a web address
Syntax
e.g.: file:///C:\...\...\HTML-Seite.html
Or
http://www.eaton.com

□

Display style

The visualization object can be displayed with any one of three different designs.



1. No frame
2. with frame
3. 3D frame

You can use the Load URL and Load URL Var functions to switch between different URL pages.

3.11.4 HTML viewer functions

3.11.4.1 Load URL

A single screen can contain multiple HTML Viewer visualization objects at the same time. This function can be used to load other URL addresses (.htm, .html,) in the HTML Viewer visualization object.
The URL address for the HTML Viewer visualization object must be passed as a fixed string in Parameter[2].

→ Not all device types support this function

Number of the Function	5888 (0x1700)
Executable from the PLC	Yes
Function with ON/OFF state	No
Parameter 1 HTML Viewer	Used to select the HTML Viewer (if multiple instances are used on the screen) based on the top left position (e.g., X=20, Y=20)
Parameter 2 String	Fixed string used to open a different URL address, e.g., http://www.eaton.com or access to the <Custom> directory
Function available in standard screens	Yes
Function available with keyboards	No
Function available in script	Yes
Function available in application entry script	Yes

Example

```
HTMLViewer.LoadUrl(Start.HVO, "www.eaton.com");
```

3.11.4.2 Load URL Var

A single screen can contain multiple HTML Viewer visualization objects at the same time.
This function can be used to load other URL addresses (.htm, .html,) in the HTML Viewer visualization object.

The URL address for the HTML Viewer visualization object must be passed as a tag in Parameter[2].

→ Not all device types support this function

Number of the Function	5889 (0x1701)
Executable from the PLC	No
Function with ON/OFF state	No
Parameter 1 HTML Viewer	Used to select the HTML Viewer (if multiple instances are used on the screen) based on the top left position (e.g., X=20, Y=20)
Parameter 2 String tag	Used to select a string tag that contains an editable URL address.
Function available in standard screens	Yes
Function available with keyboards	No
Function available in script	Yes
Function available in application entry script	Yes

See also

→ section "Load URL", page 437

3.12 Image

Used to insert an image box as a placeholder. An image in .png format or an image sequence in .gif format can be inserted into this image box. To do this, the image or image sequence must first be loaded into the project via *Project Tree\Media*.

Animated image sequence

→ Animated image sequences are supported only by XV300 and XP500 devices, as well as by GALILEO Open.

The project image must be in one of the following formats: *.png, *.jpg/*.jpeg , *.tiff/*.tif, *.gif , *.svg or *.bmp,

image sequences with the *.gif format.

The project video must be in one of the following formats: *.mp4, *.mov or *.avi.

Icon library images are in .svg format.

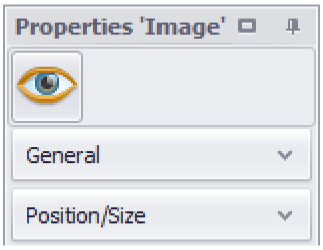
→ GALILEO is only able to import images (graphs, icons, logos, etc.), i.e., these images cannot be edited in the program.
If you want to make modifications to images, you will need to do so externally with an appropriate graphics editor that can be used to directly open the images by using the context menu in *Project\Media tree*.
The graphics editor that will be used by default can be selected in the → section "Window Application Options", page 135.

The Properties 'Image' pane can be used to fill the box and format and position the image.

Animated image sequences consist of multiple images in .gif format that are grouped and saved in what is referred to as a "multiple-frame file" in .gif format. When this approach is used, the individual files are displayed one after the other, resulting in an animation. This animation can be started, paused, and stopped while it is running. In order to use these animations for your project, the files in .gif format need to be converted to .png format. In addition, you will need to use image editing software in order to create the multiple-frame files you want to use in advance.

For more information, please refer to the following sections: → " Media", page 68 and → "Step 5: Adding an image", page 174

Controls\ tab



'Image' properties

→ section "Visibility", page 259

→ "General", page 440

→ section "Position/Size", page 242

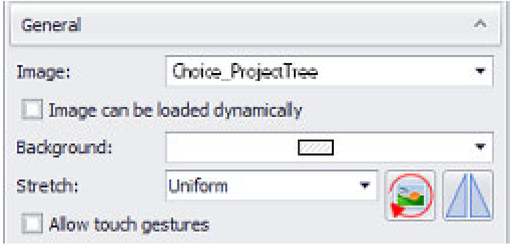
3.12.1 General

Defines the object's appearance on the screen

The specific properties will depend on whether you are using a single-frame file (an image) or a multiple-frame file (an image sequence).

Single-frame file - simple image

Controls\ image



General

Image: Choice_ProjectTree

□ Image can be loaded dynamically

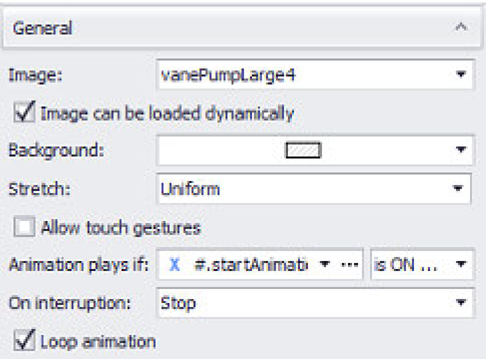
Background: ▨

Stretch: Uniform

□ Allow touch gestures

Figure 311: Window Properties Image

Multiple-frame file - image sequence



General

Image: vanePumpLarge4

☑ Image can be loaded dynamically

Background: ▨

Stretch: Uniform

□ Allow touch gestures

Animation plays if: X #.startAnimati ... is ON ...

On interruption: Stop

☑ Loop animation

Figure 312: Properties 'Image' pane for image sequences

Image:

Name of the image or image sequence in the selected image box or selection with the images available for the project in *Project Tree\Media*.
For more information, please refer to the following sections: → " Media", page 68 and → "Step 5: Adding an image", page 174

Image can be loaded dynamically

This option is only available for single-frame files (not for animations).
If the checkbox is enabled, you will be able to use the 💡 quick menu functions for the visualization object. The specific functions available in the quick menu will be Load , Load Var, Select, and Select Var.

Background:

A transparent background will be selected for the image box by default. You can select a color instead if you want.
For detailed information on the available colors, please refer to the following section: → section "Color choice - Object's Color Palette", page 234

Stretch:

The following options are available:

- None

The image will be aligned with the upper left corner of the area occupied by the visualization object. It will be inserted with its original size, i.e., if the image is larger than the area occupied by the visualization object, only part of the image will be displayed.

- Uniform
  The image will be centered in the area occupied by the visualization object. The aspect ratio will be maintained. In addition, the image will be resized to fit the area This means that if the image's aspect ratio does not match the aspect ratio of the visualization object's area, the selected background color will fill out the rest.
- Fill
  The image will be centered in the area occupied by the visualization object. In addition, the image will be resized to fit the area (the image's aspect ratio will not be maintained).

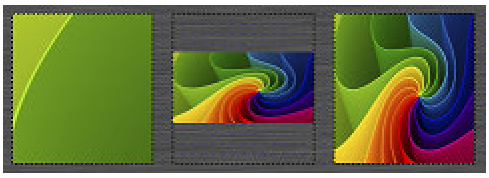


Figure 313: Stretch = None, Uniform and Fill

Image Rotation

The buttons shown above can be used to rotate the image 90°, 180°, or 270° as needed.

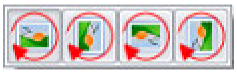
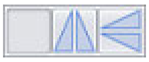


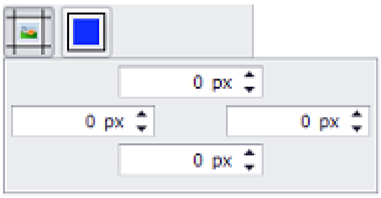
Image Flipping

The buttons shown above can be used to flip the image.



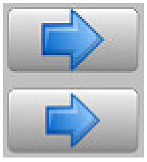
Distance from edge

Can be used to specify the distance from the visualization object edges that should be maintained. This can be used, for example, to use an image with Uniform stretching without it going all the way to the edge of the visualization object.



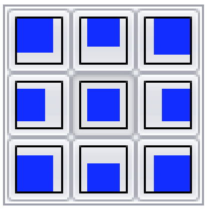
0 px
0 px
0 px
0 px

Example showing a button with an icon with Uniform stretching. The button on top does not have a distance from the edges, while the button on the bottom has a distance of 5 px from the top and bottom edges:



Alignment

Used to set the icon's alignment within the visualization object.



Example showing a button in which the icon is top-left-aligned and has a distance of 5 px from the edge:

Allow touch gestures

These touch gestures can only be used with PTC devices such as the XV300 and XP500.
This function is also supported by GALILEO Open applications.

- Drag
  Large lists, such as parameter lists, alarm lists, and recipes, can be completely scrolled through with one finger.
- Panning
  If an image is too big for the corresponding display, it can be moved around by pressing on it with a finger and dragging it around. This can be done with graphs, for example, in which case it will only be possible to move the graph within the defined display area.
- Pinch close/Pinch open
  You can zoom into images and graphs or into curves (e.g., the X-axis of graphs) in order to view contents in greater detail.
  This gesture requires the use of more than one finger at the same time – for instance, touching the screen with your thumb and index finger and spreading them apart will zoom in.
  Please note that the object element (e.g., image) cannot be made smaller than the defined display area.
- Flick
  If there are multiple user controls next to or on top of each other (e.g., in a container) and the display object is too small, they can be moved around with a horizontal or vertical flick gesture on the screen, within the defined display area.
- Double tap
  This will restore the object's (e.g., image) original appearance or show the image in twice the resolution.

The touch function can be configured in the properties for the object. If the corresponding checkbox is ☑ enabled, device operators will be able to use touch gestures for control at runtime.
This option is only available for single-frame files (not for animations).

Only available for dynamically loaded animations

Animation plays if

If you enable this option, the animation will be started when the specified condition is met.

On interruption
You can use this drop-down menu to specify whether the animation should be stopped when the condition is met (<Stop>) or whether it should just be paused (<Pause>).
The animation will remain in the <Stop> or <Pause> state until the condition applies again, at which point the animation will become active again and start or resume.

Loop animation
☑ Turn on Enabling the checkbox
The image sequence will be looped indefinitely.
If the option is not □, it will only be run once.

3.12.2 Image functions

3.12.2.1 Load Image

This function loads an image sequence and shows it on the screen using the image visualization object.
The image name must be specified in Parameter2.

The following image file formats are supported: PNG, JPG, and BMP. The bit depth of JPG and BMP images must be at least 24 bits.

For a description of the way images are handled, please refer to: → section "Step 5: Adding an image", page 174.

The image name can be specified with or without a filename extension, e.g., <Sample.png> and <Sample>.

→ If the function is run using a script, the referenced Image visualization object must be found on the screen currently being displayed. Otherwise, the function will not be run.

Number of the Function	5376 (1500h)
Executable from the PLC	Yes
Function with ON/OFF state	No
Parameter 1	Name or coordinates of the Image visualization object
Parameter 2	Name with which the image is saved. Name example: `"Sample"` If you want to use a specific storage location, you can include the path in the name. Name example: `"C:/temp/Sample"`
Function available in standard screens	Yes
Function available with keyboards	No
Function available in script	Yes
Function available in application entry script	Yes

3.12.2.2 Load Image Var

This function loads an image and shows it on the screen using the image visualization object.
The image name must be specified in Parameter2 with a tag of data type STRING.

The following image file formats are supported: PNG, JPG, and BMP. The bit depth of JPG and BMP images must be at least 24 bits.

For a description of the way images are handled, please refer to: → section "Step 5: Adding an image", page 174.

The image name can be specified with or without a filename extension, e.g., <Sample.png> and <Sample>.

→ If the function is run using a script, the referenced Image visualization object must be found on the screen currently being displayed. Otherwise, the function will not be run.

Number of the Function	5377(1501h)
Executable from the PLC	No
Function with ON/OFF state	No
Parameter 1	Name or coordinates of the Image visualization object
Parameter 2	Tag of data type STRING The tag contains the name with which the image is saved. Tag content example: `"Sample"` If you want to use a specific storage location, you can include the path in the name. Tag content example: `"C:/temp/Sample"`
Function available in standard screens	Yes
Function available with keyboards	No
Function available in script	Yes
Function available in application entry script	Yes

3.12.2.3 Select Image

This function loads an image and shows it on the screen using the image visualization object.

A system keyboard will appear so that the device operator can look for the image in a directory.

The directory must be specified in Parameter2.

For a description of the way images are handled, please refer to: → section "Step 5: Adding an image", page 174.

→ If the function is run using a script, the referenced Image visualization object must be found on the screen currently being displayed. Otherwise, the function will not be run.

Number of the Function	5378(1502h)
Executable from the PLC	Yes
Function with ON/OFF state	No
Parameter 1	Name or coordinates of the Image visualization object
Parameter 2	The path where the image is saved. Example: `"C:/temp/"`

Function available in standard screens	Yes
Function available with keyboards	No
Function available in script	No
Function available in application entry script	Yes

3.12.2.4 Select Image Var

This function loads an image and shows it on the screen using the image visualization object.

A system keyboard will appear so that the device operator can look for the image in a directory.

The directory must be specified in Parameter2 with a tag of data type STRING.

Only images in PNG format will be shown.

→ If the function is run using a script, the referenced Image visualization object must be found on the screen currently being displayed. Otherwise, the function will not be run.

Number of the Function	5379(1503h)
Executable from the PLC	No
Function with ON/OFF state	No
Parameter 1	Name or coordinates of the visualization object Image
Parameter 2	Tag of data type STRING; The tag should contain the path where the image is saved. Example: `"C:/temp/"`
Function available in standard screens	Yes
Function available with keyboards	No
Function available in script	Yes
Function available in application entry script	Yes

3.13 ≡ Menu button

This visualization object can be used to display a specific user control.

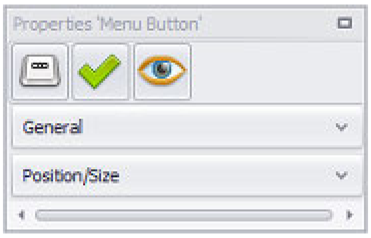
Depending on the selected menu alignment, the selected user control will slide into the standard screen from the top, bottom, left, or right edge.

Depending on the selected menu behavior, the user control will either slide in so that it overlaps the standard screen or the standard screen will be pushed to the side as far as necessary for the user control to be displayed.

You can use the Menu drop-down menu in the properties for the menu button in order to specify which user control should be displayed. The menu alignment and the menu behavior can also be set in these properties.

By default, this visualization object is an input object with a touch zone.

Controls\ ≡ tab



'Menu button' properties

▭ → " States", page 448

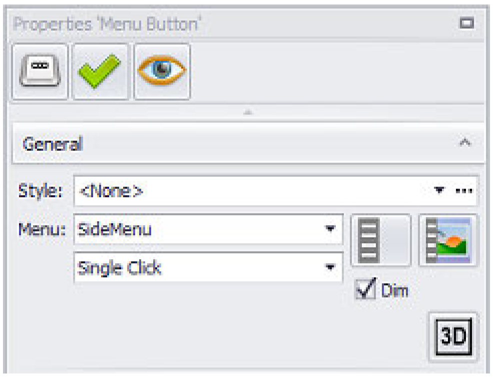
✓

👁 Visibility

→ "General", page 446

3.13.1 General

Controls\ ≡ Menu Button tab


Properties 'Menu Button'

General

Style: <None>

Menu: SideMenu

Single Click

☑ Dim

3D

Figure 314: Properties 'Menu Button' pane

Style

Settings that affect the appearance of visualization objects can be bundled and stored as a style. After this, it will be possible to use the style's name to use these settings at any time and apply them to new visualization objects of the same type throughout the entire project. In addition, when changes are made to a style, the changes will automatically be applied to all visualization objects using the style, ensuring that all visualization objects using the same style will have a consistent appearance. For detailed information on how to create a style, please refer to → section "Style management", page 458.

Menu

Clicking on the drop-down menu arrow will open a list of available user controls. The selected object will slide in from the edge of the screen when the button is tapped.

Menu Alignment

- Left
- Right
- Up
- Bottom

Menu Behavior

- Push Away
  When this option is selected, the user control will push the standard screen to the side from the direction specified with the menu alignment.
- Overlapping
  When this option is selected, the user control will slide over the standard screen from the direction specified with the menu alignment.

Dim

If this checkbox is enabled, everything except the selected user control will be dimmed when the menu button is tapped.

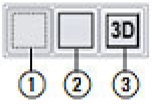
Selection mode on panel Single Click

The following modes can be adjusted for visualization objects with a keyboard:

- Single Click
  Tapping once on a list item or a button on the device screen will activate the selection.
- Delayed
  This option is ideal for implementing a safety function designed to protect against accidental actions. In order for a button on the device screen to be activated, it will first have to be pressed and held (long press) for the configured delay time. By default, this delay time will be 0.5 s, but it can be changed in *Home\Settings\Runtime tab\Touch Settings*.
- Double Click
  Tapping twice on a list item or button on the device screen will select the item / activate the button. By default, the double-click speed will be 0.5 s for a double-click, but it can be changed in*Home\Settings\Runtime tab\Touch Setting*.

Frame 3D

The visualization object can be displayed with any one of three different designs.



① no frame

② with frame

③ 3D frame

3.13.2 States

Menu button visualization objects have two states: *Inactive* and *Active*.

The basic configuration parameters for the visualization object's foreground and background are described in detail in the following section: Defining text series

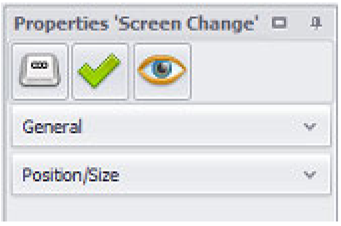
3.14 Screen change properties

In order to make it possible to switch between standard screens, a screen change visualization object needs to be added to them. When a device operator taps on the corresponding button on the device screen at runtime, the system will switch between screens (screen changes can also be triggered directly by the corresponding PLC). There is a small screen change example that is described in the → section "Step 8: Adding screen changes", page 189 section.

Prerequisites

To be able to follow the descriptions below on the program, click on a screen in *Project Tree\Screens* in order to open the Properties pane.

Controls\ tab



'Screen change' properties

⌨ → "States", page 451

✓

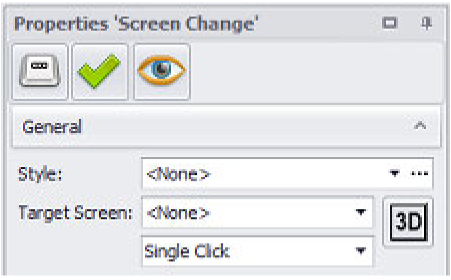
👁

→ section "General", page 449

The ✓ accessibility button will only be shown in the Properties pane if a tag and a keyboard have been assigned to the visualization object, as it would not make sense to define accessibility conditions otherwise.

3.14.1 General

Controls\ Screen Change tab



Style: <None>
Target Screen: <None>
Single Click

Figure 315: Properties pane for the "Start" standard screen\General

Style

Settings that affect the appearance of visualization objects can be bundled and stored as a style. After this, it will be possible to use the style's name to use these settings at any time and apply them to new visualization objects of the same type throughout the entire project. In addition, when changes are made to a style, the changes will automatically be applied to all visualization objects using the style, ensuring that all visualization objects using the same style will have a consistent appearance. For detailed information on how to create a style, please refer to → section "Style management", page 458.

Target Screen

This property is used to specify the screen to which the device should switch when the button is pressed at runtime. Please note that only standard screens can be set as a target screen.

Animating screen changes

Animated touch panelGALILEO OPENscreen changes can be configured for XV300 and XP500 s, as well as for .

"Animation" means that the target screen will "fly in" or that the current screen will "fly out," revealing the target screen.

<No animation> will be selected by default. When this option is selected, the device will switch straight to the target screen without an animated transition.

These options are grouped into rows.

The following settings are available:

1. No animation – The device will switch straight to the target screen.
2. Cover – The current screen will remain in place and the target screen will fly in
   - Cover from left – The target screen will fly in from the left
   - Cover from right – The target screen will fly in from the right
   - Cover from top – The target screen will fly in from above
   - Cover from bottom – The target screen will fly in from below
3. Uncover – The target screen will remain in place and the current screen will fly out
   - Uncover to left – The current screen will fly out towards the left
   - Uncover to right – The current screen will fly out towards the right
   - Uncover to top – The current screen will fly out upwards
   - Uncover to bottom – The current screen will fly out downwards
4. Push – The target screen will move in, "pushing" the current screen out
   - Push left – The target screen will move in, pushing the current screen out to the left
   - Push right – The target screen will move in, pushing the current screen out to the right
   - Push up – The target screen will move in, pushing the current screen up and out
   - Push down – The target screen will move in, pushing the current screen down and out

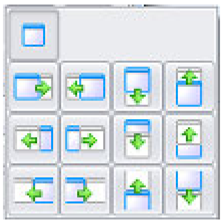
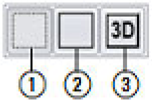


Figure 316: Available animation options for the screen change object

In certain cases, such as <Cover from left>, <Uncover to right>, and <Push right>, the difference between the options may only be noticeable if there are multiple visualization objects on the screens.

Frame 3D

The visualization object can be displayed with any one of three different designs.



① no frame

② with frame

③ 3D frame

Selection mode on panel Single Click

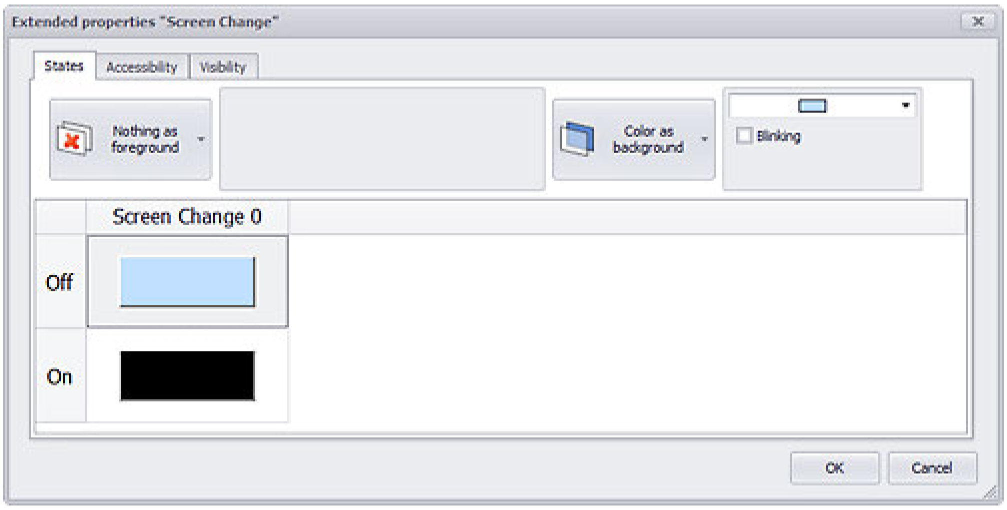
The following modes can be adjusted for visualization objects with a keyboard:

- Single Click
  Tapping once on a list item or a button on the device screen will activate the selection.
- Delayed
  This option is ideal for implementing a safety function designed to protect against accidental actions. In order for a button on the device screen to be activated, it will first have to be pressed and held (long press) for the configured delay time. By default, this delay time will be 0.5 s, but it can be changed in *Home\Settings\Runtime tab\Touch Settings*.
- Double Click
  Tapping twice on a list item or button on the device screen will select the item / activate the button. By default, the double-click speed will be 0.5 s for a double-click, but it can be changed in*Home\Settings\Runtime tab\Touch Setting*.

3.14.2 States

The *States* tab can be used to define what the visualization object should look like when displayed. Screen change visualization objects have two definable states: ON and OFF. A background color, text color, font, font size, and text format can be defined for these states.

Controls\ Screen Change



Extended properties "Screen Change"

	Screen Change 0
Off	🟦
On	■

Figure 317: States of a Screen Change

The basic configuration parameters for the visualization object's foreground and background are described in detail in the following section: Defining text series

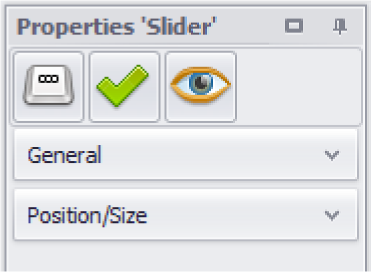
3.15 Slider

Slider visualization objects can be used to quickly adjust tag values at the cost of less precision. For example, they can be used to dim lights. Sliders can be defined either with a horizontal or vertical sliding direction.

→ On IR-touch devices, the minimum shift position of the slider is fixed by the resolution of the light matrix.

If this visualization object is used multiple times to display a large number of system components with the same parameter sets, it is recommended to use parameters. Using parameters will optimize your work when it comes to assigning tags to the visualization object.
For detailed information on how to work with parameters, please refer to → section "Using user controls to optimize work", page 279.

Controls\ tab



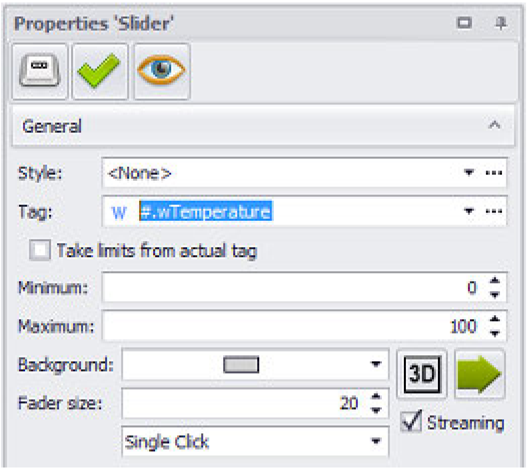
'Slider' properties

⌨	→ " States", page 457
✓	
👁	Visibility
	→ "General", page 454

The ✓ accessibility button will only be shown in the Properties pane if a tag and a keyboard have been assigned to the visualization object, as it would not make sense to define accessibility conditions otherwise.

3.15.1 General

Controls\ Slider



Properties 'Slider'

General

Style: <None>

Tag: W #.wTemperature

□ Take limits from actual tag

Minimum: 0

Maximum: 100

Background:

Fader size: 20

3D

☑ Streaming

Single Click

Figure 318: Properties 'Slider' pane, General section

Style

Settings that affect the appearance of visualization objects can be bundled and stored as a style. After this, it will be possible to use the style's name to use these settings at any time and apply them to new visualization objects of the same type throughout the entire project. In addition, when changes are made to a style, the changes will automatically be applied to all visualization objects using the style, ensuring that all visualization objects using the same style will have a consistent appearance. For detailed information on how to create a style, please refer to → section "Style management", page 458.

Variable

The visualization object will show the value of the selected tag on the device screen at runtime. Clicking on the drop-down menu arrow will open a list of available tags (BYTE, WORD, DWORD, FLOAT). Please note that in order for the tag you want to be shown in this list, the tag must first be imported or created in *Project Tree\Tags*.

A parameter can also be specified instead of a tag. In this case, the Tag field will turn into a Type field. In order to be able to work with parameters, the visualization object must be placed on a subscreen. For detailed information on how to work with parameters, please refer to → "Using input parameters", page 279.

Values do not necessarily have to be displayed or entered the way the PLC needs them or the way they are stored in the tags. The Value Conversion Manager can be used to convert tag values and assign a variety of units to them before they are shown by visualization objects. The way value conversion works is described in greater detail in a different section – please refer to → "Conversions", page 921.

Minimum

Minimum is the lowest value of the range for the slider's fader knob. This minimum is not the same as the minimum for the tag value, and accordingly can have a different value.

The minimum can be set as a constant value or as a dynamic tag-based value. To set it as a constant value, simply enter the number you want into the corresponding field. To set it as a dynamic limit instead, enter the tag you want into the Minimum field. For detailed information on how to set the minimum as a dynamic value, please refer to → "Numeric tag limits", page 563

Maximum

Maximum is the highest value of the range for the slider's fader knob. This maximum is not the same as the maximum for the tag value, and accordingly can have a different value.

The maximum can be set as a constant value or as a dynamic tag-based value. To set them as constant values, simply enter the numbers you want into the corresponding fields. To set it as a dynamic limit instead, enter the tag you want into the field.

Defining dynamic values with tags

Instead of entering a fixed value, you can link the field to a tag, so that the value can be changed dynamically. To link to a tag, follow the steps below:

- ▶ Place the cursor inside the field.
- ▶ Right-click to open the context menu and select the Use Tag option.

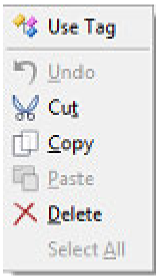


Figure 319: Context menu with Use Tag option

The field will go from looking like this: 0 ▴▾ to looking like this: ▾ ….

- ▶ Click on the drop-down menu arrow and select the tag that should represent the value.
- ▶ Clicking on the … icon will open the Properties <tag name> dialog box. Make sure that the settings for the tag are correct, e.g, Default, Maximum. If necessary, you can also use tags here in order to define limits. During compilation, the system will check whether there are recursive tag references and, if so, will output an error message.

→ All value tags are always declared in decimal format regardless of the selected display format (decimal, hex, binary, octal, BCD)!

→ All values are PLC values. They will be transmitted to the PLC the way they are, i.e., without being converted.

Background color

If this option is deactivated, only the Streaming of the Object is drawn. Otherwise, the entire background area of the slider is filled with the background color.

You can change the color if you want. The following options are available for this purpose:

- Transparent
- Standard Colors
- More Colors...

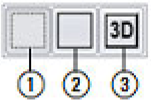
The most recently used colors will be shown (user-specific).

For detailed information on the available colors, please refer to the following section: → section "Color choice - Object's Color Palette", page 234

→ If the background is set to "Transparent," it will not be allowed to place other visualization objects in this area!

Frame 3D

The visualization object can be displayed with any one of three different designs.



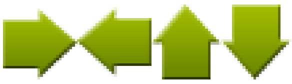
① no frame

② with frame

③ 3D frame

Direction ➡

The arrow buttons can be used to define the direction in which the slider should move.



Fader size

The width of the fader knob in the slider object is defined in pixels.

Streaming

If this checkbox is enabled, the current position of the fader knob will be continuously saved in the specified tag when the knob is moved. This can come in handy when an action should be carried out while the slider's fader knob is being moved, e.g., adjusting a light. If the checkbox is disabled, the position will be saved only after the fader knob is released.

Selection mode on device screen Single Click

The following modes can be adjusted for visualization objects with a keyboard:

- Single Click
  Tapping once on a list item or a button on the device screen will activate the selection.
- Delayed
  This option is ideal for implementing a safety function designed to protect against accidental actions. In order for a button on the device screen to be activated, it will first have to be pressed and held (long press) for the configured delay time. By default, this delay time will be 0.5 s, but it can be changed in *Home\Settings\Runtime tab\Touch Settings*.

3.15.2 States

Slider visualization objects only have one definable state,

The basic configuration parameters for the visualization object's foreground and background are described in detail in the following section: Defining text series

3.16 Style management

Styles eliminate a lot of the configuration work that usually goes into designing a user interface. Settings that affect the appearance of visualization objects can be bundled and stored as a style. After this, it will be possible to use the style's name to use these settings at any time and apply them to new visualization objects of the same type throughout the entire project.

Starting with GALILEO 10.1, pre-defined style sets are included for the following devices:

- XV-300
- XP-500
- GALILEO OPEN

The button Import from style set...:button can be used to access and use these style sets.

→ Please note that this option is only available for certain panels.
Described in section:
→ section " Select panel", page 221
The option will not be available for XV-100 and XV-400 panels so as to ensure that complex styles will not have an impact on the panels' performance.

A style set contains styles for multiple visualization objects of different types.

In addition, when changes are made to a style, the changes will automatically be applied to all visualization objects using the style ensuring that all visualization objects using the same style will have a consistent appearance.

Elements encompassed by styles

The properties that can be saved in a style will depend on the specific type of visualization object to which the style applies. For example, the style for all types of visualization objects includes a background color, font, and font size → section "Elements encompassed by styles", page 1137

Certain visualization objects of different types can use the same style if they have the same display properties. These objects are categorized accordingly in the Style Manager.

Table 4: Style Manager scope

Style	Can be used on the following types of visualization objects
Bargraph	Bargraph
Static	Text, Line, Rectangle, Ellipse, Polygon
Display/Entry	PLC Text
Button based	Button, Screen Change, Help Button, Function Key
Slider	Slider
Text based	Text, Date/Time Display
No Style Available	Image, Switch, Flag Display, Gauge, Graph Block, Parameter List, Recipe, Alarm List, Help Information, Container

Style Limitations

When you create a style, you will be able to apply it to any visualization object belonging to the corresponding type. This also means that styles, as a general rule, cannot be applied to more than one visualization object type. For example, you can assign the style for a button to another button, but not to a slider. The reason for this is that different visualization object types have different properties, meaning that styles do not always carry over to a different visualization object type. For example, almost all visualization objects feature the font and font size properties, but the fader size property is only found in slider controls; please refer to the → section "Elements encompassed by styles", page 1137 section

Styles can be created both in the Properties '<visualization object>' pane for a visualization object and in the Style Manager. To open the Style Manager, click on *Home\Style Manager*.

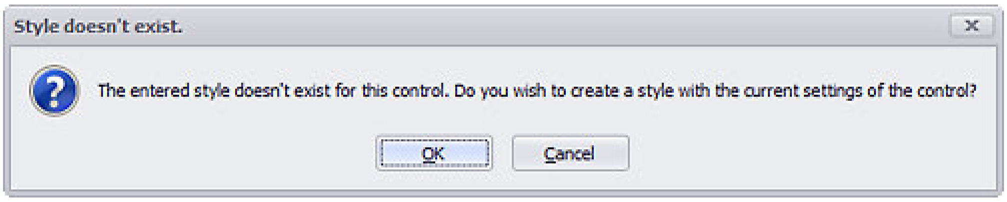
Prerequisites
You must have already created a visualization object, such as a button.

1. → "Creating a style using the Properties '<visualization object>' pane", page 459
2. → "Create Styles with the Style Manager", page 461
3. → "Applying a style", page 464
4. → "Deleting a style", page 463
5. → "Duplicating a style", page 463
6. → "Modifying a style", page 465
7. → "Style sets", page 467

3.16.1 Creating a style using the Properties '<visualization object>' pane

- ▶ Click on a visualization object, such as a button.
- ▶ Position the object
  Move the cursor to the screen and draw the area for the visualization object.
- ▶ Configure all the properties for the button the way you want your style to be.
- ▶ In the Properties pane, enter a name for the new style into the Style field, e.g., <Buttons_1>.
- ▶ Press the Enter key to confirm.
- ▶ The The entered style doesn't exist … prompt will appear. Click on OK.

Controls\Button\Properties\Style/prompt



Style doesn't exist.

The entered style doesn't exist for this control. Do you wish to create a style with the current settings of the control?

You have now created a new style called Buttons_1 .

Example showing how to create a style

Say that you have several machines in your production line. Each machine has two buttons – one that lowers a gripper and another that allows the assembly line conveyor to keep running.

Now say you want the buttons to have the same appearance on every machine.

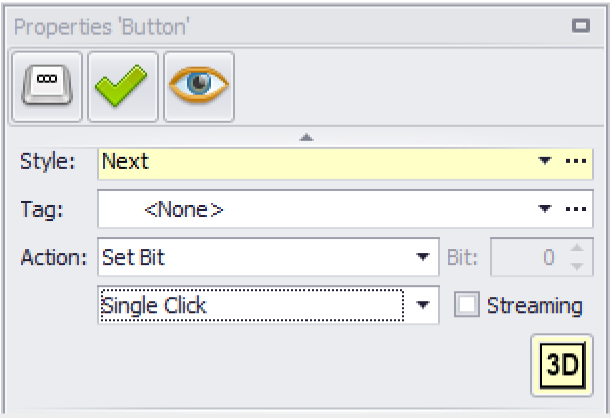
To do this, you could define two styles: Continue< and ><Lower>. Let's assume that you want buttons A1 and B1 to be dark green when off.

Moreover, let's assume that you want buttons A2 and B2 to be red when off.

Create two styles by following step 1, as outlined above, to create a style for buttons A1 and A2.

- ▶ Click on button A1.
- ▶ Set a dark green background color for the button when the button is in the off state. For information on how to change the state color, please refer to → section " Button", page 365
- ▶ For the new style, enter Continue into the <Style> field in the Properties pane.
- ▶ Confirm by pressing the Enter key.

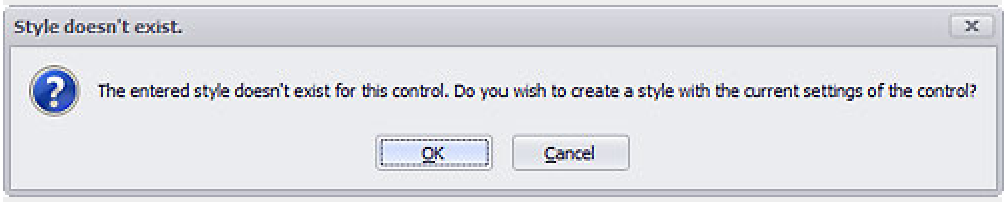
Controls\Button\Properties/Style



Style: Next
Tag: <None>
Action: Set Bit
Bit: 0
Single Click
□ Streaming

Figure 320: Properties 'Button' pane with Style field when creating a style

- ▶ Click on OK in the prompt that appears.

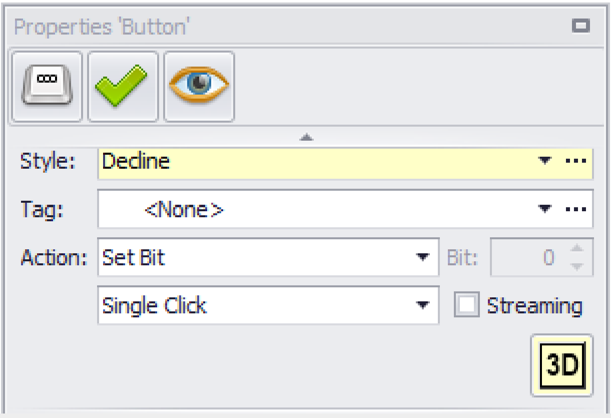

Style doesn't exist.

The entered style doesn't exist for this control. Do you wish to create a style with the current settings of the control?

Figure 321: Prompt when creating a style

- Click on button A2.
- Select the 3D option for the button's frame and set a red background color for the button when the button is in the off state. For information on how to change the state color, please refer to → section " Button", page 365
- For the new style, enter <Lower> into the Style field in the 'Button' Properties pane.
- Confirm by pressing the Enter key.

Controls\Button\Properties\Style



Properties 'Button'

Style: Decline

Tag: <None>

Action: Set Bit

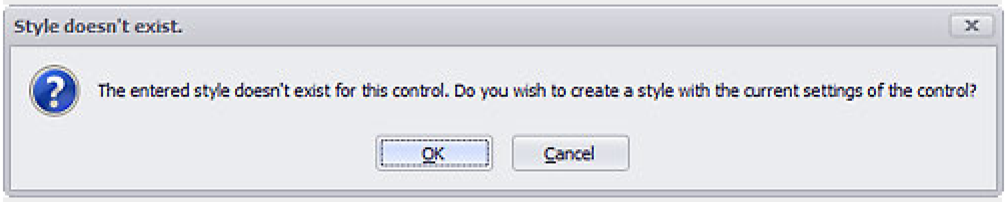
Bit: 0

Single Click

□ Streaming

Figure 322: Properties 'Button' pane with Style field when creating a style

- Click on OK in the prompt that appears.



Style doesn't exist.

The entered style doesn't exist for this control. Do you wish to create a style with the current settings of the control?

Figure 323: Prompt when creating a style

You have now created two styles: Continue and Lower.

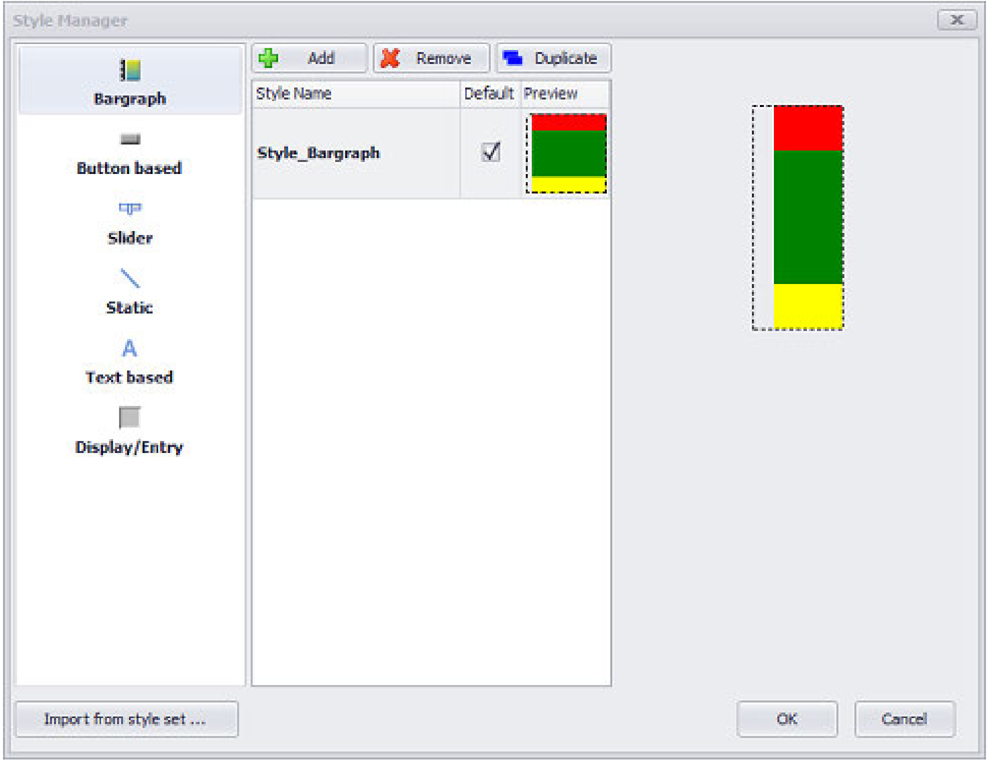
3.16.2 Create Styles with the Style Manager

Existing styles are shown in the Style Manager.

- Open the Home tab and click on Style Manager.
- Select the visualization object for which you want to create a style, e.g., Bargraph.
- Click on Add.
- If you want the style to be applied to every visualization object of the corresponding type by default, ☑ the checkbox under Default.

- ▶ To assign a name of your choice, click on the name. Spaces and special characters are not allowed.

Home\Style Manager tab



Style Name	Default	Preview
Style_Bargraph	☑	

Figure 324: Creating a style and setting it as the default style

- ▶ Click on OK.
- ▶ Open a standard screen.
- ▶ Create a visualization object of the type for which you just created a style.
- ▶ Now modify the style you created according to your needs. For detailed information on how to modify a style's properties, please refer to the → section "Modifying a style", page 465 section.
- ▶ When you make a change, the style will be reset to <None> in the *Properties '<visualization object'>, \Style field*.
- ▶ Click on the arrow for the >Style field in the the Properties 'visualization object' pane to open the corresponding drop-down menu.
- ▶ Select the following option: <Apply to 'name of style'>.
- ▶ Click on Caution! with OK.

The Style Manager provides a preview for the ON/OFF states for all Button based and Value Display/Entry visualization objects. In addition, it also provides a preview for the Upper Limit, <In Range>, and <Lower Limit> states for <Value Display/Entry> visualization objects.

Default

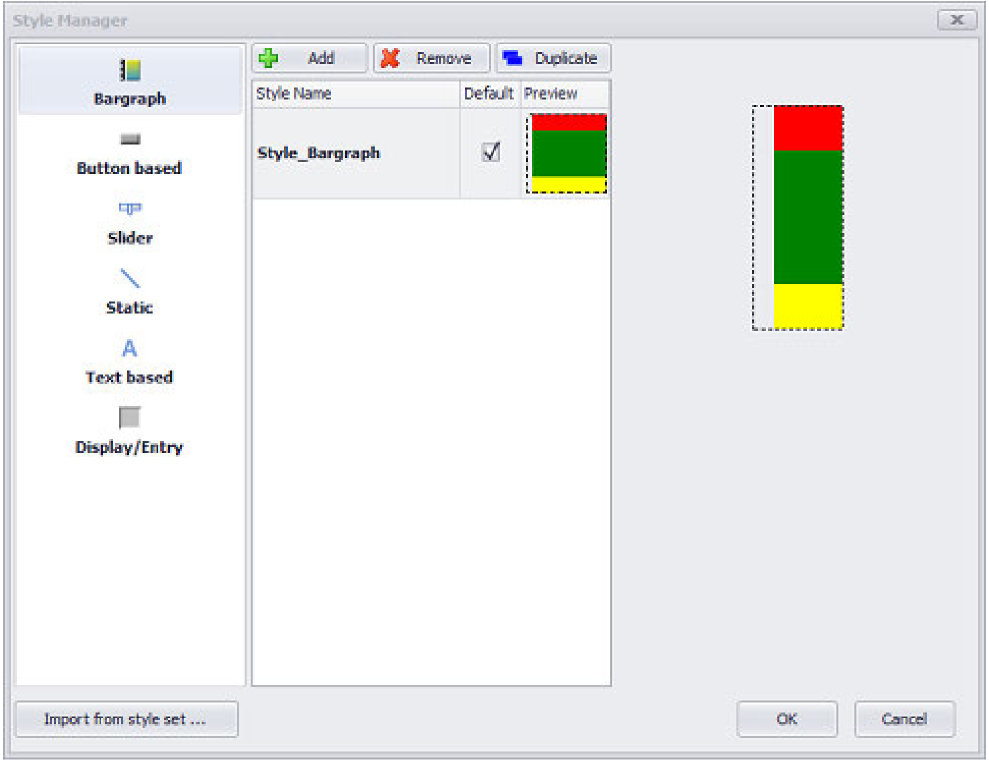
If a style is set as the <default> style, every new visualization object will be created with this style. When a style set is selected for a new project, every visualization object for the style set will already be enabled. However, when a style set is imported after a project is created, it will be necessary to enable a style for each visualization object listed in the Style Manager by enabling the <Default> checkbox.

3.16.3 Deleting a style

To delete a style, follow the steps below:

- Open the Home tab and click on Style Manager.
- Select the type of visualization object for which you want to delete a Style e.g., <Bargraph>.
- Select the Style that you want to delete e.g., <Style_Bargraph>.
- Click Remove.
- Click on OK.

Home\Style Manager tab



Style Name	Default	Preview
Style_Bargraph	☑	

Figure 325: Deleting a style

3.16.4 Duplicating a style

To duplicate a style, follow the steps below:

- Open the Home tab and click on Style Manager 🟥.
- Select the type of visualization object for which you want to duplicate a style, e.g., Bargraph.
- Select the style that you want to duplicate.
- Click Duplicate.

3.16.5 Applying a style

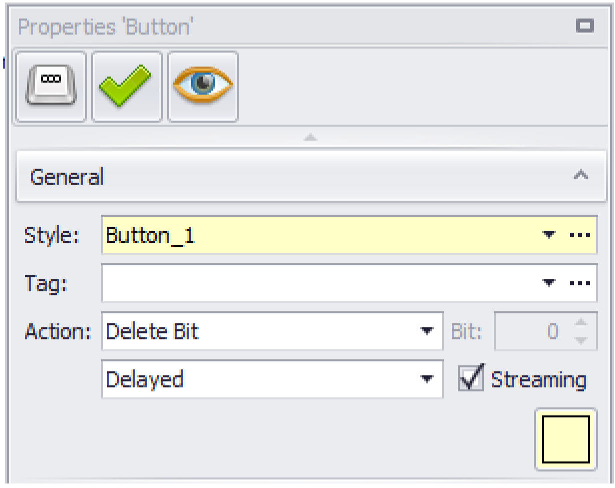
To assign a style to a visualization object, follow the steps below:

- Click on the visualization object to which you want to apply the style.
- Use the drop-down menu for the <Style> field in the Properties 'visualization object' pane to select the style you want.

The visualization object's appearance on the screen will change according to the selected style.

→ The Style field will be highlighted yellow in order to indicate that a style is being used.

Controls\Button\Properties/Style


Properties 'Button'

General

Style: Button_1

Tag:

Action: Delete Bit Bit: 0

Delayed ☑ Streaming

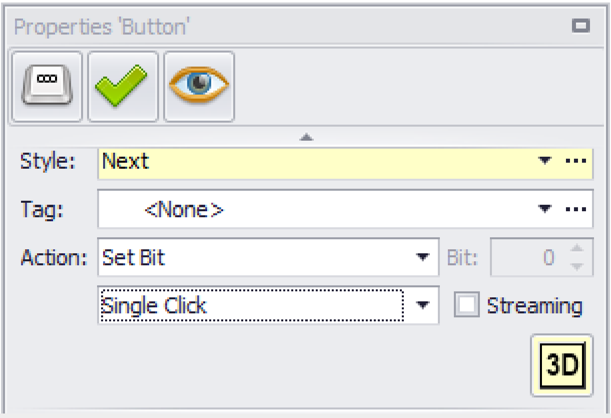
Figure 326: Properties 'Button' pane and Style field when a style is being used

Example showing how to apply a style

You must have already created the Continue and Lower styles. Say you want to apply these styles to buttons B1 and B2.

- Click Button B1.
- Use the drop-down menu for the Style field in the Properties pane to select the Continue <style>.

Controls\Button\Properties/Style


Properties 'Button'

Style: Next

Tag: <None>

Action: Set Bit

Bit: 0

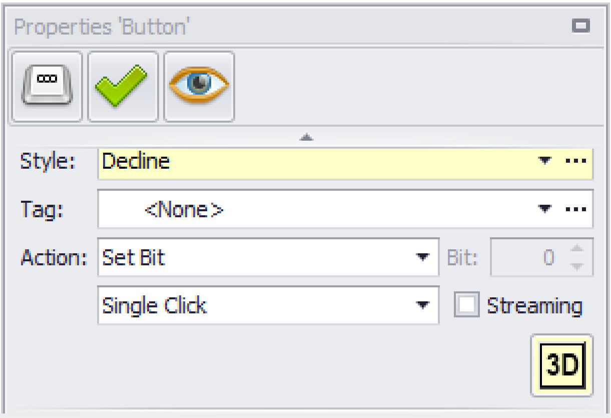
Single Click

□ Streaming

Figure 327: Properties 'Button' pane and Style field when a style is being used

- Click Button B2.
- Use the drop-down menu for the Style field in the Properties pane to select the Lower <style>.

Controls\Button\Properties/Style


Properties 'Button'

Style: Decline

Tag: <None>

Action: Set Bit

Bit: 0

Single Click

□ Streaming

Figure 328: Properties 'Button' pane and Style field when a style is being used

3.16.6 Modifying a style

To modify an existing style, follow the steps below:

- Click on a visualization object for which you want to modify the style being used, e.g., a button.
- Modify the properties for the visualization object as needed. For example, by selecting the 2D frame option and setting a green background for the off state.

When you make a change, the style will be reset to <None> in the *Properties '<visualization object'>, \Style field*.

- Click on the arrow for the >Style field in the the Properties 'visualization object' pane to open the corresponding drop-down menu.

- Select the following option: <Apply to 'name of style'>.
- Click on Caution! with OK.

You have now modified the style, and GALILEO will automatically change the appearance of every object using the style accordingly.

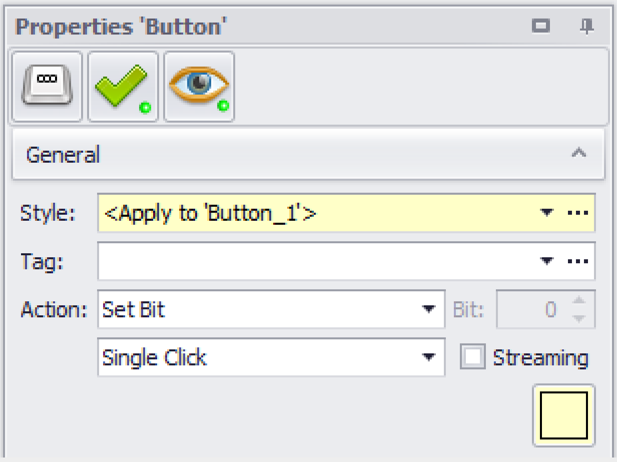
It is not possible to replace the style of all visualization objects with another style with one single action. In addition, the style of certain existing visualization objects cannot be changed either. For instance: If you want to change the style for all Bargraph visualization objects from <Dark> to <Pastel>, you will first need to select all visualization objects by selecting multiple objects at once. After doing so, your will be able to change the style for all the properties in the visualization objects.

Example showing how to modify a style

You must have already created the <Continue> style and applied it to multiple buttons. Now say you realize that a dark green color for the buttons is not noticeable enough and you want to change the background color for the off state to light green instead. In other words, that you want to modify the style.

- Click on the button A1.
- Set a light green background color for the button when the button is in the off state. For information on how to change the state color, please refer to → " Button", page 365
- Click on the arrow for the Style field in the Properties 'Button' pane to open the corresponding drop-down menu.
- Select <Apply to 'Continue'>.
- Click on Caution! with OK.

Controls\Button\General/Style



Properties 'Button'

General

Style: <Apply to 'Button_1'>

Tag:

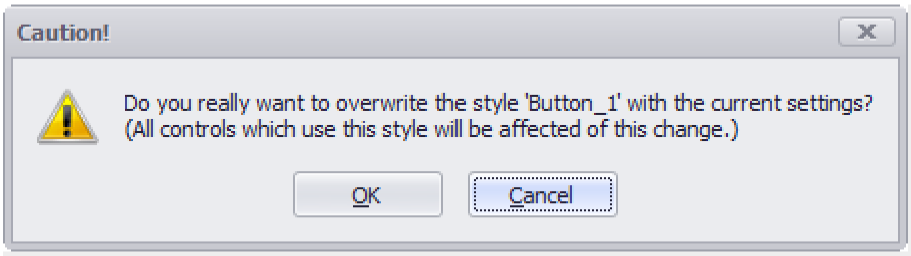
Action: Set Bit

Bit: 0

Single Click

□ Streaming

Figure 329: Properties 'Button' pane and Style field when modifying a style



Caution!

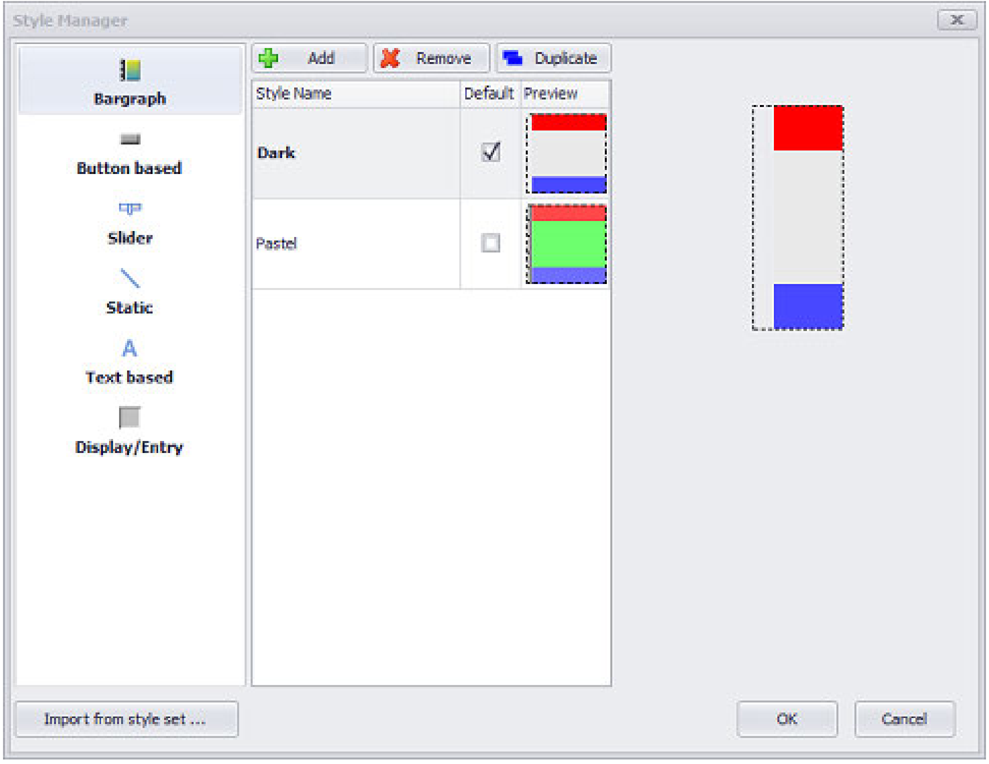
Do you really want to overwrite the style 'Button_1' with the current settings?
(All controls which use this style will be affected of this change.)

Figure 330: Prompt when modifying a style

3.16.7 Style sets

You can select pre-defined style sets for XV-300 devices, XP-500 devices, and GALILEO OPEN when creating a new project. Moreover, you can import these style sets into existing projects. The styles in these imported style sets will be applied as usual.

Home\Style-Manager tab



Style Name	Default	Preview
Dark	☑	🟥⬜🟦
Pastel	☐	🟥🟩🟦

Figure 331: Example showing a project that originally has the <Dark> style set and for which the <Pastel> style set imported later on.

→ The style in an imported style set will not be set as the default style automatically.

Multiple style sets can be imported at the same time. The following style sets are available:

Home\Style Manager\Import from style set… tab

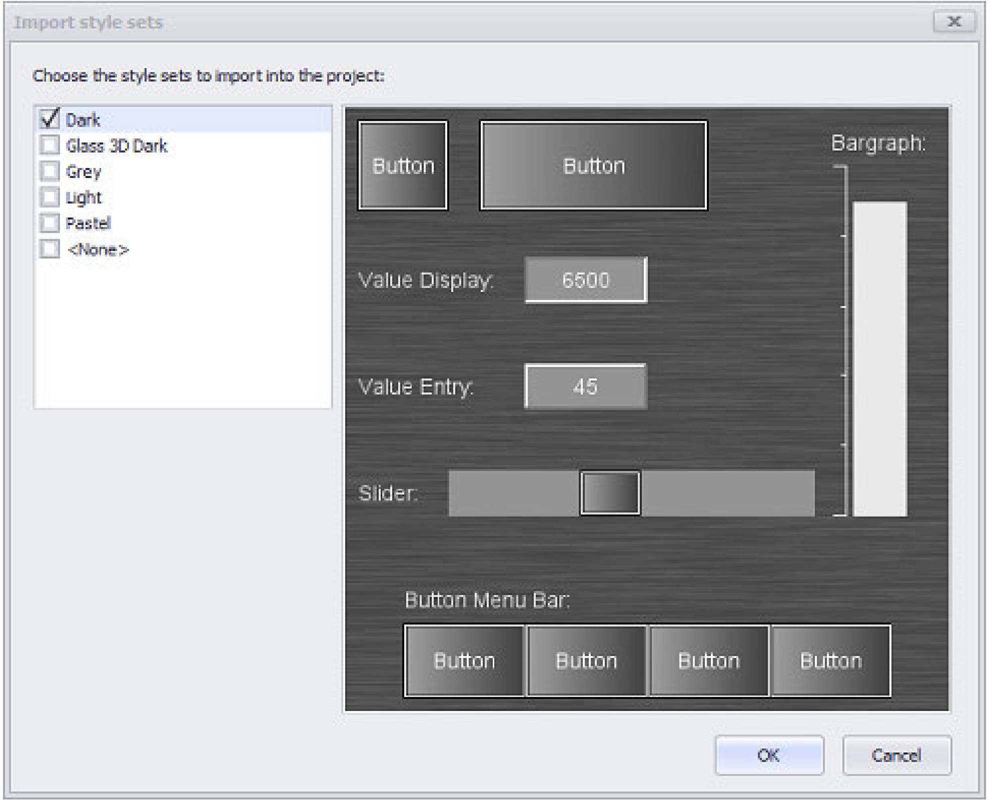


Figure 332: Selection dialog box with the style sets to be imported

3.17 Shapes

Shapes are used for the graphical design of screens. Mostly the properties (e.g. line width, line color) are configured statically. However, there is also the possibility to make these properties dependent on a tag. These dynamic settings are explained in the separate section "Dynamic properties".

The following drawing shapes are available:

Settings

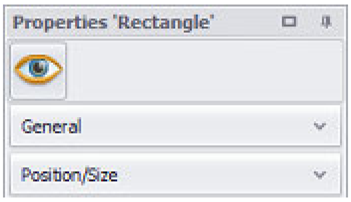
- → " Rectangle", page 468
- → " Line", page 471
- → " Ellipse", page 474
- → " Polygon", page 476

3.17.1 Rectangle

Used to insert a rectangular shape into a screen.

This shape can then be configured as necessary using the Properties 'Rectangle' pane.

Controls\ tab



'Rectangle' properties

→ section "Visibility", page 259

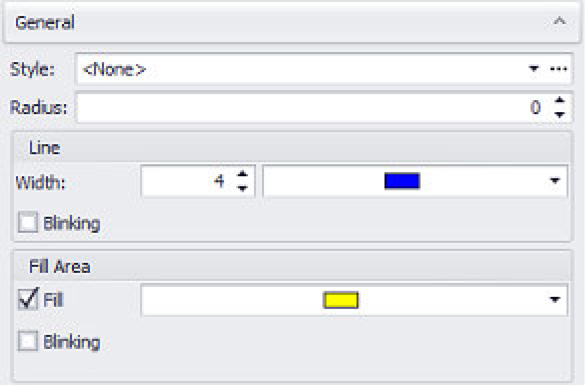
→ section "General", page 469

→ section "Position/Size", page 242

3.17.1.1 General

Defines the object's appearance on the screen

Controls\ Rectangle



Style: <None>
Radius: 0
Width: 4
□ Blinking
☑ Fill
□ Blinking

Figure 333: Rectangle Properties pane

Style

Settings that affect the appearance of visualization objects can be bundled and stored as a style. After this, it will be possible to use the style's name to use these settings at any time and apply them to new visualization objects of the same type throughout the entire project. In addition, when changes are made to a style, the changes will automatically be applied to all visualization objects using the style, ensuring that all visualization objects using the same style will have a consistent appearance. For detailed information on how to create a style, please refer to → section "Style management", page 458.

Rectangle attributes

Radius:

Used for corner radii. The corners will be drawn with the set radius.

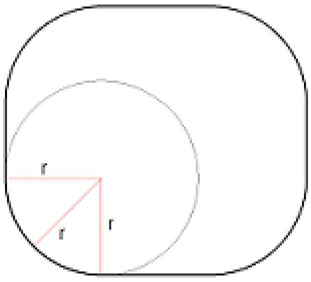


Figure 334: Rectangle corners with radii

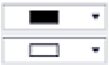
Specified in pixels

Line:

The line thickness for the shape outline drawn with the selected color.

Specified in pixels

Color Selection



The color of the shape outline and the color with which the rectangle will be filled can be selected separately.

For detailed information on the available colors, please refer to the following section:
→ section "Color choice - Object's Color Palette", page 234

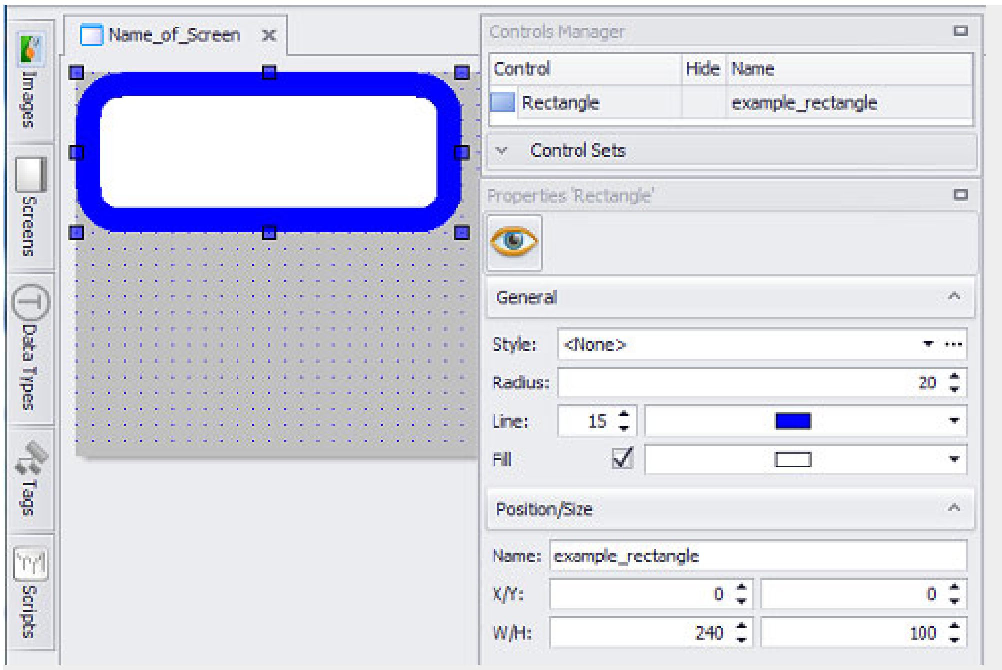
Filled

☑ Turn on Enabling the checkbox

Enabling this checkbox will fill the rectangle with the → "Color choice - Object's Color Palette", page 234.

Example

A rectangle filled with white, featuring rounded corners and a blue outline
Attributes: Line = 15; Colors = blue, white fill; Radius = 20; inserted at the top of the screen



Controls Manager

Control	Hide	Name
Rectangle		example_rectangle

Control Sets

Properties 'Rectangle'

General

Style: <None>
Radius: 20
Line: 15
Fill ☑

Position/Size

Name: example_rectangle
X/Y: 0 0
W/H: 240 100

Figure 335: Rectangle object example

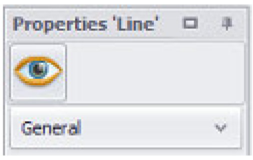
3.17.2 Line

The line static control can be used to draw straight lines and geometrical shapes derived from straight lines. You can select any color and line thickness for these lines.

Used to insert a line on the screen.

After this, the line can be configured as necessary using the Properties 'Line' pane.

Controls\ tab



'Line' properties

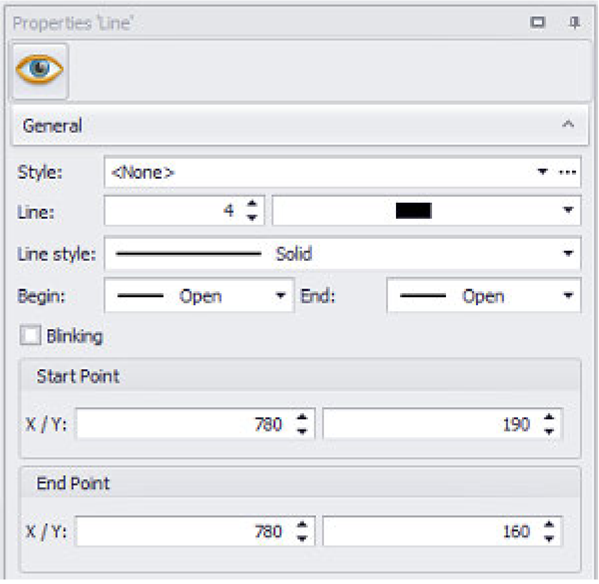
→ section "Visibility", page 259

→ section "General", page 471

3.17.2.1 General

Defines the object's appearance on the screen

Controls\ Line



Properties 'Line'

General

Style: <None>

Line: 4

Line style: Solid

Begin: Open End: Open

□ Blinking

Start Point

X / Y: 780 190

End Point

X / Y: 780 160

Figure 336: Rectangle Properties pane

Style

Settings that affect the appearance of visualization objects can be bundled and stored as a style. After this, it will be possible to use the style's name to use these settings at any time and apply them to new visualization objects of the same type throughout the entire project. In addition, when changes are made to a style, the changes will automatically be applied to all visualization objects using the style, ensuring that all visualization objects using the same style will have a consistent appearance. For detailed information on how to create a style, please refer to → section "Style management", page 458.

Line attributes

Line:

Line thickness of the line drawn with the color you want.

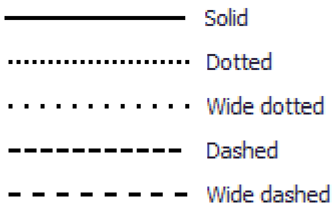
Specified in pixels

You can select the color you want for the line.

For detailed information on the available colors, please refer to the following section: → section "Color choice - Object's Color Palette", page 234

Line style

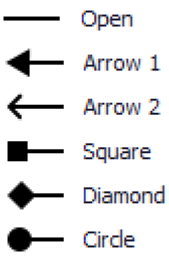
The following styles are available:



Solid
Dotted
Wide dotted
Dashed
Wide dashed

Begin/End

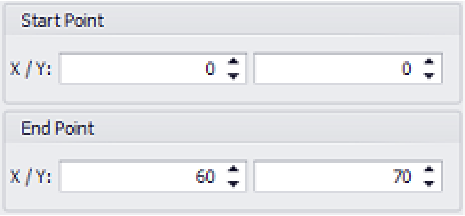
To choose from for the line:



Open
Arrow 1
Arrow 2
Square
Diamond
Circle

Start Point and End Point

Used to set the line's position and length on the screen.



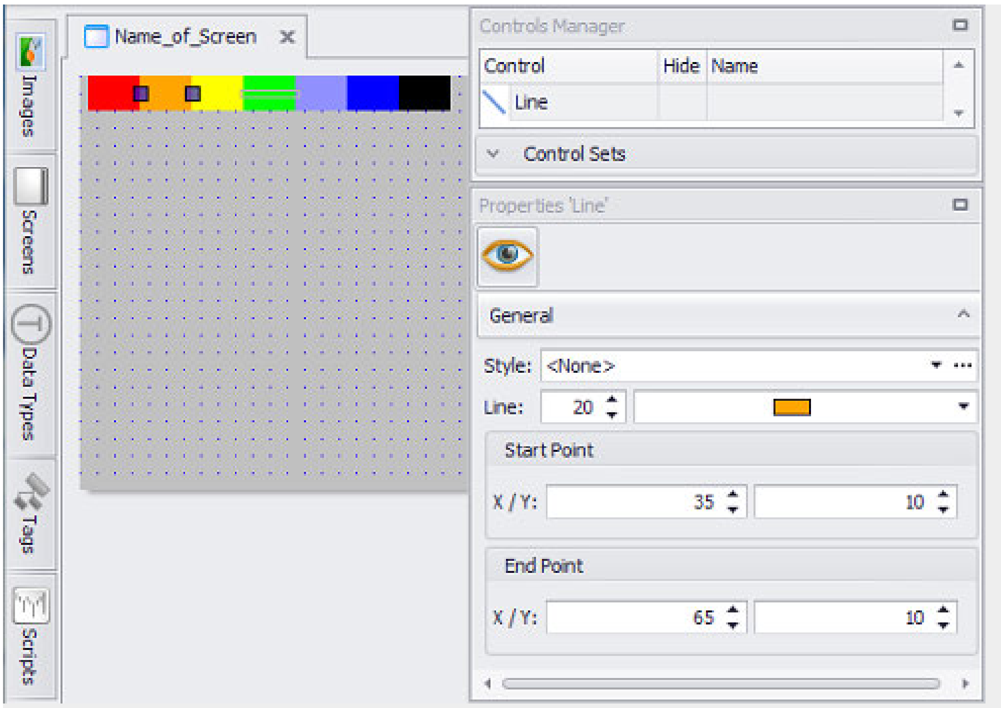
Start Point
X / Y: 0 0

End Point
X / Y: 60 70

X/Y Absolute horizontal (X) and
vertical (Y) position relative to the screen
0/0 origin at top right corner
Specified in pixels

Example

Rainbow graphic made up of 7 identical lines (35 pixels long, 5 pixels wide)
Attributes: Line = 20, Colors = Various, Start Point = End Point of previous line



Control	Hide	Name
Line		

Style: <None>

Line: 20

Start Point

X / Y: 35 10

End Point

X / Y: 65 10

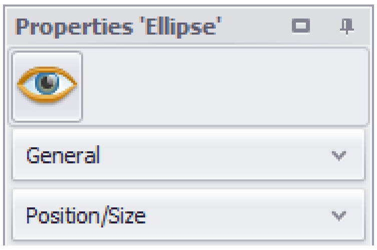
Figure 337: Line static control example

3.17.3 Ellipse

Used to insert an ellipse shape into a screen.

After this, the ellipse can be configured as necessary using the Properties 'Ellipse' pane.

Controls\ tab



'Ellipse' properties

→ section "Visibility", page 259

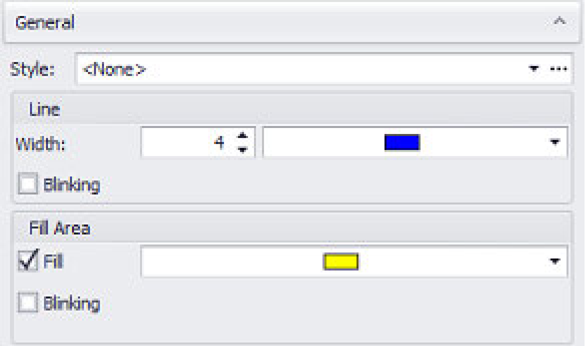
→ "General", page 474

→ section "Position/Size", page 242

3.17.3.1 General

Defines the object's appearance on the screen

Controls\ Ellipse


General

Style: <None>

Line

Width: 4

□ Blinking

Fill Area

☑ Fill

□ Blinking

Figure 338: Ellipse Properties pane

Style

Settings that affect the appearance of visualization objects can be bundled and stored as a style. After this, it will be possible to use the style's name to use these settings at any time and apply them to new visualization objects of the same type throughout the entire project. In addition, when changes are made to a style, the changes will automatically be applied to all visualization objects using the style, ensuring that all visualization objects using the same style will have a consistent appearance. For detailed information on how to create a style, please refer to → section "Style management", page 458.

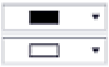
Ellipse attributes

Line:

Width: The line thickness for the shape outline drawn with the selected color.

Specified in pixels

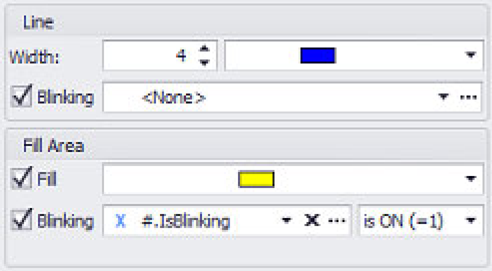
Color Selection


The color of the shape outline and the color with which the polygon will be filled can be selected separately.

For detailed information on the available colors, please refer to the following section: → section "Color choice - Object's Color Palette", page 234

Blinking

Lines and fill areas can be configured as blinking.


Line

Width: 4

☑ Blinking <None>

Fill Area

☑ Fill

☑ Blinking #.IsBlinking is ON (=1)

Blinking without tag assignment: The line or the filling area blinks continuously.

Blinking with tag assignment: A bit tag can be assigned and the logic (ON/OFF) configured. The line or the fill area will only blink if the condition is met.

Fill Area:

Filled

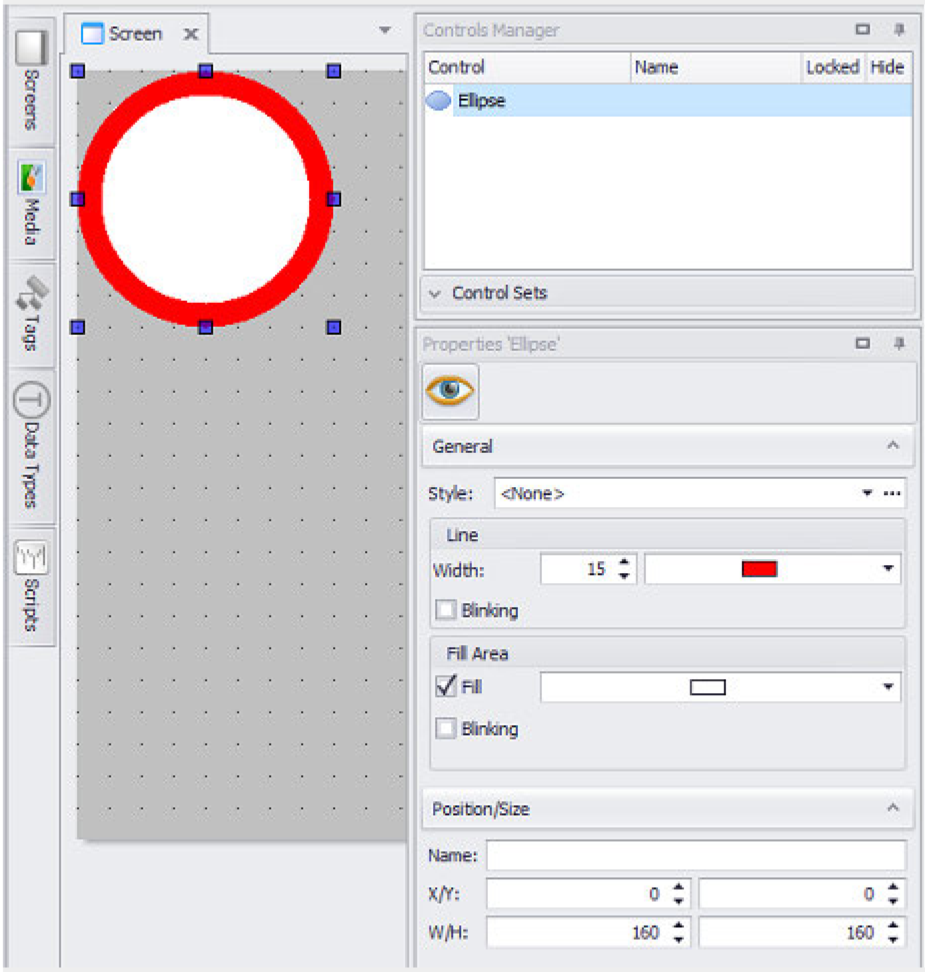
☑ Turn on Enabling the checkbox

Enabling this checkbox will fill the ellipse with the → "Color choice - Object's Color Palette", page 234.

Example

A circle filled with white and with a red outline

Attributes: Radius = 160; Line = 15; Colors = red, white fill; inserted at the top of the screen



Control	Name	Locked	Hide
Ellipse			

Style: <None>

Line
Width: 15 🟥
□ Blinking

Fill Area
☑ Fill □
□ Blinking

Name:
X/Y: 0 0
W/H: 160 160

Figure 339: Ellipse static control example

3.17.4 Polygon

Used to insert a "polygon" drawing element in order to draw special shapes.

To achieve this, several straight lines are connected by vertices.

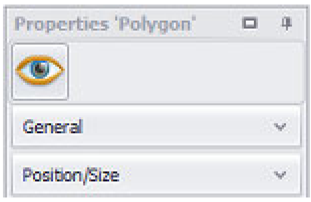
The end point of the last line will automatically be connected to the start point of the first line, closing the object.

The context menu can be used to insert and remove vertices.

The shape is drawn by clicking on the screen.A polygon vertex will be inserted at every position where the left mouse button is clicked. To finish entering the polygon, double-click on the start point or press the ESC or Enter key.

Once you are done, you can use the Properties 'Polygon' pane to configure the polygon.

Controls\ tab



'Polygon' properties

→ section "Visibility", page 259

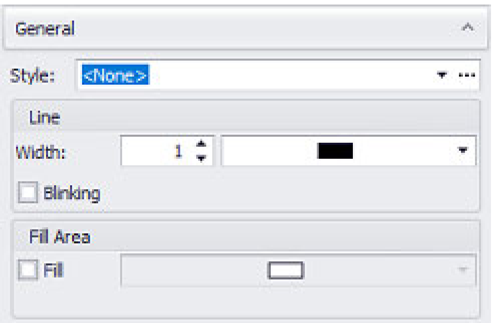
→ section "General", page 477

→ section "Position/Size", page 479

3.17.4.1 General

Defines the object's appearance on the screen

Controls\ Polygon



Style: <None>

Width: 1

☐ Blinking

☐ Fill

Figure 340: Polygon Properties pane

Style

Settings that affect the appearance of visualization objects can be bundled and stored as a style. After this, it will be possible to use the style's name to use these settings at any time and apply them to new visualization objects of the same type throughout the entire project. In addition, when changes are made to a style, the changes will automatically be applied to all visualization objects using the style, ensuring that all visualization objects using the same style will have a consistent appearance. For detailed information on how to create a style, please refer to →

section "Style management", page 458.

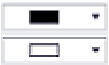
Polygon attributes

Line:

Width: The line thickness for the shape outline drawn with the selected color.

Specified in pixels

Color Selection



The color of the shape outline and the color with which the polygon will be filled can be selected separately.

For detailed information on the available colors, please refer to the following section:
→ section "Color choice - Object's Color Palette", page 234

Fill Area:

Filled

☑ Turn on Enabling the checkbox

Enabling this checkbox will fill the ellipse with the → "Color choice - Object's Color Palette", page 234.

Example

A yellow polygon with an outline with a constant color
Attributes: Line = 5, Colors = yellow, yellow fill; inserted at the top of the screen

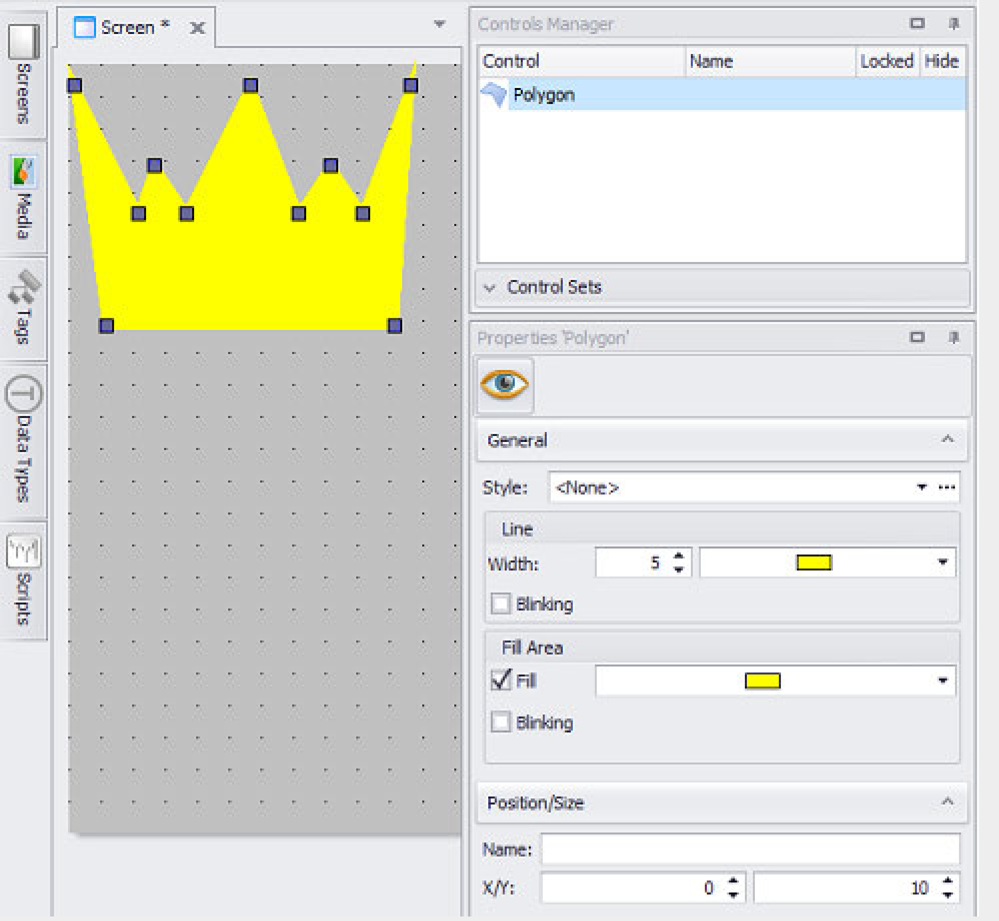
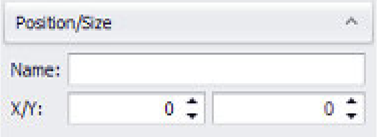


Figure 341: Polygon static object example

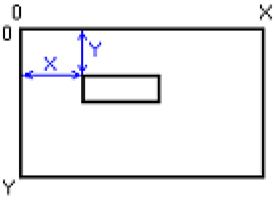
3.17.4.2 Position/Size

Used to define the location on the screen

The coordinates entered will be fore the vertex that is farthest up and to the left.



Name:	Used to enter a Type name for identifying the static control Notation: No spaces or special characters, no umlauts
X/Y	Absolute horizontal (X) and vertical (Y) position relative to the screen 0/0 origin at top right corner



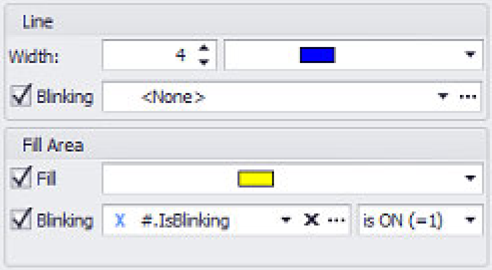
Specified in pixels

Locking the object so that it cannot be moved

☑ Enable the "Locked" checkbox in the → "Controls Manager window", page 71 pane.

Blinking

Lines and fill areas can be configured as blinking.



Line

Width: 4

☑ Blinking <None>

Fill Area

☑ Fill

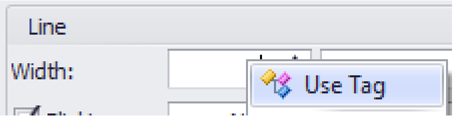
☑ Blinking #.IsBlinking is ON (=1)

Blinking without tag assignment: The line or the filling area blinks continuously.

Blinking with tag assignment: A bit tag can be assigned and the logic (ON/OFF) configured. The line or the fill area will only blink if the condition is met.

Dynamic line width

The line width can be made dependent on a tag value at runtime. To configure a dynamic line width, the context menu of the edit field for the line width must be called up with the right mouse button. There, "Use tag" can then be selected:



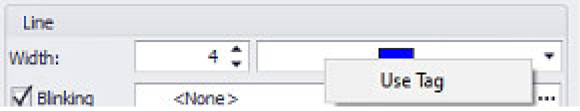
Subsequently, a numeric tag can be selected.

Negative tag values are interpreted as 0. No line is then drawn.

Dynamic coloring

The line and fill color can be made dependent on a variable value at runtime. To configure a dynamic coloring, the context menu of the edit field for the line or fill color must be called up with the right mouse button.

There you can select "Use tag":

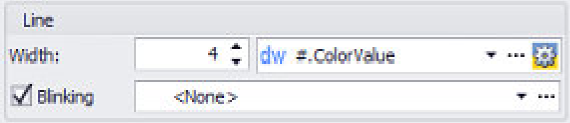


Line

Width: 4

☑ Blinking <None>

After a tag has been selected, the color setting icon ⚙ can be clicked to select the configuration type:



Line

Width: 4 dw #.ColorValue

☑ Blinking <None>

In the dialog that appears you can choose between two different color modes:

ARGB

The value of the assigned tag is interpreted as ARGB color value and the line or fill color is drawn accordingly. A value of 4'294'901'760 (Hex: 0xFFFF0000) corresponds to the color red, a value of 2'030'042'880 (Hex: 0x78FFFF00) corresponds to the color yellow with an alpha transparency of 120 (0x78).

In ARGB mode only tags of type DWORD are allowed. For tags of other types a compiler error message is generated.

→ Alpha transparencies are supported at runtime only for fill colors. Lines can only be drawn fully colored or fully transparent.

Color set

Color sets can be used to achieve different colorings based on variable values, linked to conditions.

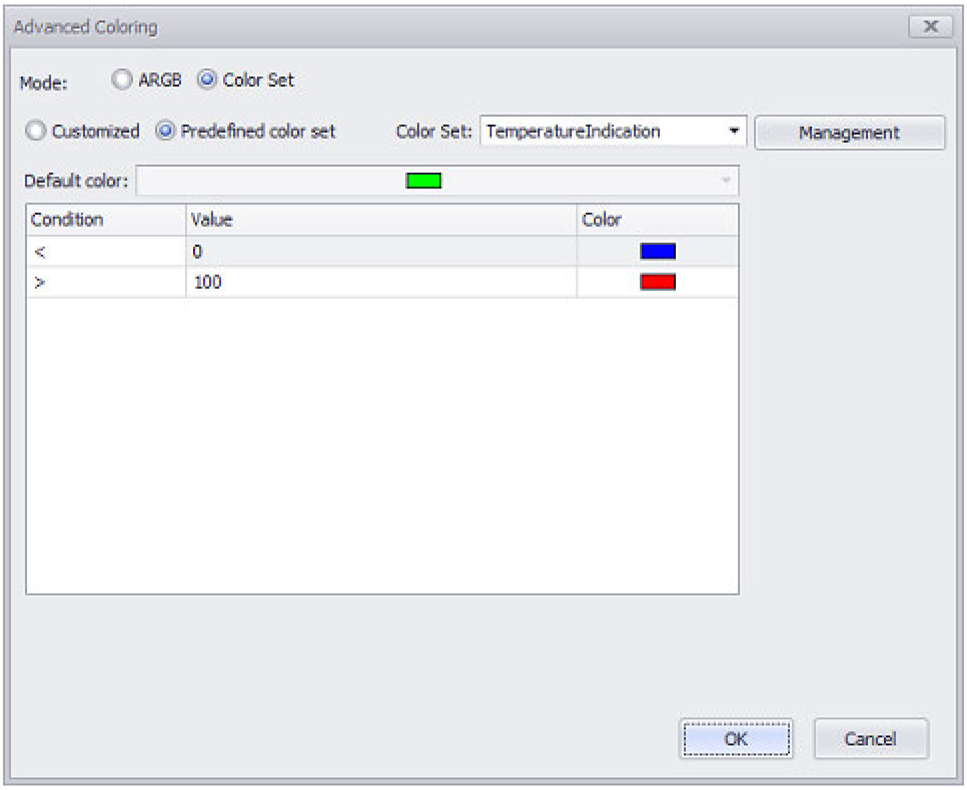
Color sets are therefore very well suited for limit value representations, for example. For example:

- Variable value < 0 → Blue line
- Variable value >= 0 and <= 100 → Green line
- Variable value > 100 → Red line

Color sets are distinguished between two different types:

- Customized: the color set configuration exists exclusively for the current configuration purpose (line color, fill color) at the current drawing object.
- Predefined color set: Project-wide available color sets can be configured. These can then be used for several drawing objects. A change to the color set itself directly affects all drawing objects that use this color set.

The dialog for using color sets looks like this:



Advanced Coloring

Mode: ○ ARGB ● Color Set

○ Customized ● Predefined color set

Color Set: TemperatureIndication

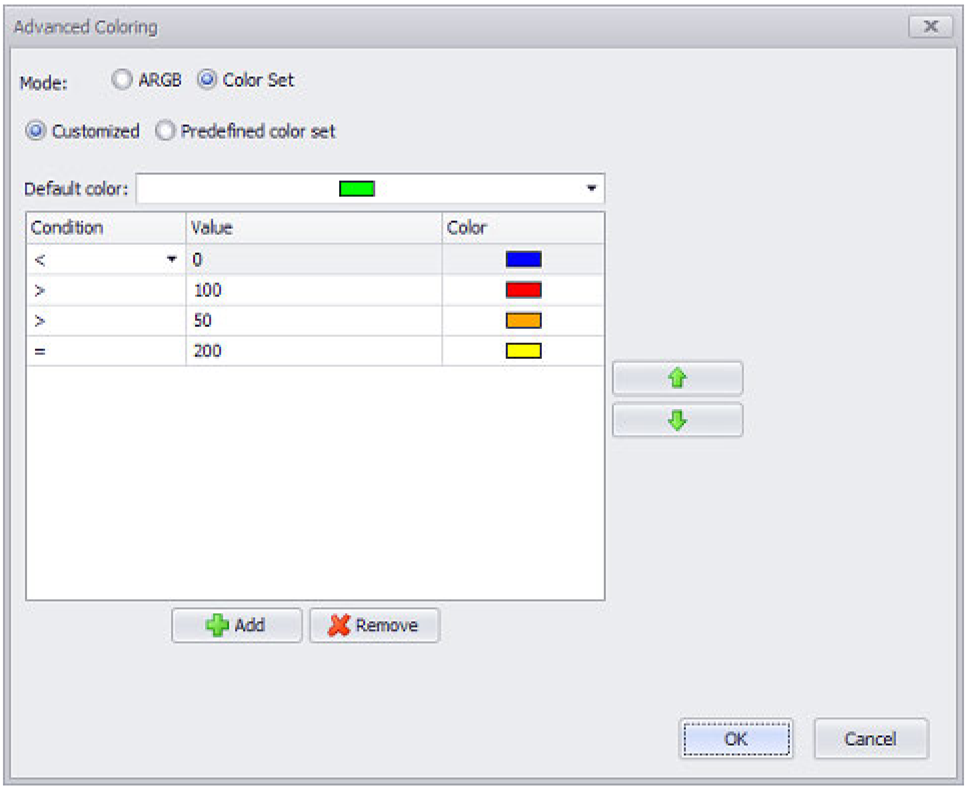
Default color: 🟩

Condition	Value	Color
<	0	🟦
>	100	🟥

When selecting a predefined color set, the fields for default color and the table with conditions are in a "read-only" mode and are for information only.

The management of color sets and their settings can be done using the "Management" button.

A color set configuration (when set to Customized or in the color set management) consists of the following:



Mode: ○ ARGB ● Color Set

● Customized ○ Predefined color set

Default color: 🟩

Condition	Value	Color
<	0	🟦
>	100	🟥
>	50	🟧
=	200	🟨

default color: color that will be used if none of the configured conditions apply

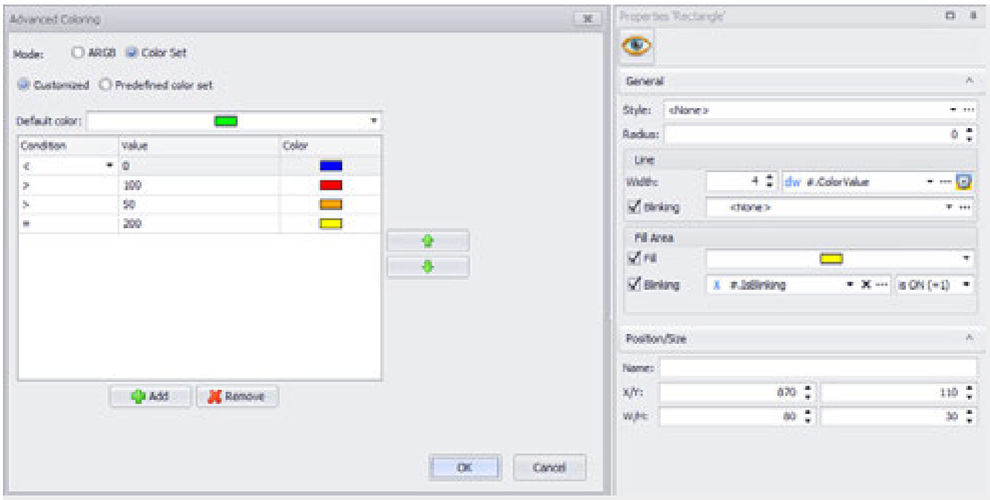
Conditions, consisting of:

- Condition logic: <, <=, =, >, >=.
- Value: the specification of a static value or a tag for checking the condition (toggleable from the context menu). In case of parameterized user-defined objects, it is also possible to select the parameter instead of a tag in the "Individual" mode.
- Color: color to be used if the condition is true.

At runtime, the conditions are checked in the order configured here. The order can therefore be changed with the arrow keys. The check at runtime is terminated as soon as the first condition applies.

For example, the configuration shown in the case of the line "Value = 200 Yellow" is nonsensical: Due to the order in which the conditions are checked, a value of 200 would already fulfill the condition " > 100" and thus the color red would be applied.

Example of a configuration



Mode: ○ ARGB ● Color Set

● Customized ○ Predefined color set

Default color: 🟩

Condition	Value	Color
<	0	🟦
>	100	🟥
>	50	🟧
=	200	🟨

Style: <None>

Radius: 0

Width: 4 dw #.ColorValue

☑ Blinking <None>

☑ Fill 🟨

☑ Blinking X #.IsBlinking is ON (=1)

Name:

X/Y: 870 110

W/H: 80 30

- the rectangle uses the internal Galileo variable "ColorValue" for the line color
- if the value of "ColorValue" is < 0, the line is drawn blue
- if the value of "ColorValue" is > 100, the line is drawn red
- if the value of "ColorValue" is > 50 (but <= 100), the line will be drawn orange
- the configuration of "= 200" is invalid as mentioned before
- if no condition applies (which is the case for values between 0 and 50), the line is drawn green

3.18 Switch

In real-life applications, this visualization object is used to make it possible to select between various control sequences by means of a value change. When the button for the visualization object is tapped on, the value for a numeric tag of data type BIT, BYTE, WORD, or DWORD will be changed to a specific value. This tag can also be a PLC tag.

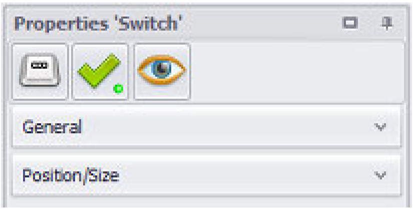
There are two different functions that can be applied to this visualization object. The Set action will change the value of a numeric tag to a specific bit sequence. Meanwhile, the Shift action will move the bit within a numeric tag to the left or right, which for all intents and purposes is also a change in value for the corresponding tag. This means that switches can be used as an input object in that the corresponding tag will assume different defined values every time the switch is clicked or tapped on. However, it is important to note that switches can also be used as output objects. The settings in the States tab can be used to configure a different appearance for every bit at a specific bit position that is set to <1>.

The visualization object will be displayed as a button, with the corresponding appearance configured using the settings in the States tab. A separate state with its own appearance can be defined for every step / every bit sequence.

If several Switches are assigned the same tag, it is possible to configure them to toggle between switch on/off.

If this visualization object is used multiple times to display a large number of system components with the same parameter sets, it is recommended to use parameters. Using parameters will optimize your work when it comes to assigning tags to the visualization object.
For detailed information on how to work with parameters, please refer to → section "Using user controls to optimize work", page 279.

Controls\ tab



Properties 'Switch'

▭ → " States", page 491

✓

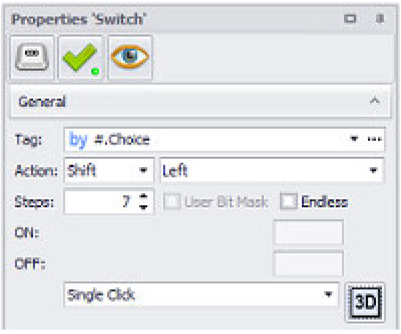
👁

→ "General", page 486

The ✓ accessibility button will only be shown in the Properties pane if a tag and a keyboard have been assigned to the visualization object, as it would not make sense to define accessibility conditions otherwise.

3.18.1 General

Controls\ Switch



Properties 'Switch'

General

Tag: #.Choice

Action: Shift Left

Steps: 7 ☐ User Bit Mask ☐ Endless

ON:

OFF:

Single Click 3D

Figure 342: Properties 'Switch' pane with the SHIFT action selected

Variable

The visualization object will show the value of the selected tag on the device screen at runtime. Clicking on the drop-down menu arrow will open a list of available tags (BIT, BYTE, WORD, DWORD, QWORD, FLOAT). Please note that in order for the tag you want to be shown in this list, the tag must first be imported or created in the Tags tab.

A parameter can also be specified instead of a tag. In this case, the Tag field will turn into a Type field. In order to be able to work with parameters, the visualization object must be placed on a UserControl. For detailed information on how to work with parameters, please refer to → "Using input parameters", page 279.

Values do not necessarily have to be displayed or entered the way the PLC needs them or the way they are stored in the tags. The Value Conversion Manager can be used to convert tag values and assign a variety of units to them before they are shown by visualization objects. The way value conversion works is described in greater detail in a different section – please refer to → "Conversions", page 921.

Action Shift

When the Shift action is selected, tapping on the button for the switch visualization object on the device screen will move the tag bit with a value of <1> one position to the left or right. This will be done as many times as specified in the Steps field. The bit with a value of <1> will remain at the last position to which it is moved. The maximum number of steps will depend on the value range for the selected tag, as well as on whether the corresponding limits are signed or unsigned!

For unsigned tags, the maximum number of steps will be as follows: BYTE = 8, WORD = 16, DWORD = 32.

For signed tags, the maximum number of steps will be as follows: BYTE = 7, WORD = 15, DWORD = 31.

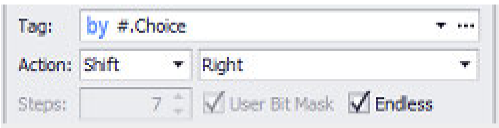
If the Endless checkbox is enabled and the specified number of steps is reached, the bit will start over from bit 0 when using LEFT bit shifting and from the most significant bit when using RIGHT bit shifting. In other words, enabling this checkbox will allow the bit to move in an endless loop.

If a tag's default value has multiple bits with a value of <1>, only the least significant bit will be taken into account and shifted one position. All other bits will be set to <0>.

If none of the tag's bits have a value of <1>, clicking or tapping on the switch will cause the bit at position 1 to be set to <1> in the case of left shifting. In the case of right shifting, the bit at position 8, 16, or 32 (as applicable) will be set to 1.

Steps	8	7	6	5	4	3	2	1
Tag of data type byte, unsigned, is:	0	0	0	0	0	0	0	0
Click with the following shifting direction:								
Left	0	0	0	0	0	0	0	1
Right	1	0	0	0	0	0	0	0

Example Shift Left, Endless



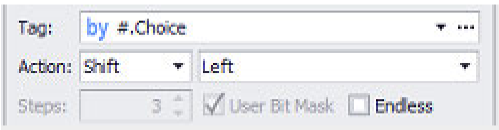
Tag: by #.Choice

Action: Shift Right

Steps: 7 ☑ User Bit Mask ☑ Endless

The <choice> tag is of data type BYTE and has a value of 8_{dec}, i.e., the following bit sequence: 0000 1000. When theswitch visualization object is clicked or tapped on, the bit will be moved one position to the left, resulting in the following bit sequence: 0001 0000, or a value of 16_{dec}. Clicking or tapping on the switch visualization object once more will change thechoice tag's value to 32_{dec}, and doing so again after that will make it change to 64_{dec}. This last change will also mean that the maximum number of steps has been reached, so another click or tap will make the bit sequence start over at 0000 0001.

Example Shift Left



Tag: by #.Choice

Action: Shift Left

Steps: 3 ☑ User Bit Mask □ Endless

The <choice> tag is of data type BYTE and has a value of 26_{dec}, i.e., the following bit sequence: 0001 1010.

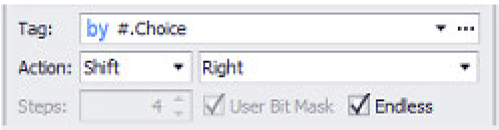
When the switch visualization object is clicked or tapped on, the tag's least significant bit will be moved one position to the left.

In this case, the initial bit sequence will be: 0000 0010. After the least significant bit is moved to the left, the new bit sequence will be 0000 0100, or a value of 4_{dec}.

If the object is clicked or tapped on again, it will not be possible to move the bit another position to the left. Accordingly, the tag will keep the following bit sequence: 0000 0100.

If the Endless checkbox had been enabled, the set bit would be moved to the first position in the bit sequence, resulting in the following bit sequence: 0000 0001.

Example Shift Right



Tag: by #.Choice
Action: Shift Right
Steps: 4 ☑ User Bit Mask ☑ Endless

The <choice> tag is of data type byte and has a value of 6_{dec}, i.e., the following bit sequence: 0000 0110.

When the switch visualization object is clicked or tapped on, the tag's least significant bit will be moved one position to the right.

In this case, the initial bit sequence will be: 0000 0010. After the least significant bit is moved to the right, the new bit sequence will be 0000 0001, or a value of 1_{dec}.

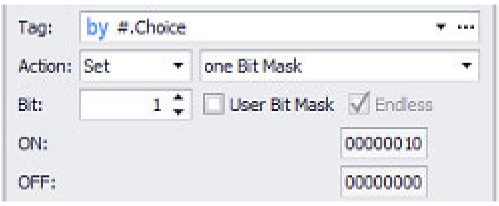
If the switch visualization object is clicked or tapped on again, it will not be possible to move the bit another position to the right. Since the Endless checkbox is enabled, the set bit will be moved to the fourth position in the bit sequence, resulting in the following bit sequence: 0000 1000.

This means that the value of 4 in the Steps field determines the maximum position of the set bit in the bit sequence!

Action Set

When the Set action is selected, tapping on the switch visualization object on the device screen will set the selected tag to a defined value. This value is entered as a bit sequence in the ON field. The switch can also be designed as a button, in which case the tag will be assigned the bit sequence in the OFF field as soon as the switch is released.

The length for the bit sequences in the ON and OFF field will depend on the selected tag's value range. The counting order for the corresponding bits is from right to left, and the length for the aforementioned bit sequences is as follows: byte = bits 0 to 7; word = 0 to 15; DWord = 0 to 31.



Tag: by #.Choice
Action: Set one Bit Mask
Bit: 1 □ User Bit Mask ☑ Endless
ON: 00000010
OFF: 00000000

Bit

The value in the Bit field is used to specify which bit in the ON field should be set to <1>. Only one bit in the ON field can be set; all other bits will have a value of <0>. The value for the ON field cannot be entered directly. When the switch visualization object is clicked or tapped on, the selected tag will be assigned the bit sequence in the ON field.

one Bit Mask

If the one Bit Mask option is selected, the switch will remain in the ON state and the tag will keep the assigned value from the ON field.

two Bit Masks

If the two Bit Masks option is selected, the mode Stream ON or Toggle can be selected.

Stream ON
The selector switch has the ON state and the tag takes the assigned value from the ON field as long as the selector switch button is pressed.
If the button is released, the selector switch changes to the OFF state and the tag takes the value from the OFF field.

Toggle
The selector switch has the ON state and the tag has the assigned value from the ON field when the button is pressed and remains at this value until the button is pressed again.
When the button is released, the selector switch changes to the OFF state and the tag takes the value from the OFF field.

The value in the OFF field is fixed at 0 and cannot be changed. If another value is to be used, User bit mask must be selected.

User Bit Mask

Any bit mask can be specified in the OFF field. To determine which state the selector switch currently has, the bit at Bit for on/off state behavior is used.

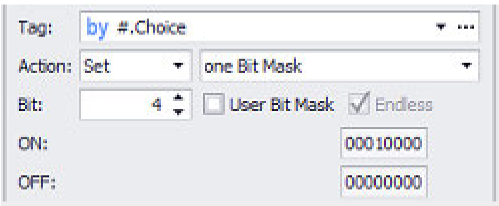
Example

Bit mask for ON = 00001111

Bit mask for OFF =00001000

To be able to force a distinction of the states, bit 0, 1 or 2 must be specified as Bit for on/off state behavior. Only these 3 bits differ in the two masks and thus allow a unique state identification.

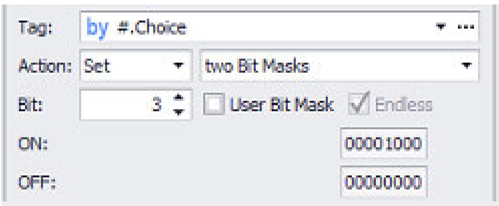
Example one Bit Mask



Tag: by #.Choice
Action: Set | one Bit Mask
Bit: 4 ☐ User Bit Mask ☑ Endless
ON: 00010000
OFF: 00000000

The value set in the Bit field will set bit 4 in the ON field to a value of <1>, which corresponds to a value of 16_{dec}. When the switch visualization object on the device screen is tapped, the choice tag will be assigned the bit sequence in the ON field, i.e., the choice tag will assume a value of 16_{dec}. If the operator ceases to press on the object, the tag will still maintain its value.

two Bit Masks example



Tag: by #.Choice

Action: Set | two Bit Masks

Bit: 3 | □ User Bit Mask | ☑ Endless

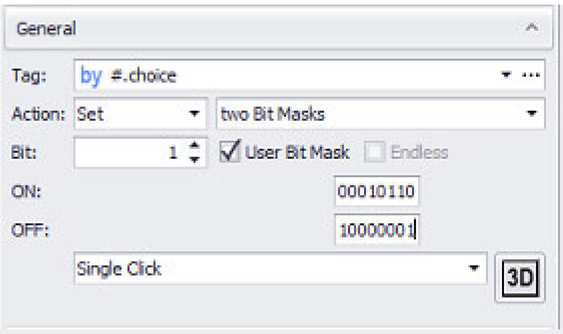
ON: 00001000

OFF: 00000000

Figure 343: Section from Properties 'Switch' pane with choice tag of data type BYTE

The value set in the Bit field will set bit 3 in the ON field to a value of <1>, which corresponds to a value of 8_{dec}. When the switch visualization object is tapped, the choice tag will be assigned the bit sequence in the ON field, i.e., the choice tag will assume a value of 8_{dec}. If the operator ceases to press on the object, the tag's value will change to the bit sequence in the OFF field, i.e., choice will assume a value of 0_{dec}.

Example User Bit Mask



General

Tag: by #.choice

Action: Set | two Bit Masks

Bit: 1 | ☑ User Bit Mask | □ Endless

ON: 00010110

OFF: 10000001

Single Click | 3D

The User Bit Mask checkbox is enabled and the <two Bit Masks> option has been selected. A bit sequence of 00010110 is entered into the <ON> field, i.e., a value of 22_{dec}. Meanwhile, a bit sequence of 10000001 is entered into the <OFF> field, i.e., a value of 129_{dec}. When the switch visualization object is pressed, the choice tag will be assigned the bit sequence in the ON field, i.e., the choice tag's value will change to 22_{dec}. When the tag is released, it will be assigned the bit sequence in the OFF field, i.e., the choice tag's value will change to 129_{dec}.

Please note that the tag must be defined as an unsigned tag. OtherwiseGALILEO will output an error message.

Selection mode on panel Single Click

The following modes can be adjusted for visualization objects with a keyboard:

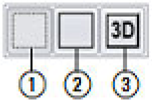
- Single Click
  Tapping once on a list item or a button on the device screen will activate the selection.
- Delayed
  This option is ideal for implementing a safety function designed to protect against accidental actions. In order for a button on the device screen to be activated, it

will first have to be pressed and held (long press) for the configured delay time. By default, this delay time will be 0.5 s, but it can be changed in *Home\Settings\Runtime tab\Touch Settings*.

- Double Click
  Tapping twice on a list item or button on the device screen will select the item / activate the button. By default, the double-click speed will be 0.5 s for a double-click, but it can be changed in*Home\Settings\Runtime tab\Touch Setting*.

Frame 3D

The visualization object can be displayed with any one of three different designs.



①	no frame
②	with frame
③	3D frame

3.18.2 States

The available states for a switch for which the *Shift* action has been selected in Properties\General will be different from the states available for a switch for which the Set action has been selected.

Shift

If the Shift option is selected, the number of states will be equal to the number of steps defined in the Properties\General: Steps field, plus one more state if all bits are set to <0>.

Set

If the Set option is selected, there will be exactly two states for the visualization object: one in which the tag will contain the bit sequence in the ON field and one in which it will contain the bit sequence in the OFF field.

The basic configuration parameters for the visualization object's foreground and background are described in detail in the following section: Defining text series

3.19 A Text

Settings

→ " Text", page 492

→ " HTML viewer", page 435

Used to insert a text box as a placeholder.

The Properties 'Text' pane can be used to fill the box and format and position the corresponding text.

You can either enter new text or select an existing standard text for insertion.

→ The standard texts available in the project can be managed by clicking on Home/Texts and will be listed with an ID number in the Text Editor dialog box that appears.

Normally, texts are handled as a fixed, static object.

However, GALILEO can assign a dynamic behavior to this static object.

To do this, you will need to use placeholders in the running text so that these placeholders will be filled out with the current tag value when the text is displayed.

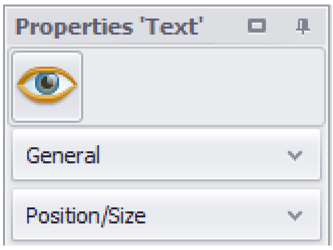
→ There can be a maximum of five placeholders per text.

By using this functionality, you can have displays such as <The current temperature is 23 °C, the humidity is 85%, and the wind speed is 4 Bft.>
and have the readings be integrated as dynamic tag values.

This option is available for visualization objects to which Dynamic string parameters can be assigned as a property. These objects include buttons, as well as screen changes and flag displays.

For more information please refer to → "Working with the Text Editor ", page 640

Controls\A tab



A 'Text' properties

→ section "Visibility", page 259

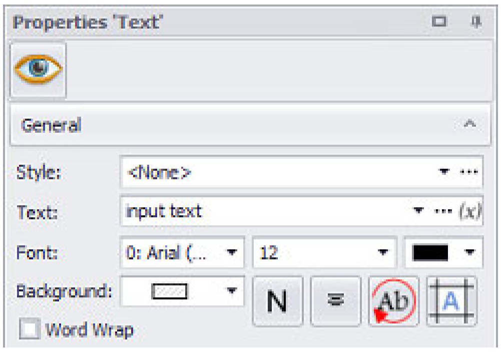
→ section "General", page 492

→ section "Position/Size", page 242

3.19.1 General

Defines the object's appearance on the screen

Controls\ A Text



Properties 'Text'

General

Style: <None>

Text: input text

Font: 0: Arial (... 12

Background:

☐ Word Wrap

Figure 344: Text Properties pane

Style

Settings that affect the appearance of visualization objects can be bundled and stored as a style. After this, it will be possible to use the style's name to use these settings at any time and apply them to new visualization objects of the same type throughout the entire project. In addition, when changes are made to a style, the changes will automatically be applied to all visualization objects using the style, ensuring that all visualization objects using the same style will have a consistent appearance. For detailed information on how to create a style, please refer to → section "Style management", page 458.

Text

This field can be used to enter text directly. Theoretically, the text length per entry is unlimited with the available fonts.

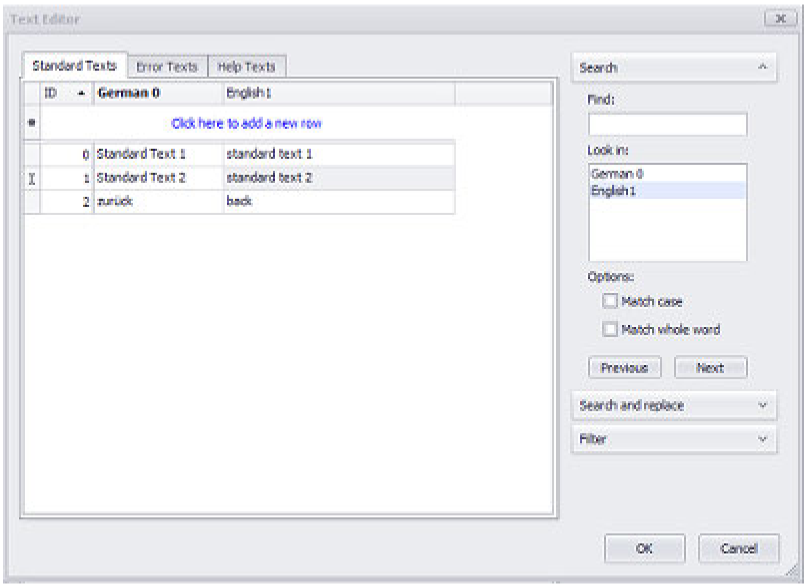
The text you enter will be added to the standard texts for the project.

To add a text that already exists, click on ….

The Text Editor dialog box will appear and show the standard texts that already exist.

For more information please refer to → "Working with the Text Editor ", page 640

Controls\ Text



ID	German 0	English 1
0	Standard Text 1	standard text 1
1	Standard Text 2	standard text 2
2	zurück	back

Figure 345: Properties 'Text' pane after inserting a text box into the screen in order to select an existing standard text

Adding parameters for dynamic text with the (x) button.

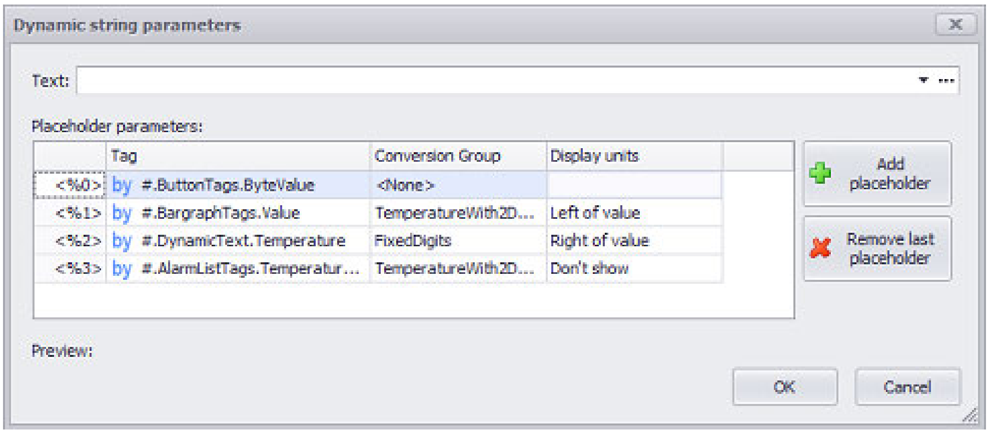
The Dynamic string parameters dialog box will open.

You can use the Text box to either enter a new text or select a text from a list of existing ones.

Clicking on Add placeholder will add a parameter set to which you can assign a tag you have created and a unit (Conversion Group). The Display units option can be adjusted individually for each placeholder.

For more information please refer to → "Working with the Text Editor ", page 640

Controls\Text\Properties\General/Text tab



Dynamic string parameters

Text:

Placeholder parameters:

	Tag	Conversion Group	Display units
<%0>	by #.ButtonTags.ByteValue	<None>	
<%1>	by #.BargraphTags.Value	TemperatureWith2D...	Left of value
<%2>	by #.DynamicText.Temperature	FixedDigits	Right of value
<%3>	by #.AlarmListTags.Temperatur...	TemperatureWith2D...	Don't show

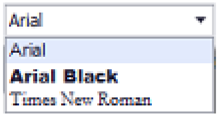
Figure 346: Dynamic string parameters dialog box after adding placeholders and making the necessary assignments

The icon color will be (x) to immediately show that the tags have been correctly assigned or (x) to indicate that an assignment is missing or invalid.

Text attributes - Font, Background, Word Wrap

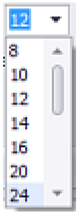
Font

With the exception of the <Arial> default font, the fonts available will be those that have been added in *Home\Languages\Project Languages*.

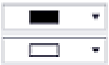


Font Size

The font size can be left with its default value or can be manually changed to a value within a range of 1 to 999 pt. The line height will be automatically adjusted. In addition, the text in the inserted text box will automatically be broken as required for the object's size if the automatic word wrap option has been enabled.



Color Selection

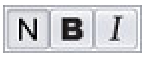


The font color and the color for the text box's background can be selected separately.

For detailed information on the available colors, please refer to the following section:
→ section "Color choice - Object's Color Palette", page 234

Font style N

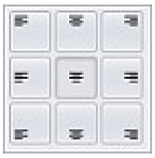
The following formatting options are available:



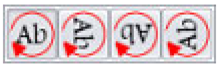
- Default (**N**ormal)
- **Bold** (**B**old)
- *Italic (**I**talic)*

Alignment

The buttons shown below can be used to configure the text to be left-aligned, centered, or right-aligned vertically and horizontally.



Text rotation



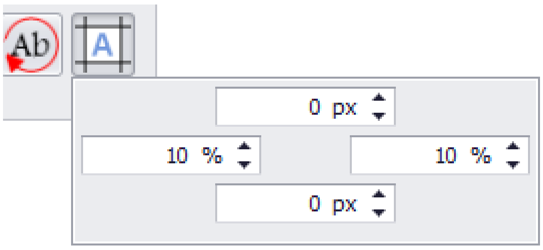
The four buttons shown above can be used to rotate the inserted text 90°, 180°, or 270° as needed.

This rotation is not part of the text style.

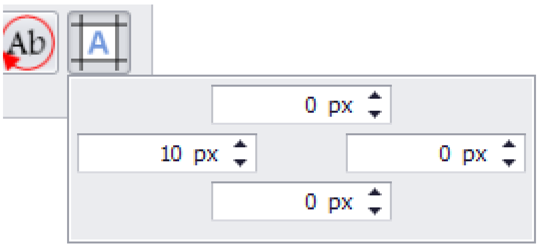
Distance from edge

Can be used to specify the distance from the visualization object edges that should be maintained.
The unit can either be pixels or a percentage (relative to the size of the object).

To select the latter, click on the unit. px The mode will change to "%".

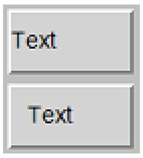


0 px
10 %
10 %
0 px



0 px
10 px
0 px
0 px

You can use these settings, for example, to define a left distance of 10 px for a left-aligned text so that the text will be indented 10 px on the object.


Text
Text

Background:

A transparent background will be selected for the text box by default. You can select a color instead if you want.

For detailed information on the available colors, please refer to the following section: → section "Color choice - Object's Color Palette", page 234

Word Wrap

☑ Turn on Enabling the checkbox

If this checkbox is enabled, the text will be broken into multiple lines in the text box if it is too long.

3.20 Video Player

3.20.1 Integrating videos

Please note that the video player visualization object is only available for devices with GALILEO OPEN, the XP500 and XV300 series, and the simulator.

The video player visualization object can be used to enable users to look for a video file and start, stop, and pause it as necessary.

The following formats are supported:

- *.mp4
- *.mov
- *.avi

→ Projects for the XV300 series only support the *.mp4 format.

3.20.2 Inserting an image box as a placeholder

→ Animated image sequences are supported only by XV300 and XP500 devices, as well as by GALILEO Open.

Settings

→ " Image", page 439

→ " Webcam", page 515

→ " Video Player", page 498

Used to insert an image box as a placeholder.

You can use the Properties 'Video player' pane to integrate the Video Player.

3.20.3 Video player

Only one single video player visualization object can be defined per screen.

If the video will be played back with GALILEO OPEN or the simulator, you will need to check whether a codec is needed for the corresponding PC. More specifically, a codec features both an encoder and a decoder that make it possible to compress and decompress video signals. If you try to use a video file that requires a codec that has not yet been installed, a system message will be output.

The following website, for example, has a variety of Win32 codecs available:

Codecguide.com/

Video player visualization objects can be used on standard screens, help screens, and background screens, but not on UserControls or containers.

The maximum resolution for all platforms is full HD (1920 x 1080 pixels).

The video player visualization object cannot be placed on transparent backgrounds.

When a video player visualization object is used, it will always be placed in the top layer on the device screen, i.e., with the exception of system keyboards and system messages, it will be superimposed on all other visualization objects. Accordingly, make sure not to place any visualization objects on top of the video player visualization object. If you do, they will be concealed at runtime.

The video player visualization object features several functions that enable operators to look for and play video files at runtime. All these functions can be triggered with function key visualization objects or with script functions.

When a user switches screens back to a screen with the video player object, the object will return to the same state in which it was when the user originally left the screen. If you want the same video file to always be shown after a screen change, you can achieve by using an entry script.

Audio output is supported, but requires corresponding hardware. Within this context, the AudioVolume system tag can be used to adjust the volume (the volume will be set to 100% by default).

The volume value can be adjusted with keyboard input or with a slider.

Video player on help screens

Using the video player visualization object on help screens requires a few special steps.

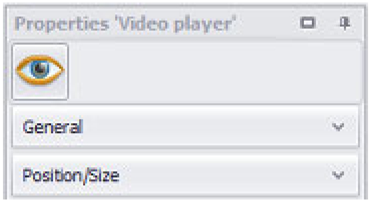
To place the video player visualization object on a help screen, start by selecting a help information visualization object. Then go to the properties for the help information object and, in the Information type field, select the <Video> option.

The video player visualization object will then be inserted into the help screen with the following fixed parameters:

- ☑ Loop video
- ☑ Autostart
- □ Menu enabled (it will not be possible to enable this checkbox)
- The background color for the video player visualization object will be the same as the help screen's background color

Prerequisites
You must have already added an → " Inserting an image box as a placeholder", page 498.
In order to be able to follow the descriptions below, you must first have clicked on a video player visualization object on the current screen.
There must be a video in the Project video folder in the project tree.

Controls\▶ tab

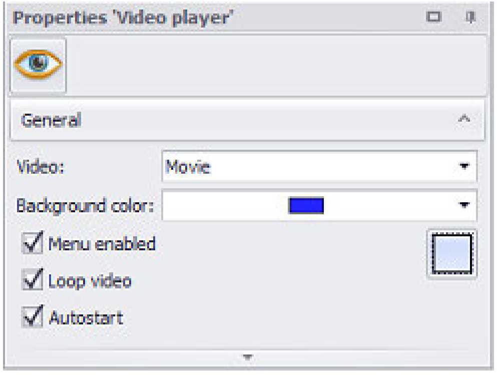


Properties 'Video player'	
👁	→ section "Visibility", page 259Visibility
	→ section "General", page 500
	→ section "Position/Size", page 242

3.20.3.1 General

The General section can be used to configure all the properties that affect how the visualization object will be displayed, with the exception of its size and position.

Controls\ Video player



Properties 'Video player'

General

Video: Movie

Background color:

☑ Menu enabled

☑ Loop video

☑ Autostart

Figure 347: Properties Video player pane, General section

Video

Clicking on the drop-down menu arrow will open a list containing the video files in the project tree.

If the list does not contain any videos, you will first need to import one or more video files into the *Media\Project video* folder in the project tree. To do so, follow the same basic steps you would in order to import an image into the project tree – please refer to → "Substep 1: Adding an image (medium) to the project tree", page 174 as well.

Background color

The background color you choose will be used to fill the video player visualization object and accordingly will serve as the background for the video.

You can change the color if you want. The following options are available for this purpose:

- Standard Colors
- More Colors...

The most recently used colors will be shown (user-specific).

For detailed information on the available colors, please refer to the following section: → section "Color choice - Object's Color Palette", page 234

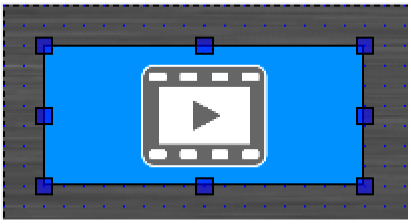
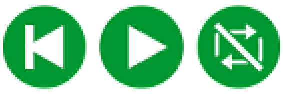


Figure 348: Video player with background

→ The "transparent" background color will not be available.

Menu enabled

If this checkbox is enabled, the following menu will be displayed for the device operator at runtime. This menu can be used to start, stop, and rewind the video.



Infinite loop

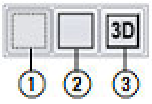
If this checkbox is enabled, the video will play in an endless loop.

Autostart

If this checkbox is enabled, the video will start as soon as the screen is shown on the device screen.

Frame 3D

The visualization object can be displayed with any one of three different designs.



① no frame

② with frame

③ 3D frame

3.20.4 Functions Video player

3.20.4.1 Video:Load

Only one video player visualization object is permitted per screen. This function can be used to load video files (*.MP4, *.AVI, *.MOV) in the video player visualization object. The name or the coordinates of the video player visualization object need to be passed as the first parameter, while the name of the video file needs to be passed as the second parameter. The video file will not be started automatically after being loaded – the Video:Start function needs to be called with the function keys for this purpose.

→ Not all device types support this function

→ If the function is run using a script, the referenced video object must be found on the screen that is currently being displayed. Otherwise, the function will not be run.

Number of the Function	5123(1403h)
Executable from the PLC	Yes
Function with ON/OFF state	No
Parameter 1	Name or coordinates of the video player visualization object
Parameter 2	Name of video file
Function available in standard screens	Yes
Function available with keyboards	No
Function available in script	Yes

Function available in application entry script	Yes

3.20.4.2 Video:LoadVar

Only one video player visualization object is permitted per screen. This function can be used to load video files (*.MP4, *.AVI, *.MOV) in the video player. The name or the coordinates of the video player visualization object need to be passed as the first parameter, while the name of the video file needs to be passed as the second parameter. The video file will not be started automatically after being loaded – the Video:Start function needs to be called with the function keys for this purpose.

→ If the function is run using a script, the referenced video object must be found on the screen that is currently being displayed. Otherwise, the function will not be run.

Number of the Function	5124(1404h)
Executable from the PLC	Yes
Function with ON/OFF state	No
Parameter1	Name or coordinates of the video player visualization object
Parameter2	Tag of data type STRING The tag should contain the name of the video file. If you want to use a specific storage location, you can include the path in the name. Tag content example: `"C:/temp/VideoSample.mp4"`
Function available in Standard screens	Yes
Function available with keyboards	No
Function available in script	Yes
Function available in application entry script	Yes
Remark	Not all device types support this function

3.20.4.3 Video:Pause

This function will pause the video player visualization object on the current screen.

→ If the function is run using a script, the referenced video object must be found on the screen that is currently being displayed. Otherwise, the function will not be run.

Number of the Function	5122(1402h)
Executable from the PLC	Yes
Function with ON/OFF state	No
Parameter1	Name or coordinates of the video player visualization object
Function available in Standard screens	Yes
Function available with keyboards	No
Function available in script	Yes
Function available in loop scripts	No
Function available in event scripts	Yes
Function available in application entry script	Yes
Remark	Not all device types support this function

3.20.4.4 Video:Select

Only one video player visualization object is permitted per screen. This function can be used to load video files (*.MP4, *.AVI, *.MOV) in the video player visualization object. The name or the coordinates of the video player visualization object need to be passed as the first parameter, while the directory for the video files needs to be passed as the second parameter. When the button is tapped at runtime, a system keyboard will appear and allow the device operator to select one of the video files.

The video file will not be started automatically after being loaded – the Video:Start function needs to be called with the function keys for this purpose.

Number of the Function	5125(1405h)
Executable from the PLC	Yes
Function with ON/OFF state	No
Parameter1	Name or coordinates of the video player visualization object
Parameter2	Name of the directory where the video files are stored. Example: `"C:/temp/"`
Function available in Standard screens	Yes
Function available with keyboards	No
Function available in script	Yes
Function available in application entry script	Yes
Remark	Not all device types support this function

3.20.4.5 Video:SelectVar

Only one video player visualization object is permitted per screen. This function can be used to load video files (*.MP4, *.AVI, *.MOV) in the video player visualization object. The name or the coordinates of the video player visualization object need to be passed as the first parameter, while a tag of data type STRING that contains the path to the video files needs to be passed as the second parameter. When the button is tapped at runtime, a system keyboard will appear and allow the device operator to select one of the video files in the directory.

The video file will not be started automatically after being loaded – the Video:Start function needs to be called with the function keys for this purpose.

→ If the function is run using a script, the referenced video object must be found on the screen that is currently being displayed. Otherwise, the function will not be run.

Number of the Function	5126(1406h)
Executable from the PLC	Yes
Function with ON/OFF state	No
Parameter1	Name or coordinates of the video player visualization object
Parameter2	Tag of data type STRING; The tag should contain the directory where the video files are stored. Example: `"C:/temp/"`
Function available in Standard screens	Yes

Function available with keyboards	No
Function available in script	Yes
Function available in application entry script	Yes
Remark	Not all device types support this function

3.20.4.6 Video:Start

This function starts the video player visualization object on the current screen.

→ If the function is run using a script, the referenced video object must be found on the screen that is currently being displayed. Otherwise, the function will not be run.

Number of the Function	5120(1400h)
Executable from the PLC	Yes
Function with ON/OFF state	No
Parameter1	Name or coordinates of the video player visualization object
Function available in Standard screens	Yes
Function available with keyboards	No
Function available in script	Yes
Function available in application entry script	Yes
Remark	Not all device types support this function

3.20.4.7 Video:Stop

This function stops the video player visualization object on the current screen.

→ If the function is run using a script, the referenced video object must be found on the screen that is currently being displayed. Otherwise, the function will not be run.

Number of the Function	5121(1401h)
Executable from the PLC	Yes
Function with ON/OFF state	No
Parameter1	Name or coordinates of the video player visualization object
Function available in Standard screens	Yes
Function available with keyboards	No
Function available in script	Yes
Function available in application entry script	Yes
Remark	Not all device types support this function

3.21 1.2 Value Display/Entry

In real-life applications, this visualization object is used as a simple tool for displaying a numeric tag or string tag or enabling a device operator to enter a value for it on the device.

The actual visualization object will be purely an output object unless it is configured with a keyboard. Using a keyboard will make it possible for device operators to enter a new value or a new tag of data type STRING instead of the one being shown.

This keyboard will be displayed on the PC after clicking on the visualization object or on the device after tapping the object.

There are no functions that can be applied to this visualization object.

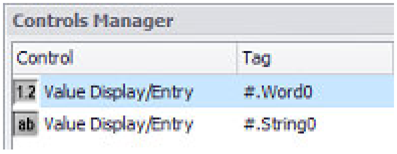
It is possible to define various states for this visualization object. This makes it possible to change the visualization object's appearance as soon as one of the tags' range limits is exceeded or fallen below. The corresponding input range will be limited by the defined minimum and maximum for the tag being used.

If this visualization object is used multiple times to display a large number of system components with the same parameter sets, it is recommended to use parameters. Using parameters will optimize your work when it comes to assigning tags to the visualization object.
For detailed information on how to work with parameters, please refer to → section "Using user controls to optimize work", page 279.

A detailed example showing how to use this visualization object is provided in section: → section "Your own first project", page 141.

The content of a "value display/entry" visualization object is configurable and depends on the assigned tag.
If a numeric tag is assigned, the corresponding properties will be shown.
If a string tag is assigned, the corresponding properties, e.g., for PLC texts, will be shown.

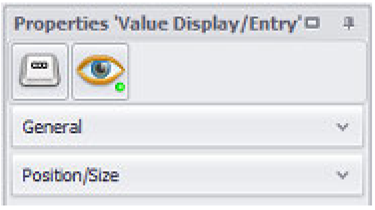
The Controls Manager and the message window both distinguish these objects based on the type of assigned tag.



Controls Manager

Control	Tag
1.2 Value Display/Entry	#.Word0
ab Value Display/Entry	#.String0

Controls\1.2 tab



Properties ''

⌨ → " States", page 513

✓

👁 → "General", page 507

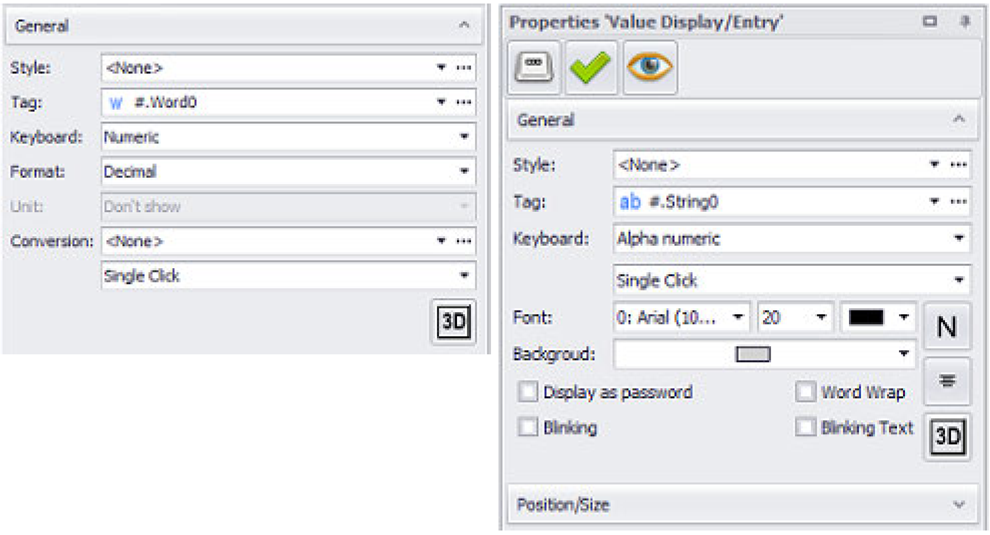
The ✓ accessibility button will only be shown in the Properties pane if a tag and a keyboard have been assigned to the visualization object, as it would not make sense to define accessibility conditions otherwise.

→ "Conversions", page 921
→ "Using user controls to optimize work", page 279
→ "Using function keys to trigger actions", page 389
→ "Keyboards", page 307

3.21.1 General

The properties will depend on the type of tag.

Controls\1.2 Value Display/Entry



General

Style: <None>
Tag: W #.Word0
Keyboard: Numeric
Format: Decimal
Unit: Don't show
Conversion: <None>
Single Click

Properties 'Value Display/Entry'

General

Style: <None>
Tag: ab #.String0
Keyboard: Alpha numeric
Single Click
Font: 0: Arial (10... 20
Backgroud:
□ Display as password □ Word Wrap
□ Blinking □ Blinking Text

Figure 349: Properties 'Value Display/Entry' pane with numeric tag on the left and string tag on the right

Style

Settings that affect the appearance of visualization objects can be bundled and stored as a style. After this, it will be possible to use the style's name to use these settings at any time and apply them to new visualization objects of the same type throughout the entire project. In addition, when changes are made to a style, the changes will automatically be applied to all visualization objects using the style, ensuring that all visualization objects using the same style will have a consistent appearance. For detailed information on how to create a style, please refer to → section "Style management", page 458.

Variable

The visualization object will show the value of the selected tag on the device screen at runtime. Clicking on the drop-down menu arrow will open a list of available tags (BIT, BYTE, WORD, DWORD, QWORD, FLOAT, STRING). Please note that in order for the tag you want to be shown in this list, the tag must first be imported or created in the Tags tab.

A parameter can also be specified instead of a tag. In this case, the Tag field will turn into a Type field. In order to be able to work with parameters, the visualization object must be placed on a UserControl. For detailed information on how to work with parameters, please refer to → "Using user controls to optimize work", page 279

Numeric values do not necessarily have to be displayed or entered the way the PLC needs them or the way they are stored in the tags. The Value Conversion Manager can be used to convert tag values and assign a variety of units to them before they are shown by visualization objects. The way value conversion works is described in greater detail in a different section – please refer to → section "Conversions", page 921.

Button ...

Limits

Clicking on the ... icon will show the limits for the value tag in a separate dialog box. You can also reach this dialog box via *Project Tree\Tags\<Value tag>\Properties*. When a tag is created, it will already be assigned limits based on the value range defined for it. This value range can be signed or unsigned and will depend on the relevant tag's data type.

The limits can be adjusted as necessary within the value range. In the case of bit tags, you will only be able to specify a default value.

For a description of the tag properties and limits, please refer to: → section "Numeric tag limits", page 563.

Tag Address

This field will only appear if you select a PLC tag, i.e., it will not appear for internal tags.
It will show the PLC address for the selected tag. The properties for PLC tags are described in greater detail in a different section – please refer to → section "Tag address", page 569.

If a single tag is transmitted to more than one communication, the corresponding settings will need to be configured in *Home\Gateway*. For detailed information on how to work with a gateway, please refer to → section "Using the gateway functionality to communicate with multiple PLCs", page 596.

Numeric tags only

Keyboard

Clicking on the drop-down menu arrow will open a list of available keyboards. If a keyboard is selected, the visualization object will turn into a touch zone at runtime and the person using the device will be able to enter or change a value. If a keyboard is not selected, the visualization object will be a display object without a touch zone.

A numeric system keyboard can be assigned to numeric tags.

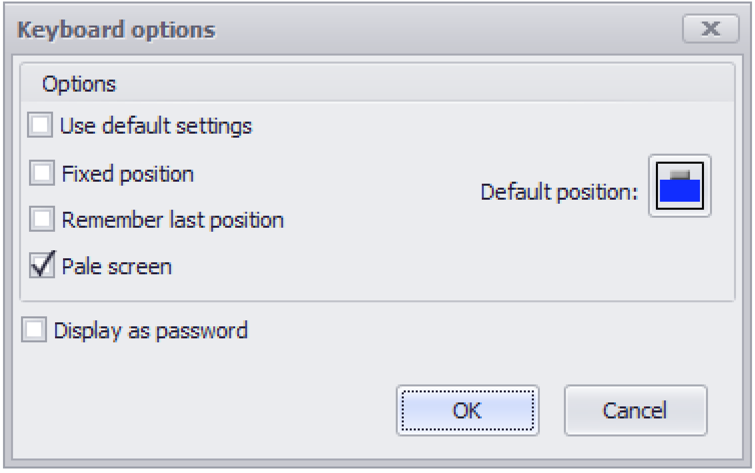
The following are the available numeric system keyboards:

- inc/dec 1
- inc/dec 10
- Numeric
- Numeric Digit Min/Max
- Numeric Min/Max
- Numeric Password
- Select Numeric
- Select Numeric Digit

If you select Select Numeric or Select Numeric Digit as a keyboard, it will be possible to adjust the value with a function from the Keyboards or Characters/Stream characters group, e.g., with the Inc/dec 1 function. For a description of how to use functions, please refer to the → "Controlling functions with the PLC", page 602 section.

To configure the display options for the keyboard, click on the ⚙ icon:

Controls\...\Properties\Keyboard⚙



Keyboard options

Options

☐ Use default settings

☐ Fixed position

☐ Remember last position

☑ Pale screen

Default position:

☐ Display as password

Figure 350: Keyboard options dialog box

Use default settings: If this option is enabled, the other options will be hidden and the project settings will be applied instead.

Fixed Position: If this option is enabled, an active keyboard cannot be moved around

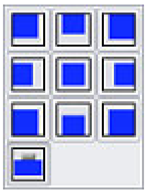
Remember last position: If this option is enabled, the keyboard's position will be saved when closed.
The next time the same keyboard is opened, it will be automatically moved to this saved position.

Gray out screen: If this option is enabled, the view behind the keyboard will be dimmed.

Default position: Used to specify the keyboard's default position relative to the object calling it.
The alignments available for selection include "top left," "center center," "bottom center," etc. plus "docked."
If you select the "docked" option, the Galileo runtime system will decide where the keyboard can best be displayed. If the "Remember last position" option is enabled, this setting will only be applied the very first time the keyboard is opened.

■ Default position

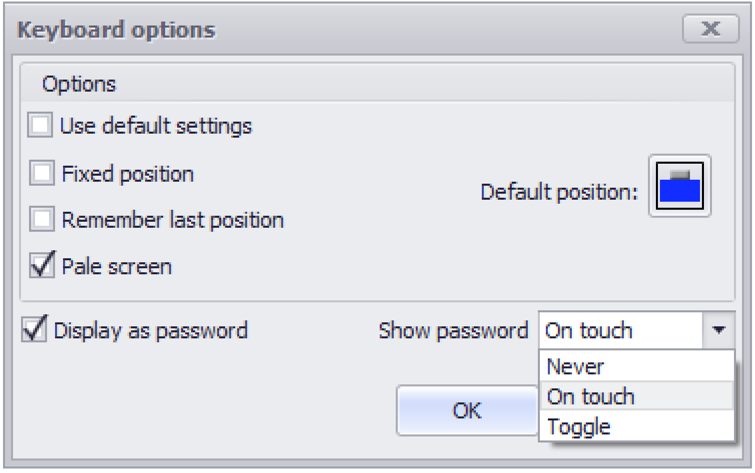
The keyboard will be positioned based on the option selected here. The default position on the screen can be defined by clicking inside the selection box as follows:



Top left ■, top center ■ or top right ■ Center left ■, center center ■ or center right ■ Bottom left ■, bottom center ■ or bottom right ■ or ■ Docked on the button

Display as password: If this option is enabled, only "****" will be displayed instead of the actual value.
The following additional options will be available when the option is enabled:

Controls\...\Properties\Keyboard



Keyboard options

Options

☐ Use default settings

☐ Fixed position

☐ Remember last position

☑ Pale screen

Default position: ■

☑ Display as password

Show password: On touch

Figure 351: Keyboard options dialog box

Never There will be no way to show the entered password in cleartext at runtime

On touch As soon as a number or letter is entered, an 👁 icon will appear in the input field. If this icon is tapped, the password will be shown in cleartext.

As soon as the 👁 icon is no longer being touched, the password will be shown as "****" again.

Toggle As soon as a number or letter is entered, an 👁 icon will appear in the input field. If this icon is tapped, the password will be shown in cleartext.
The password will continue to be shown in cleartext until the icon is tapped again.

→ A previously entered value cannot be shown in cleartext and will remain secret. Only current value input can be shown as cleartext.

For detailed information on how to put together keyboards made up of multiple individual function key visualization objects, please refer to the → section "User Defined Keyboards", page 316 section.

→ In order for this field to be enabled, a tag needs to be selected in the Tag field first.

Selection mode on panel Single Click

The following modes can be adjusted for visualization objects with a keyboard:

- Single Click
  Tapping once on a list item or a button on the device screen will activate the selection.
- Delayed
  This option is ideal for implementing a safety function designed to protect against accidental actions. In order for a button on the device screen to be activated, it will first have to be pressed and held (long press) for the configured delay time. By default, this delay time will be 0.5 s, but it can be changed in *Home\Settings\Runtime tab\Touch Settings*.
- Double Click
  Tapping twice on a list item or button on the device screen will select the item / activate the button. By default, the double-click speed will be 0.5 s for a double-click, but it can be changed in*Home\Settings\Runtime tab\Touch Setting*.

Format

This drop-down menu will only be available for tags of data type BYTE, WORD, DWORD, QWORD, FLOAT and is used to select a display format.

The following formats are supported:

Decimal	Numbers 0...9, Sign, msb is the sign
hexadecimal	Numbers 0...9, letters A..F

Octal	Numbers 0...7
Binary	Numbers 0 and 1
BCD	Numbers 0...9
hh:mm/mm:ss	Time in hours:minutes/minutes:seconds

Display limitations for hh:mm/mm:ss time format

This format can be used to enter and display values directly in time format, i.e., instead of a decimal point, a colon will be displayed, and only values within a range of 0 to 59 will be allowed after the colon. The time value will be converted to an appropriate value for the PLC without having to enter the corresponding formula. Example: "3:26" = 3 x 60 + 26 = 206.

Therefore arise the following Display-Limitations:

Tag data type	PLC Range	Display Range
BYTE	0...255	00:00...4:15
WORD	0...65'535	00:00...1'092:15
DWORD	0...4'294'967'295	00:00...71'582'788:15
FLOAT	not possible	not possible

Important consideration when using keyboards

When using keyboards of type Inc/dec1, "00:59" will be incremented to "1:00" and "1:00" will be decremented to "00:59," for example. Decimal point keyboards should not be used, as it will not be possible to change the value after the colon with them.

→ All value tags are always declared in decimal format regardless of the selected display format (decimal, hex, binary, octal, BCD)!

Custom keyboards can be used to implement numeric keypads for entering values in hex format.

Unit

Every value tag can be assigned a unit. Clicking on the drop-down menu arrow will open a list with the options available for positioning the unit.

- Don't show
- On the right
  The unit will be shown to the right of the value, separated by a space.
- On the top
  The unit will be shown above the value, in a separate line.

In order for this field to be enabled, a conversion group needs to be selected in the Conversion field first.

Conversion

Values do not necessarily have to be displayed or entered the way the PLC needs them or the way they are stored in the tags. The Value Conversion Manager can be used to convert tag values and assign a variety of units to them before they are shown by visualization objects. The way value conversion works is described in greater detail in a different section – please refer to → "Conversions", page 921.

For string tags only

Alphanumeric keyboards

For string tags, there are a number of alphanumeric system keyboards available for selection. These system keyboards can be expanded with custom keyboards. As soon as a keyboard is selected, the visualization object will become a touch zone.

The alphanumeric system keyboards are as follows:

- Alpha numeric
- Alphanumeric ALT
- Alphanumeric ALT CAPS
- Alphanumeric CAPS
- Select Alphanumeric

For a description of the font, orientation, and background color, please refer to section:.

Display as password

If the Display as password checkbox is enabled, the content of the tag of data type STRING will be shown as "*" instead throughout the entire project.

3.21.2 States

The following states are available for value display/entry objects: Lower Limit, In Range, and Upper Limit.

The basic configuration parameters for the visualization object's foreground and background are described in detail in the following section: Defining text series

< Lower Limit / in Range / > Upper Limit

The current value of the tag being displayed will be compared with the limits defined for the tag.

The visualization object will then be displayed as configured for the relevant state based on whether:

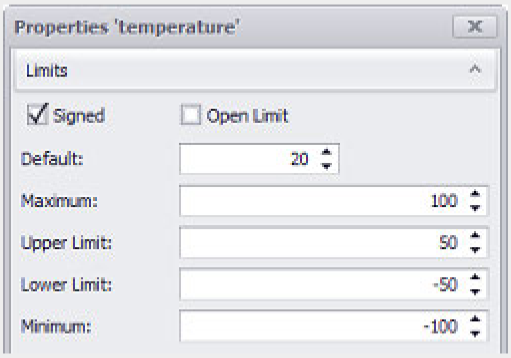
- The current value is less than (<) the lower limit
- The current value is within the range between the upper and lower limits
- The current value is greater than (>) the upper limit

The limits can be defined either by clicking on *Properties pane\General\Tag\*... or by using the *Properties pane* for the tag directly. If applicable, the conversion factors set in the Value Conversion Manager will be applied to all of the tag's limits, as well as to its current value.

The basic configuration parameters for the visualization object's foreground and background are described in detail in the following section: Defining text series

Example: Value display/entry color change when the corresponding value changes

- Add a value display/entry object to the start screen.
- Create a tag of data type BYTE called temperature and configure its settings as follows:



Properties 'temperature'

Limits

☑ Signed □ Open Limit

Default: 20

Maximum: 100

Upper Limit: 50

Lower Limit: -50

Minimum: -100

- Assign the temperature tag to the value display/entry object as shown in the Properties 'Value Display/Entry' screenshot. Do not forget to select a keyboard for entering values.
- Configure the states as shown in the → " Value Display/Entry", page 506 screenshot.
- Start the Galileo Simulator with function key F5 or by clicking on Home\ . Build.

As soon as the application starts, the green value display/entry object will appear with the default value of <20>. Click on the value display/entry object at runtime. The keyboard you selected will appear. Use it to enter a value of <51>. When you do this, the value display/entry object will change color to red. Now enter a value of <-51>. When you do this, the value display/entry object will change color to blue.

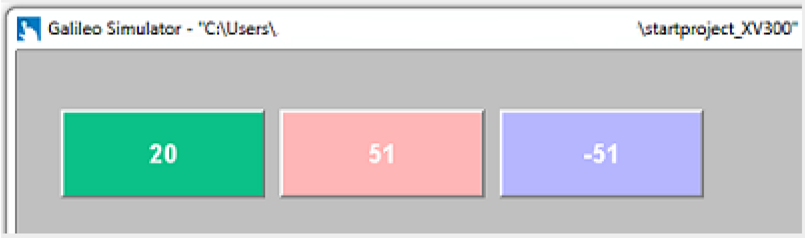


Figure 352: Device screen section showing three value display/entry visualization objects

3.22 Webcam

3.22.1 Integrating a webcam

The webcam visualization object is available on all devices, as well as in the simulator.

The webcam visualization object shows accessible live images from a webcam on the device screen.
To achieve this, the images are refreshed by downloading them from the relevant web address at a specific refresh rate, after which they are stored on the device and displayed.
Both the aforementioned refresh rate and the web address (URL) are defined in the properties for the visualization object (since the lowest available refresh rate is 100 ms, live streaming is not possible).
The target directory where the images should be stored is defined in the Project Settings.

→ Live streaming is not supported.
During live streaming, the images from a webcam are compressed so that the data transfer rate available on the Internet will be fast enough to transmit the images in real time.

The following image formats are supported:

- *.jpg
- *.bmp
- *.png

The images will be stored in the corresponding file format, in the target directory, the exact same way they are downloaded from the web address.

A single screen can show multiple webcam visualization objects at the same time.

The lower right corner of the visualization object will show the current status of the webcam at runtime with the ▶ and ⏸ icons.

Please note that it is not possible to control the webcam with gestures at runtime.

The webcam visualization object features several functions, including starting and stopping the webcam and capturing and saving snapshots (the file format for the snapshots will be the same file format that the webcam uses to store images at the relevant web address). Please keep in mind that images in BMP format will result in large file sizes and make image transmission slower. All the functions can be triggered with the function key visualization object or with the script function.

3.22.2 Inserting an image box as a placeholder

→ Animated image sequences are supported only by XV300 and XP500 devices, as well as by GALILEO Open.

Settings

→ " Image", page 439

→ " Webcam", page 515

→ " Video Player", page 498

Used to insert an image box as a placeholder.

You can use the Properties 'Webcam' pane to integrate a camera.

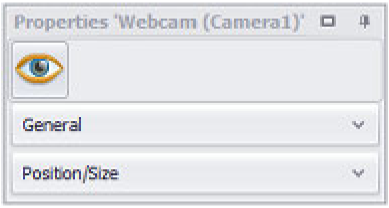
3.22.3 Webcam

Prerequisites

You must have already added an → " Inserting an image box as a placeholder", page 516 in the screen

In order to be able to follow the descriptions below, you must first have clicked on a webcam visualization object on the current screen.

Controls\ tab



Properties 'Webcam (Camera1)'

General

Position/Size

Properties 'Webcam'

Visibility

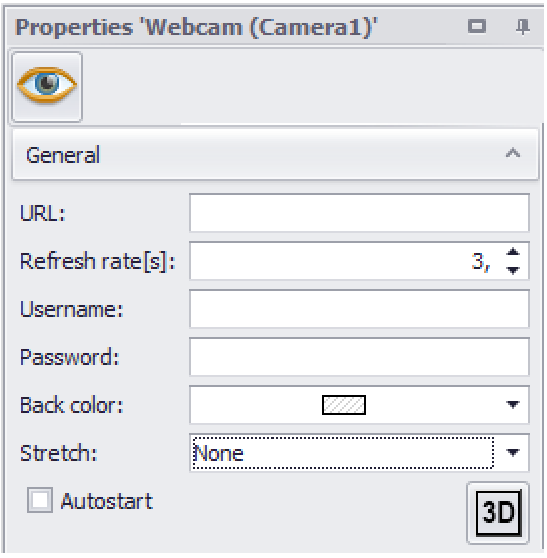
→ section "General", page 517

→ section "Position/Size", page 242

3.22.4 General

The General section can be used to configure all the properties that affect how the visualization object will be displayed, with the exception of its size and position.

Controls\ Webcam



Properties 'Webcam (Camera1)'

General

URL:

Refresh rate[s]: 3,

Username:

Password:

Back color: ▨

Stretch: None

□ Autostart

3D

Figure 353: Expanded properties pane for Webcam

URL

The URL field is used to enter the web address for the webcam capturing the images that the visualization object should show, e.g., "http://www.foto-webcam.eu/webcam/malcesine/current/400.jpg".

Please keep in mind that live streaming is not supported.

Refresh rate

The refresh rate is the interval at which images will be downloaded from the specified URL and shown in the webcam visualization object.

The lowest possible value is 0.1 s, while the highest possible value is 600 s.

Username

You can assign a username and a password. In this case, the application will prompt the user for both of them as soon as a function needs to be run on the webcam visualization object.

Password

You can assign a username and a password. In this case, the application will prompt the user for both of them as soon as a function needs to be run on the webcam visualization object.

Background color

The background color you choose will be used to fill the webcam visualization object and accordingly will serve as the background for the webcam object. If the size of the images downloaded from the webcam URL is smaller than the area occupied by the visualization object, the rest of the object will be filled with the background color.

You can change the color if you want. The following options are available for this purpose:

- Transparent
- Standard Colors
- More Colors...

The most recently used colors will be shown (user-specific).

For detailed information on the available colors, please refer to the following section:
→ section "Color choice - Object's Color Palette", page 234

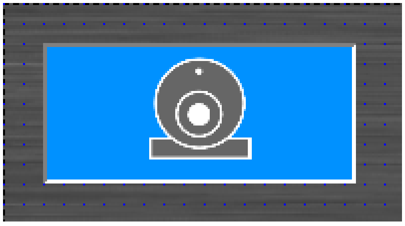


Figure 354: Webcam with background

Stretch

The following options are available:

- None
  The image will be aligned with the upper left corner of the area occupied by the visualization object. It will be inserted with its original size, i.e., if the image is larger than the area occupied by the visualization object, only part of the image will be displayed.

- Uniform
  The image will be centered in the area occupied by the visualization object. The aspect ratio will be maintained. In addition, the image will be resized to fit the area This means that if the image's aspect ratio does not match the aspect ratio of the visualization object's area, the selected background color will fill out the rest.
- Fill
  The image will be centered in the area occupied by the visualization object. In addition, the image will be resized to fit the area (the image's aspect ratio will not be maintained).

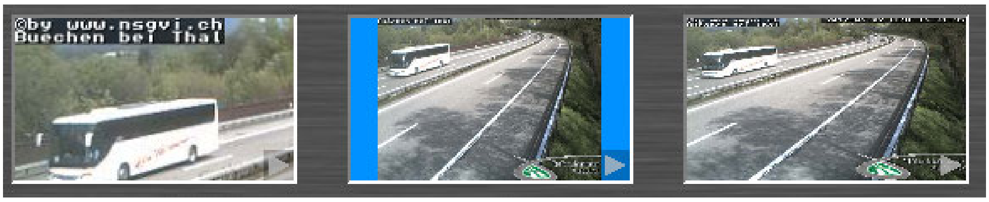


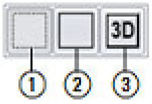
Figure 355: Stretch = None, Uniform, and Fill

Autostart

If this checkbox is enabled, the webcam will be started as soon as the screen is displayed on the device screen.

Frame 3D

The visualization object can be displayed with any one of three different designs.



(1) no frame

(2) with frame

(3) 3D frame

3.22.5 Functions Webcam

3.22.5.1 WebCam:Capture Image

This function saves the image currently being shown by the webcam visualization object on the screen.

The image will be saved with the name specified in parameter 2. If a file extension is included in the image name (e.g., <sample.png>), the file will not be converted. In other words, the image will be saved with the specified name, but will keep the file format in which it is provided by the webcam at the relevant web address regardless of whether the format matches the specified file extension or not.

→ If the function is run using a script, the referenced webcam object must be found on the screen that is currently being displayed. Otherwise, the function will not be run.

Number of the Function	4866(1302h)

Executable from the PLC	Yes
Function with ON/OFF state	No
Parameter 1	Name or coordinates of the webcam visualization object
Parameter 2	Name with which the image should be saved. Name example: "`Sample`" If you want to use a specific storage location, you can include the path in the name. Name example: "`C:/temp/Sample`"
Function available in standard screens	Yes
Function available with keyboards	No
Function available in script	Yes
Function available in application entry script	Yes

3.22.5.2 WebCam:Capture Image As

This function saves the image currently being shown by the webcam visualization object on the screen. A system keyboard will appear so that the device operator can enter the name they want for the image file.

The snapshot will then be saved with that name in the directory specified in parameter 2.

If a file extension is included in the image name (e.g., <Sample.png>), the file will not be converted. In other words, the image will be saved with the specified name, but will keep the file format in which it is provided by the webcam at the relevant web address regardless of whether the format matches the specified file extension or not.

→ If the function is run using a script, the referenced webcam object must be found on the screen that is currently being displayed. Otherwise, the function will not be run.

Number of the Function	4868(1304h)
Executable from the PLC	Yes
Function with ON/OFF state	No
Parameter 1	Name or coordinates of the webcam visualization object
Parameter 2	The path where the image should be saved. Example: "`C:/temp/`"
Function available in standard screens	Yes
Function available with keyboards	No
Function available in script	Yes
Function available in application entry script	Yes

3.22.5.3 WebCam:Capture Image Var

This function saves the image currently being shown by the webcam visualization object on the screen.

The image will be saved with the name specified in the parameter 2 tag.

If a file extension is included in the image name (e.g., <Sample.png>), the file will not be converted. In other words, the image will be saved with the specified name, but will keep the file format in which it is provided by the webcam at the relevant web address regardless of whether the format matches the specified file extension or not.

→ If the function is run using a script, the referenced webcam object must be found on the screen that is currently being displayed. Otherwise, the function will not be run.

Number of the Function	4867(1303h)
Executable from the PLC	Yes
Function with ON/OFF state	No
Parameter 1	Name or coordinates of the webcam visualization object
Parameter 2	Tag of data type STRING; The tag should contain the name with which the image should be saved. Tag content example: `"Sample"` If you want to use a specific storage location, you can include the path in the name. Tag content example: `"C:/temp/Sample"`
Function available in standard screens	Yes
Function available with keyboards	No
Function available in script	Yes
Function available in application entry script	Yes

3.22.5.4 WebCam:Capture Image Var As

This function saves the image currently being shown by the webcam visualization object on the screen. A system keyboard will appear so that the device operator can enter the name they want for the image file.

The snapshot will then be saved with that name in the directory specified in parameter 2.

If a file extension is included in the image name (e.g., <Sample.png>), the file will not be converted. In other words, the image will be saved with the specified name, but will keep the file format in which it is provided by the webcam at the relevant web address regardless of whether the format matches the specified file extension or not.

→ If the function is run using a script, the referenced webcam object must be found on the screen that is currently being displayed. Otherwise, the function will not be run.

Number of the Function	5120(1400h)
Executable from the PLC	Yes
Function with ON/OFF state	No
Parameter 1	Name or coordinates of the webcam visualization object
Parameter 2	Tag of data type STRING; The tag should contain the path where the image should be saved. Example: `"C:/temp/"`
Function available in standard screens	Yes
Function available with keyboards	No

Function available in script	Yes
Function available in application entry script	Yes

3.22.5.5 WebCam:Start

This function starts the webcam visualization object on the current screen.

→ If the function is run using a script, the referenced webcam object must be found on the screen that is currently being displayed. Otherwise, the function will not be run.

Number of the Function	4864(1300h)
Executable from the PLC	Yes
Function with ON/OFF state	No
Parameter 1	Name or coordinates of the webcam visualization object
Function available in standard screens	Yes
Function available with keyboards	No
Function available in script	Yes
Function available in application entry script	Yes

3.22.5.6 WebCam:Stop

This function stops the webcam visualization object on the current screen.

→ If the function is run using a script, the referenced webcam object must be found on the screen that is currently being displayed. Otherwise, the function will not be run.

Number of the Function	4865(1301h)
Executable from the PLC	Yes
Function with ON/OFF state	No
Parameter 1	Name or coordinates of the webcam visualization object
Function available in standard screens	Yes
Function available with keyboards	No
Function available in script	Yes
Function available in application entry script	Yes

3.23 Working with the Controls Manager

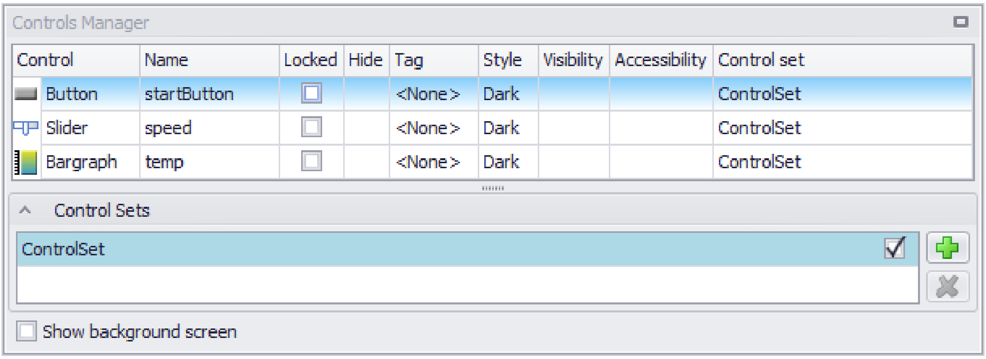
The Controls Manager is the perfect tool for getting a quick overview of the objects and functions being used on a screen. The objects and functions in the topmost layers on the screen will likewise be shown at the top of the Controls Manager pane, meaning that the order of these elements will be reflected in their vertical position. Please note that this approach is different from the layers used in GALILEO 8, and that these layers are no longer supported starting with this version. If you import a GALILEO 8 project, the corresponding layers will be converted into object sets.

Object sets can be used to group multiple objects together and show and hide them with a single click.

The Controls Manager is especially useful when multiple objects are on top of each other on a screen. It can be used to hide the objects in the upper layers so that you can select and modify the objects underneath. Moreover, the vertical order of the various visualization objects can be quickly changed by dragging and dropping the lines within the Controls Manager.

Likewise, the order in which the columns are shown can also be changed by simply dragging and dropping the columns.

Controls



Controls Manager

Control	Name	Locked	Hide	Tag	Style	Visibility	Accessibility	Control set
Button	startButton	□		<None>	Dark			ControlSet
Slider	speed	□		<None>	Dark			ControlSet
Bargraph	temp	□		<None>	Dark			ControlSet

Control Sets

ControlSet ☑

□ Show background screen

Figure 356: Controls Manager pane with three visualization objects

→ To select multiple entries in the Controls Manager, hold down the Shift or Ctrl key while clicking on the entries.

By default, the Controls Manager will show three columns.
However, this default view can be expanded with additional selections from the context menu. To access this menu, right-click on the header line, e.g., on <Control>, and select the column you want to display.

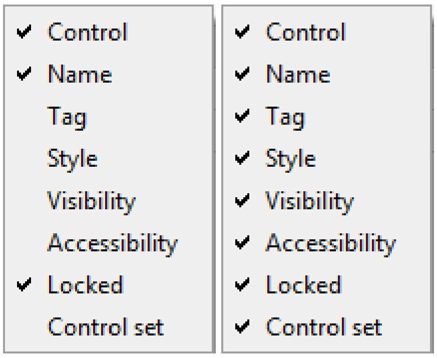


Figure 357: Controls Manager context menu

Name and Style are pure text fields. You can use these fields to assign a unique name to the visualization object and to the style.

The settings for the Controls Manager will be saved continuously. This means that when you start GALILEO again, the Controls Manager will look the exact same way it did before the application was closed.

The Controls Manager offers the following functions:

Filtering by object types

If you move the cursor over the Control column in the Controls Manager pane, the filter icon will appear. You can click on this icon in order to filter the pane by visualization object in a targeted manner.

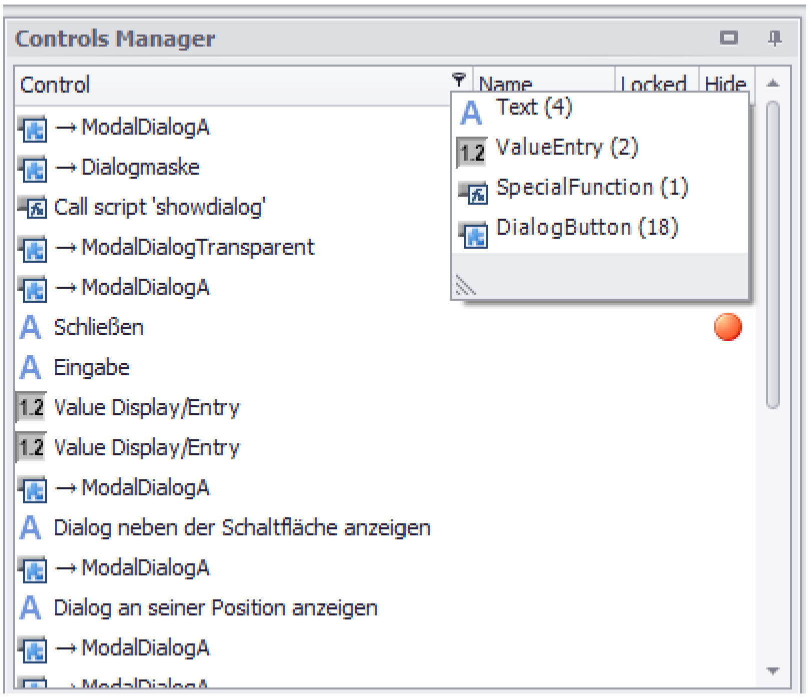
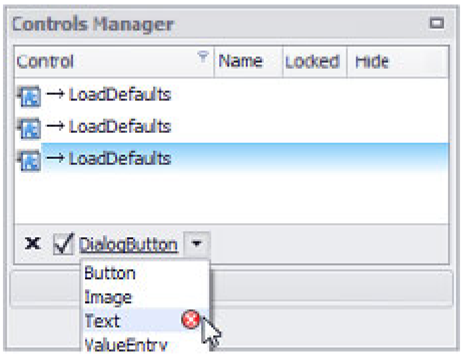


Figure 358: Example showing the Controls Manager pane filter function being used to filter by specific object types

Set filters will be displayed in the status bar, where they can also be reset.

☒ Resets all filters

☑ Resets an individual filter



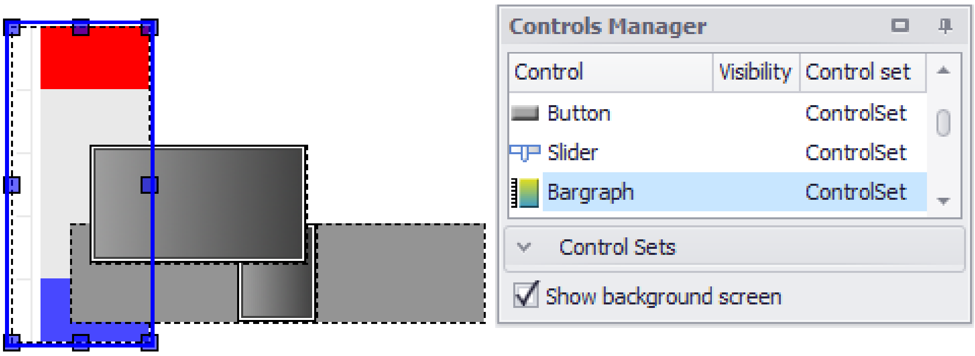
Controls Manager

Control	Name	Locked	Hide
→ LoadDefaults			
→ LoadDefaults			
→ LoadDefaults			

Figure 359: Example showing the Controls Manager pane filter function being used to filter by specific object types

Moving the cursor over object names

When the cursor is moved over an object name in the Controls Manager, the corresponding object will be highlighted with a color frame on the screen. Likewise, clicking on an object on the screen will cause the corresponding object name to be highlighted in the Controls Manager.



Controls Manager

Control	Visibility	Control set
Button		ControlSet
Slider		ControlSet
Bargraph		ControlSet

Figure 360: In this example, there are three objects on top of each other: Bargraph at the bottom, then Slider, then Button

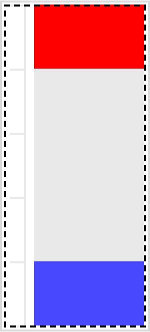
Hide

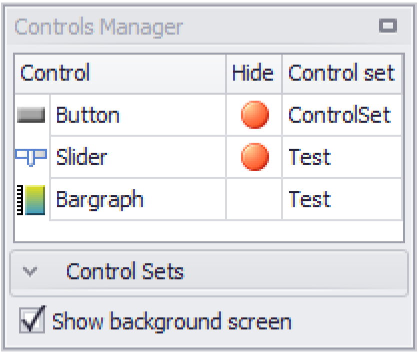
Clicking on the row for an object in the Controls Manager under the Hide column will hide the corresponding object on the screen. Clicking on it again will show the object.

If you want to hide all the objects above a certain row in the Controls Manager with a single click, use the <Hide overlying> option in the context menu.

If you want to show multiple objects in the Controls Manager, select the corresponding rows and use the <Show all> option in the context menu.

→ This function will not be available if multiple objects are selected.





Controls Manager

Control	Hide	Control set
Button	●	ControlSet
Slider	●	Test
Bargraph		Test

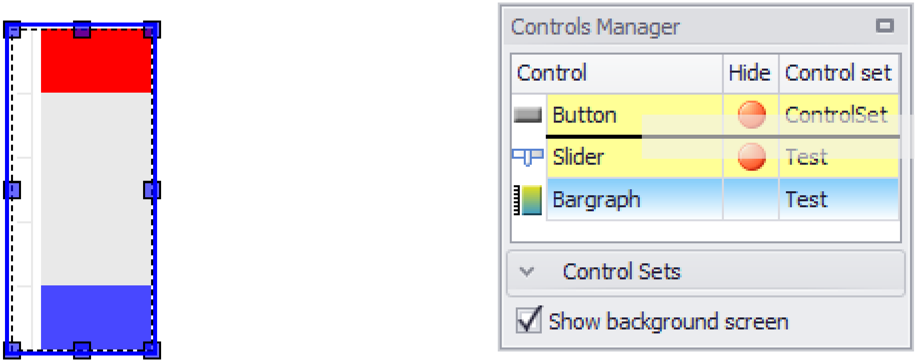
Control Sets

☑ Show background screen

Object order

The order of the objects in the Controls Manager can be changed. Simply drag the object names to the position you want in the Controls Manager. A special line will be shown to indicate where the object will be placed as you drag it, and the changes you make to the order object will apply to the screen immediately.

When an object name is highlighted in yellow or red, this means that the object being dragged will either cover another object or be covered by another object if it is dropped at the current position. The application setting configured in *Home\Style* will determine whether object names are highlighted in yellow or red.



Controls Manager

Control	Hide	Control set
Button	●	ControlSet
Slider	●	Test
Bargraph		Test

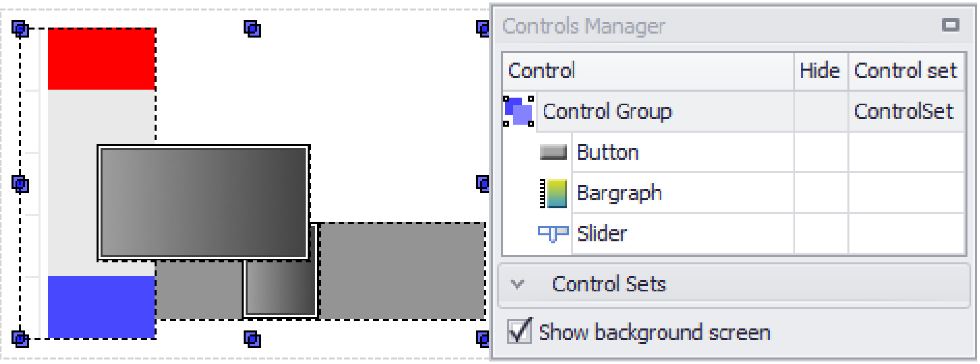
Control Sets

☑ Show background screen

Figure 361: Bargraph object being dragged between Button and Slider.

Grouping objects

You can group objects together in the Controls Manager by selecting multiple objects and then using the Group context menu option. Double-clicking on a group in the Controls Manager will show all the objects that fall under that group.



Controls Manager

Control	Hide	Control set
Control Group		ControlSet
Button		
Bargraph		
Slider		

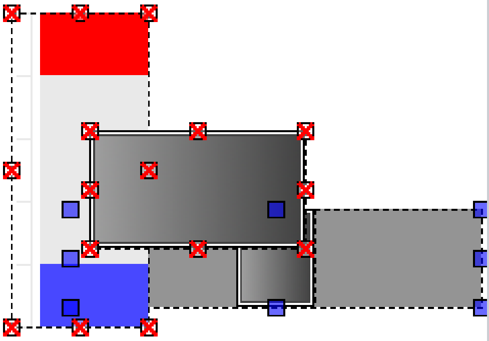
Control Sets

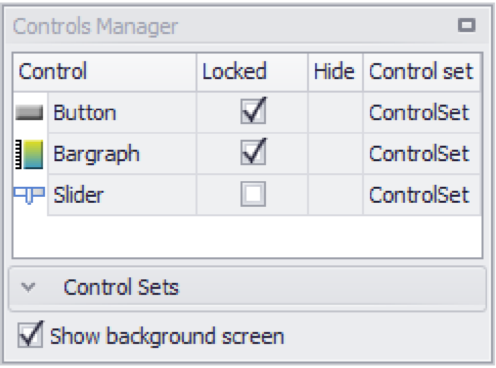
☑ Show background screen

Locked

Enabling the Locked checkbox for an object in the Controls Manager will lock the object in place, making it impossible to move the object on the screen with the mouse or with the arrow keys.

→ If an object that is locked in place is grouped with other objects, the entire group will be locked in place.





Controls Manager

Control	Locked	Hide	Control set
Button	☑		ControlSet
Bargraph	☑		ControlSet
Slider	☐		ControlSet

Control Sets

☑ Show background screen

Unlike the regular selection function, the selection of a Locked Control is indicated differently.

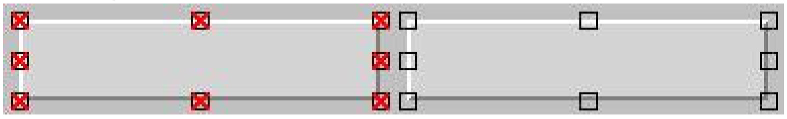


Figure 362: How a locked button and a non-locked button look when clicked on

Name

To be able to distinguish between multiple objects of the same type on a single screen in the Controls Manager, you can assign names to the objects. These objects will appear and be editable in the Controls Manager and in *Properties\<dynamic control>\Position/Size*.

Creating object sets

Object sets can be used to group multiple objects together and show and hide them with a single click. Each screen will have at least one object set in the Controls Manager.

Object sets are the equivalent of the "layers" used in Galileo 8. If you import a GALILEO 8 project, the layers used in it will be converted into object sets.

Each object can only be assigned to one single object set.

Up to 65,535 object sets can be defined, with eachobject set containing up to a maximum of 65,535 objects.

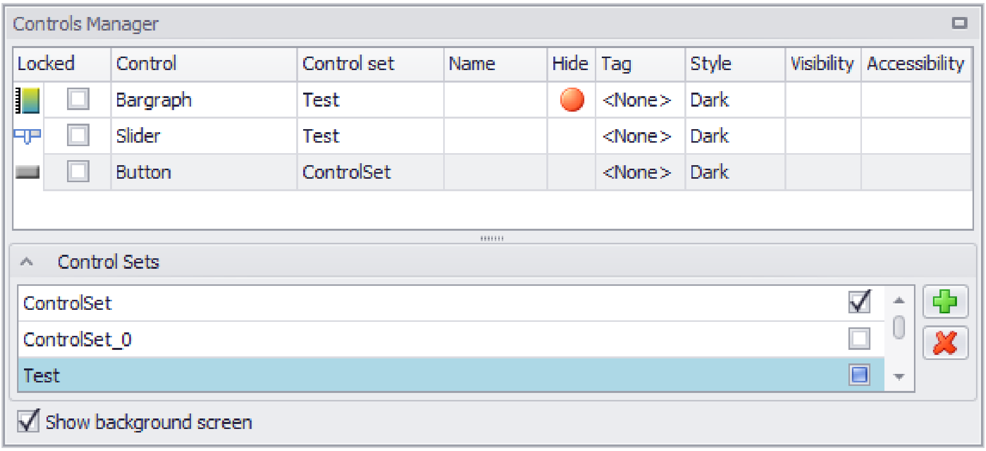
The list of object sets will be sorted in alphabetical order.

When the name of an object set is highlighted in blue, this means that the set is selected.

In contrast to a group, the objects belonging to an object set can be positioned and resized independently of each other. If a group is added to an object set, all the objects in the group will belong to that object set.

To create an object set, follow the steps below:

- ▶ Click on ➕ to create a new object set.
- ▶ Double-click on the name assigned by default and enter a name of your choice.
- ▶ Select the objects that you want to add to the object set.
- ▶ Right-click to open the context menu and click on the <Move to object set> option and then on <Name>.
- ▶ If an object set is selected (i.e., highlighted in blue), all new objects added to the screen will automatically be assigned to that object set.



Locked		Control	Control set	Name	Hide	Tag	Style	Visibility	Accessibility
	□	Bargraph	Test		●	<None>	Dark		
	□	Slider	Test			<None>	Dark		
	□	Button	ControlSet			<None>	Dark		

ControlSet	☑
ControlSet_0	□
Test	▣

☑ Show background screen

Figure 363: Two objects are assigned to the Test object set and one object is assigned to the ControlSet object set

If the objects in an object set are copied from one screen to another, they will be assigned to the object set of the same name in the new screen (provided there is one). If the new screen does not contain an object set with the same name, the objects will instead be assigned to the object set for which the checkbox is currently enabled.

Object sets can assume any of the three following states:

- ☑ All the objects in the object set will be shown
- □ All the objects in the object set will be hidden
- ▣ Some of the objects in the object set will be shown/hidden

Assigning objects to an object set

There are several ways to assign an object to an object set:

Select the object set first

- ▶ Select the object set you want in the Controls Manager, e.g., <Test>.
- ▶ Create a new object on the screen. The new object will be automatically assigned to the selected object set.

Drag&drop

- Select the object you want in the Controls Manager, e.g., <Button>. You can select multiple objects if necessary.
- Drag and drop the object onto the object set you want in the Controls Manager.

Context Menu

- Select the object you want in the Controls Manager, e.g., <Button>. You can select multiple objects if necessary.
- Right-click to open the context menu and click on the <Move to object set> option and then on <Name>.

Objects can be assigned to a different object set, but they cannot be unassigned from object sets entirely. The reason for this is that every screen has at least one object set and the first object created on the screen will automatically be assigned to that object set from the beginning

Importing layers as object sets

The layers used in GALILEO 8 are no longer supported in GALILEO. Accordingly, when a GALILEO 8 project is imported, any layers in it will be converted to object sets. For instance: If a project with 15 objects distributed among layers 1 to 3 is imported, three object sets called Layer 1, Layer 2, and Layer 3 will be created and the corresponding objects will be assigned to them.

Please note that if a group of objects is imported and the objects are distributed among several layers, the group will be kept after the import. However, only one object set will be created, and the name assigned to it will be the name of the layer containing the most objects. Moreover, this object set will contain all the objects in the group.

4. Variables /Tags

Using tags and data types

In GALILEO, all tags are created and managed in *Project Tree* *Tags*. Tags can be categorized based on their data type and based on the basic group to which they belong. There are four basic groups within this context:

1. System Tags
2. Internal Tags
3. PLC-Tags
4. Tags and web visualization

System Tags

System Tagsare identified with the sy data type and are a required component of every GALILEO project. These tags can be found in *Project Tree* *Tags* *$: System Tags*.
System tags contain device information such as <HardwareVersion>.
This system tag makes it possible to configure the behavior of a GALILEO application differently for web clients and for the local application on the device.
Please note that system tags can be read, but cannot be written to or deleted.
The only exceptions are the sy Control, sy Status, and syDate_Time system tags, which play an important role in data communications.

For a description of system tags, please refer to the → section "System tags", page 577 section.

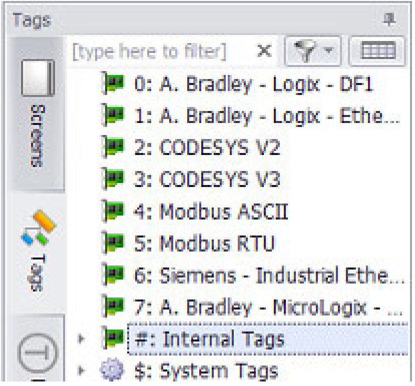
Internal Tags

These tags are automatically placed in *Project Tree\Tags\* *#:Internal Tags*. Addresses cannot be assigned to internal tags, and these tags cannot be transmitted either. Instead, internal tags are used exclusively within GALILEO as auxiliary tags. They are used for GALILEO application examples without PLC communications for testing purposes.

PLC Tags

PLC tags are data points in the Automatization Software program that are linked to a PLC via the addresses assigned to them. Tags can be used to transfer values between various machine controllers and the device using GALILEO. A maximum of eight PLC communication systems can be added to and configured in a GALILEO project. As soon as a PLC communication system is added to a project, GALILEO will automatically create a communication node in*Project Tree\Tags*. This communication node will contain the communication profile for all the PLC tags used for the corresponding communication system. In order to ensure that data will be successfully synchronized, addresses are assigned to the various tags in the tags' properties. It is possible to transmit tags to multiple PLCs. The gateway functionality provides the functions required for this purpose.

Project Organization\Tags



0: A. Bradley - Logix - DF1
1: A. Bradley - Logix - Ethe...
2: CODESYS V2
3: CODESYS V3
4: Modbus ASCII
5: Modbus RTU
6: Siemens - Industrial Ethe...
7: A. Bradley - MicroLogix - ...
#: Internal Tags
$: System Tags

Figure 364: Section from Project Tree\Tags with eight communication systems set up

Values for internal tags and PLC tags can both be entered and displayed. To this end, there are a variety of visualization objects that are linked to tags and that can be used to display or enter values. In fact, certain objects can be used for both. The "Data communication principles" section describes how the device exchanges tag data with the various communication systems. Accordingly, make sure to familiarize yourself with the various data types used for tags in GALILEO and the aforementioned data communication principles.
For a description of how to create tags, please refer to the → section "Working with tags", page 539 section.

Tags and web visualization

All the PLC tags for a communication system are shared. Within the context of the web visualization functionality, "shared" means that when a PLC tag is modified in a web client, the change will take effect immediately on all other web clients as well.

System tags are also shared.

In contrast, internal tags are not shared by default. If you do want to share them, you will need to access the properties for the internal tags and enable the Shared option. In order for this option to be available, go to *Home\Settings\WebVisu tab* and select <Web only> or <Web and local> as the visualization mode.

The following applies to structures and arrays: If you share a structure or an array, all the elements in the structure or array will be shared at first. However, you can enable or disable sharing for individual elements afterwards.

4.1 Tags and Data Types

This section goes over the tags' categorization based on their data type. It covers both internal tags and PLC tags. System tags are not included, as they are already assigned to a set data type.

Variables

Every visualization object in a screen and every tag object in a project that processes data uses tags. These tags represent a value or bit in the corresponding PLC. Moreover, every tag belongs to a specific data type. An address can be assigned to tags, including individual tags, tag arrays, and structures. Depending on the PLC used, tags, arrays, and structures to which an address is assigned must have a specific data size. It is also possible not to assign an address to a tag, in which case the tag will only be used internally in the device and will not be processed by the PLC. If, in contrast, the tag is assigned an address, it will be considered a PLC tag immediately. For a description of how addresses are assigned, please refer to the → section "Assigning addresses to PLC tags", page 535 section. Depending on the corresponding data direction, these tags will be read from the PLC, sent to the PLC, or both.

For a description of what the data direction does, please refer to the → section " Direction of data", page 537 section

GALILEO supports the following standard tags, which are instances of primitive data types:

- x Bit
- by Byte
- w Word
- dwDouble Word (DWORD)
- qw Quad Word (QWORD)
- f Float
- d Double
- ab String
- e Error

GALILEO supports the following composite tags:

- st Structures - these are groups of tags with a defined order

Structures

Structures are used to group tags of different data types with the same data direction. A structure is made up of multiple structure elements, with each structure element being a tag with a defined data type. The order in which the tags are defined in a structure will also be the order in which they are mapped in the corresponding

PLC's address space. This order can be changed within the structure by dragging& elements with the mouse and dropping them in the position you want.

Instances of custom data types

If you create custom data types, you will be able to create instances of them within the "Structure" data type.

The same requirements apply to addressing these as for Structures.

In the case of instances of custom data types, you will only be able to configure the limits, default value, and sign for the individual tags. All other properties (including the display format) will be defined by the custom data type, and it will only be possible to change them directly there.

Data Types

Data types are, in a manner of speaking, a type of "formatting" for tags. Individual tags are instances of a data type. In other words, they follow the rules of that data type: when a tag is created, the amount of storage space reserved for it and the way its individual bits are interpreted will be based on the "settings" for the data type to which it belongs.

GALILEO supports the following primitive data types:

- x Bit
- by Byte
- w Word
- dw Double Word (DWORD)
- qw Quad Word (QWORD)
- f Float
- d Double
- ab String
- e Error

GALILEO supports the following composite data types:

- T User defined Data Types

4.1.1 What the individual data types are used for

Bit

Tags of data type bit are used to display and/or set a binary state. This type of state can characterize a switch, button, or light, for example.

Numeric tags

Tags of data type BYTE, WORT, DWORD, QWORD, DOUBLE and FLOAT are also referred to as "numeric tags," and are used to hold current values. All numeric values share the same properties that can be configured, such as limits, display formats, etc. For a detailed description of these properties, please refer to → section " Tags", page 563.

Not all PLC communications support 64bit tags.

String

String tags can be used to transmit texts in the form of a string to a PLC. They can also be used, for example, to display the current username or recipe entry. String tags can be displayed using the PLC texts visualization object.

For a description of the properties of string tags, such as limits, display formats, etc., please refer to the → section "String tag limits ", page 567 section.

Errors

The alarm management function is based on the use of error tags, i.e., tags of data type error or error array. The error data type is a bit tag, while the error array type is an array of bits. When an error tag has a state of 1, it is signaling an error or fault.

For a detailed description of error tags, please refer to → section " Error Tags", page 877.

Custom data types

GALILEO supports the use of custom data types. Within this context, a "custom data type" is a structure defined by a user – in fact, one way to think of this data type is as a sort of construction manual for a structure. While a tag of data type structure is put together with a number of tags of other data types when it is created, a custom data type is instead a ready-made template for the tag structure that will be used to create instances. This means that custom data types constitute a simplified way to create a large number of similar tags of data type structure.

Tags of the following data types are allowed for use as structure elements in a custom data type:

- BIT, BIT Array
- BYTE, BYTE Array, WORD, WORD Array, DWORD, DWORD Array, QWORD, QWORD Array, FLOAT, FLOAT Array, DOUBLE, DOUBLE Array
- ERROR, ERROR Array,
- STRING, STRING Array
- References to other user defined data types

The various properties (display format, limits, units, conversions) can be configured for the individual tag types in a custom data type. These properties will then be applied to all instances of the data type. Please note that addresses can only be assigned to the actual instances.

4.2 Assigning addresses to PLC tags

Before you start assigning addresses to tags, make sure to familiarize yourself with how data communication between the device and the PLCs works – please refer to → "Data communication principles", page 580

The following properties should be noted:

- Addresses can only be assigned to tags if the tags have been created in the folder for the corresponding PLC communication system in *Project Tree\Tags*.
- An address can be assigned to every tag, every array, and every structure using its properties.
- Addresses cannot be assigned to internal tags.
- Tags that are not addressed do not communicate with the PLC.
- Arrays, structures, and custom data type tags can be assigned an address that applies to them as a whole. The addresses for the elements within the array or structure will then be automatically generated.
  GALILEO users must make sure that all tags have valid addresses!
- Arrays can only contain tags of the same type. All array elements have the same preset value and the same limit values.
- If you want the device to read tags from a PLC and write to other PLCs, you will need to configure the gateway functionality accordingly.
- For a description of the gateway functionality, please refer to the following section: → section "Using the gateway functionality to communicate with multiple PLCs", page 596.
- If a structure is addressed, this can only contain tags with the same data direction, otherwise the entire structure will always exchange data.

4.2.1 Assigning addresses to bit, numeric, and string tags

In order for an address to be assigned to a tag, the tag needs to be created in or moved to one of the communication nodes in *Project Tree\Tags*. As soon as a tag is located under one of these communication nodes, e.g., <0:CODESYS V3>, it will be assigned to the corresponding communication system. You will then need to check the tag's properties to make sure that the address is active (the tag will not be ready for communication until it is active).

→ Tags with an address will be shown in green (X). Tags to which an address has not been assigned will be shown in blue (X).

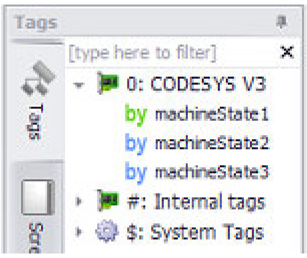


Figure 365: Section from *Project Tree\Tags* with <machineState1> as a tag with an address for the CoDeSys V3 PLC communication system and <machineState2> and <machineState3> as tags to which an address has not been assigned

4.2.2 Addressing structures

Structures can be assigned an address in one of two ways: by directly assigning the structure an address or by directly assigning an address to every individual tag within the structure.

Directly assigning the structure an address

A structure can be assigned an address directly. When this is done, all the tags within the structure will be automatically assigned addresses in consecutive order. The calculated individual addresses will depend on the structure's data structure alignment.

Directly addressing each individual tag inside the structure

However, each tag can also be addressed separately. The structure is used as a design aid in order to improve the clarity of the project.

The rules that apply when assigning addresses to structures apply when assigning addresses to custom data types as well. For more information, please refer to → "Data structure alignment", page 618.

Assigning an address with the context menu

The context menu can be used to directly assign an address to structures or data type instances by selecting the <Address tags directly> option.

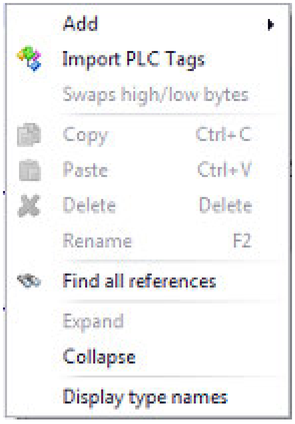


Figure 366: Context menu for tags in structures or data type instances

4.2.3 Assigning addresses to error tags

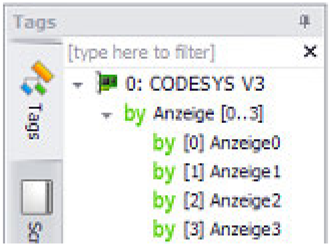
Error tags are only read from the PLC and accordingly have the 'Read (Device <- PLC)' data direction. For more information on how error tags are read from the PLC using the Control data block, please refer to the → section "Control structure", page 603 section.

4.2.4 Assigning an address to arrays

When an address is assigned to an array, it is always assigned to the array as a whole. The individual array elements will then be assigned addresses automatically.

→ Tags with an address will be shown in green (X). Tags to which an address has not been assigned will be shown in blue (X).

In the following example, the Display array is assigned an address as a whole. Individual array elements Display0 to Display3 are then assigned addresses automatically.



Tags
[type here to filter]
0: CODESYS V3
by Anzeige [0..3]
by [0] Anzeige0
by [1] Anzeige1
by [2] Anzeige2
by [3] Anzeige3

Figure 367: Section from *Project Tree\Tags* with BYTE array <Display> as a tag with an address for PLC system CODESYSV3

What will happen if the device operator enters a new value at runtime, e.g., for Display1?

Only the Display1 array element will be written from the device to the PLC.

4.2.5 Direction of data

There are several reasons why it is advisable to specify a data direction. For instance, it is possible for there to be cases in which tags need to be displayed on the device but should not be written from the device to the PLC, e.g., values that the PLC generates itself for control purposes. In such a case, it would make sense to use tags without write attributes (only read/poll attributes).

Normally, the data direction for tags is defined in the PLC – read only, write only, or read & write. When tags are imported, the data direction will be automatically copied by GALILEO.

The data direction for tags to which addresses have been assigned directly can be changed in GALILEO.

For a description of tag addresses, please refer to the → section "Tag address", page 569 section.

Read (PLC -> Device)

Please enquire	Actions that cause the device to update the actual values: e.g. saving recipes. → This setting is not used to display tag values! Tags must always be polled for this purpose instead.
At Startup	If the data is used in the project, it will be read from the PLC when the device starts.
Polling	In order to display tag values, they first need to be polled from the PLC! If data changes slowly, a lower polling rate should be set so that other data can be read more frequently. The "fast" setting will cause the data to be read as fast as possible depending on the amount of data being transferred, the PLC protocol being used, and the cycle time for the device and the PLC. If you are not sure which polling rate to use, "fast" or "1 s" is recommended. Tags are polled in the following cases: • Tag is used in a graph data block and the graph is started • Tag is assigned an address for multiple communication systems (gateway)

Write (Device -> PLC)

Please enquire	Actions that cause the device to refresh the actual values: e.g. Value entry.

4.3 Working with tags

In GALILEO, tags are visualized, transmitted, entered, and/or processed.

GALILEO can be used to configure properties for the various tags, e.g., default values, limits, and character lengths. In addition, it can be used to assign addresses to tags, which must have an address whenever the device needs to read them from the PLC and display them. Both of these are described in greater detail in the → " Tags", page 563 section. And for information on how tags are linked in a visualization object, please refer to the sections for the corresponding visualization object.

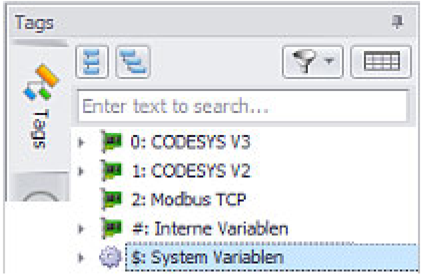
Remember:

Tags are always transferred in their whole!

When using complex data types, such as structures or arrays, all the corresponding elements will always be transferred, even if the PLC is only reading or writing to one of them.

Your project must ensure that all tags have valid addresses!
After a panel boots up, a tag will automatically have a starting value of <0> or the value assigned to it.

Project Organization\Tags



0: CODESYS V3
1: CODESYS V2
2: Modbus TCP
#: Interne Variablen
$: System Variablen

Figure 368: Tag overview with added PLC-communications, the internal GALILEO tags, and the system tags

4.3.1 Tag Overview

The *Project Tree\Tags* tab is hereafter referred to as "tag overview" for short. The tag overview is where tags are defined, tag settings are changed, etc.

The order in which tags are sorted will be the default sorting order used by Windows. This means, for example, that a tag called <bit2> will be listed after a tag called<bit15>. If, on the other hand, the tag names were <bit002> and <bit015>, tag <bit002> would be listed before tag <bit015>.

There are three basic groups of tags in a project:

- PLC tags
  These tags are stored under a node named after the corresponding communication system. If multiple communication systems are selected for a

project, this means that there will be multiple nodes for PLC tags. These nodes will be numbered in ascending order.

- Internal tags
- Systems Tags

PLC tags can be dragged& and dropped into the Internal tags node. In this case, the tags will lose their addresses and it will no longer be possible to transmit them to the PLC.

The opposite also applies, i.e., internal tags can be dragged& and dropped into a PLC communication system node. In this case, the tag will be assigned a default address for the corresponding PLC.

- PLCs that use symbolic addressing
  (Tag address is a symbolic name) The tag will be assigned a tag name as an address.
- PLCs that use absolute addressing
  (Tag address is an absolute address) The tag will be assigned the address syntax. The actual absolute address will still need to be defined.

It is necessary to check the properties for the tags to make sure that the default addresses assigned are correct and that the addresses are active.

Project Organization\Tags

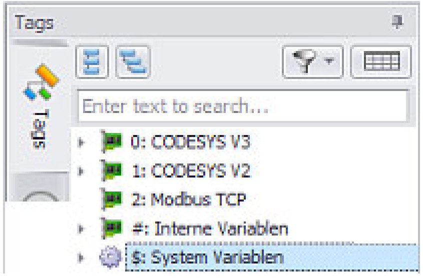


Figure 369: Tag overview with added PLC-communications, the internal GALILEO tags, and the system tags

Generally speaking, the tags in the Internal tags node will not be assigned an address. Meanwhile, the tags in the nodes for the PLC communication systems will generally be assigned to those systems. Addresses will normally be activated so that the corresponding tags have a proper address. However, these addresses can be deactivated if necessary, in which case the tags will be considered not to have an address.

Tags without an address will be shown in blue (X). Tags with an address will be shown in
green X .

The nodes for the PLC communication systems and the internal tags can contain both primitive and composite tags.

Primitive tags:

- x x Bit
- by by Byte
- dw dw Word
- d d Double
- w w Double Word (DWORD)
- qw qw Quad Word (QWORD)
- f f Float
- ab ab String

Composite tags:

- e e Error
- st st Structures
- T T Instances of user-defined data types

[Enter text to search...]

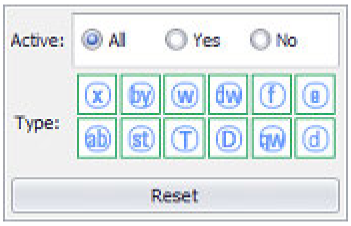
With entries in this field field can be used to search for tags directly. The nodes do not have to be expanded for this purpose.

If the search is successful, the appropriate tag can be clicked. This tag is then opened in the folder hierarchy, selected and can be opened with a double-click.

The following functions are available for the tag and data types:

/ Reduce/expand the structures

Filtering on specific tags in the overview



Active: ● All ○ Yes ○ No

Type: x by w dw f e ab st T D qw d

Reset

Active: filter for actively addressed tags, note that selecting Yes will hide all internal and system tags
Type: Select variable types, deselected types are removed from the display and shown with a red background

Reset deletes the set filters. If a filter is actively set -Key is highlighted.

Displaying the tag table,
Working with the tags table

4.3.2 Creating primitive tags

Before creating a tag, it is necessary to know whether the tag will be used exclusively in the device (i.e., internally) or whether it will be used to communicate with the PLC as well. In the first case, you will need to create an internal tag. In the second, you will need to create a PLC tag in GALILEO. In the latter case, you will also need to make sure that the PLC "knows" the tag. To do this, you can create the same tag with the same data type in the PLC.

To create a primitive tag, follow the steps below:

- ▶ Go to *Project Organization\Tags*.
- ▶ Select the Internal tags node or the node for the communication system you want, e.g., XSOFT-CODESYS-CODESYS V2.
- ▶ Open the context menu and click on Add\<data type>, e.g., DWord Array
- ▶ If you select an array as the data type, enter the number of array elements into the Array Size dialog box and click on OK.
- ▶ Enter the tag's name and confirm with OK.
  You can select any tag name you want. Upper and lower case characters from "a" to "z" and the "_" underscore character are permitted.

The tag has now been created and will be shown in the tag overview.

4.3.3 Creating a custom data type

Custom data type

Say a large number of motors need to be visualized in a project. In addition, there are three relevant pieces of data from the available data set that need to be visualized for each motor: MotorID, instantaneous motor current, and instantaneous speed.

In order not to have to create a tag of data type structure separately for each motor, you can instead create a custom data type called Motor.

Say you wanted to create a custom data type called Motor that reads or writes data using communication method 1:CODESYS.... To do this, you would follow the steps below:

- ▶ Go to *Home\Project Tree\Data Types*.
- ▶ Click on Data Types and then on User-defined types.
- ▶ Open the context menu. Select the Define a new type option. Enter the following name: <MOTOR>.
- ▶ Click on MOTOR, open the context menu, select Add\String, and enter MotorID.
- ▶ Click on MOTOR, open the context menu, select Add\Byte, and enter MotorCurrent.
- ▶ Click on MOTOR, open the context menu, select Add\Word, and enter Speed.

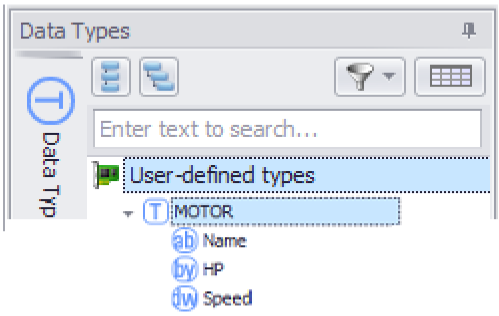


Figure 370: MOTOR data type with three structure elements

Data type instances

To add an instance of the defined custom data type, follow the steps below:

- Go to *Home\Project Tree\Tags*.
- Click on the Communication 1:CODESYS....
- Open the context menu. Select Add\User defined data type and 1.MOTOR.

Tag MOTORInstance0 will be created as an instance of custom data type MOTOR with three structure elements. You can rename this tag as necessary. However, the names of the structure elements it contains are defined by the custom data type and accordingly cannot be changed.

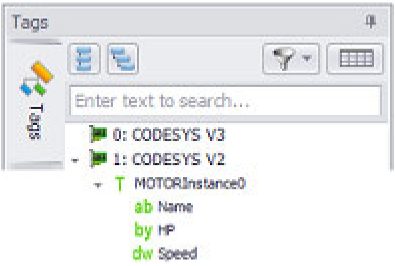


Figure 371: MOTORInstance0 tag of data type MOTOR

4.3.4 Converting a custom data type instance to a structure

A tag that is an instance of a custom data type can be converted to a tag of data type structure.

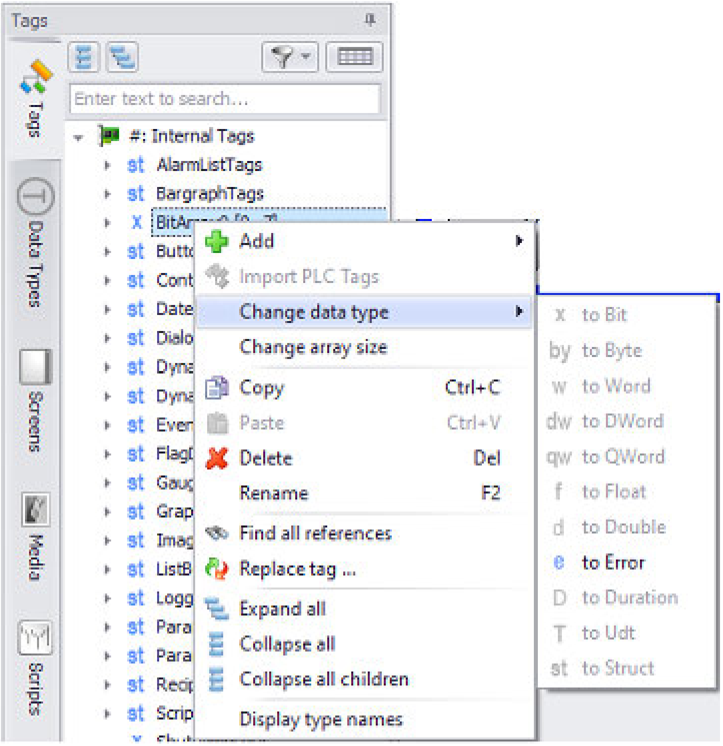
You can use this option if you realize, in the middle of creating a project, that you want to modify an existing data type with the exception of some of its individual instances. In this case, you would convert these individual instances of the custom data type to a structure before making changes to the data type.

You can also use this option if you realize, in the middle of creating a project, that you need additional tags that are similar to an existing custom data type. In this case, you would create an instance of the custom data type, convert it to a tag of data type structure, and then make the required changes.

To convert a custom data type to a new structure using the context menu, follow the steps below:

- Select the instance of the custom data type that you want to convert.
- Open the context menu and select *Change data type\to Struct*.

Home\Project Tree\Tags\context menu



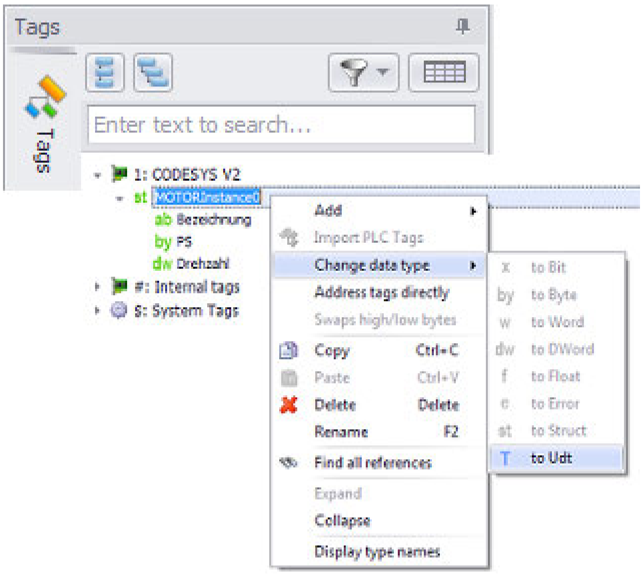
The data type change will be shown by the change to .

4.3.5 Converting a structure to an instance of a custom data type

You can use this option if, in the middle of creating tags, you realize that many of the tags have the same structure. In these cases, using a custom data type will make the process of creating additional tags much faster. To do so, you would use an existing tag to generate a custom data type.

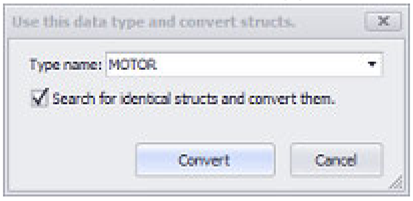
A tag of data type structure cannot be converted to an instance of a custom data type. For example, Struct0 cannot be converted to MOTORInstance1.

When a tag of data type structure is converted, the program will search for existing custom data types with the same structure. If none are found, a new custom data type will be created.



- Select the tag of data type structure that you want to convert.
- Open the context menu and select Change data type\to Udt

The following window appears:



Use this data type and convert structs.

Type name: MOTOR

☑ Search for identical structs and convert them.

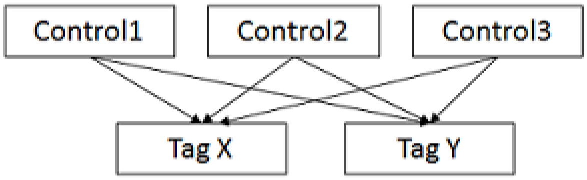
- In the Type name field, enter the name for the custom data type being created or leave the suggested name as is.
- If the <Search for identical structures and convert them.> checkbox is enabled, all tags with the exact same structure that are available for conversion will be shown. Selecting the corresponding structures will convert them to instances of the new type as well.
  Within this context, "identical structures" means that the structures have the same structure elements, with the same name, the same data type, and in the same order, as the reference structure.
- If the Search for identical structures and convert them. is disabled, the structure will be converted directly.
- Clicking on the Convert button will carry out the conversion.

4.3.6 Replacing tags in screens, scripts, recipes

GALILEO makes it easy to assign different tags to copied screens, scripts, and recipes. In fact, you can even replace a tag in the entire project tree with another one.

This means that you can use the exact same object multiple times by simply copying it and assigning different tags every time.

The references in multiple visualization objects, such as buttons, to a tag can be changed to references to a different tag with a single action. Moreover, this functionality is available both for simple and complex data types, Moreover, this functionality is available both for simple and complex data types, and replacing structures, arrays, errors, and custom data types is allowed as well.



→ Please note that tags have to have the same data type in order to be able to replace one with the other!

To replace the tags for a screen, follow the steps below:

- ▶ Open the Tags tab in the project tree.
- ▶ Select the tag that you want to replace.
- ▶ Right-click on the tag to open the context menu and select the "Replace tag ..." option.
- ▶ In the "Tag replace" dialog box, specify the tag that should be used to replace the existing tag, then confirm by clicking on Next.
- ▶ Now select the screens in which the first tag should be replaced with the second one and confirm by clicking on Finish.

4.3.7 Importing tags

The import function makes it possible to import tags from a PLC project or to re-import them in order to copy changes to the PLC project to the GALILEO project.

The changes are executed in accordance with the following points. The address is important for the comparison:

- An already existing tag is left as it is.
- Tags that are not yet in the GALILEO-project are generated again.
- Tags that are already in the GALILEO-project but not in the import file can be deleted .

Names, types and addresses are transferred with the import.

The following file formats are supported:

- Internal Tag Import Format (*.itf)
- CODESYS2 / MXpro Symbol table (*.sym_xml)
- CODESYS3 XML-Symbol table (*.xml)
- Siemens Step 7 - Symbol table (*.seq) and

DB (*.awl, *.scl)

- Siemens TIA - Symbol table (*.xlsx)
  DB (.awl, .scl)

CoDeSys symbol tables must always be generated with the Export data entries option from CODESYS V2.

Internal tag import format (*.itf)

Prerequisites

This file can be edited with any text editor or, for example, with Microsoft Excel. The program used must support text files with tab characters and must not replace these characters with spaces when saving.

The file starts with a header line that is ignored during the import. The format of this file must distinguish between PLCs with symbolic addressing and PLCs with absolute addressing.

PLCs with symbolic addressing

PLCs with symbolic addressing include "MXpro," "A. Bradley ControlLogix," etc.

The tags must be saved in the following format, with the columns separated by tabs:

Name	Part No	Device	Program	Structure	Day	Index
var_bit	BIT				var_bit	
var_byte	BYTE				var_ byte	
prog1.var_int	INT		prog1		var_int	
device.prog1.var_ float	FLOAT32	device	prog1		var_ float	
array_bit[10]	BIT				array_ bit	0
struct.var_word	WORD			struct	var_ word	

PLCs with absolute addressing

PLCs with absolute addressing include "Moeller PS4-341-MM1 Suconet K," "Siemens MPI," etc.

The tags must be saved in the following format, with the columns separated by tabs:

Name	Part No	Address
var_bit	BIT	ST2:DB20.DBB60.0
var_byte	BYTE	ST2:DB20.DBB40
prog1.var_int	INT	ST2:MB4
device.prog1.var_float	FLOAT32	FLOAT32 ST2:MB10
array_bit[10]	BIT	ST2:DB10.DBB2.0
struct.var_word	WORD	ST2:AB10

Name

Contains the tag name including the complete structure. Arrays can be defined in one line, and the array size must be appended to the tag/file name with [xy].

Part No

This column contains the format of the tag :

Part No	Context
BIT	0…1
BYTE	0…255
SINT	-128…127
WORD	0…65535
INT	-32768…32767
DWORD	0…4294967295
LINT	-2147483648…2147483647
FLOAT32	-3.402823E+038…3.402823E+038
STRING	

Device, Program, Structure, Tag, Index

The Device, Program, Structure, Tag, and Index columns represent the symbolic address. The index column is only used for arrays. The addressing of the 0th element starts with this index.

Address

This column contains the identical address to the one used in GALILEO.

Prerequisites

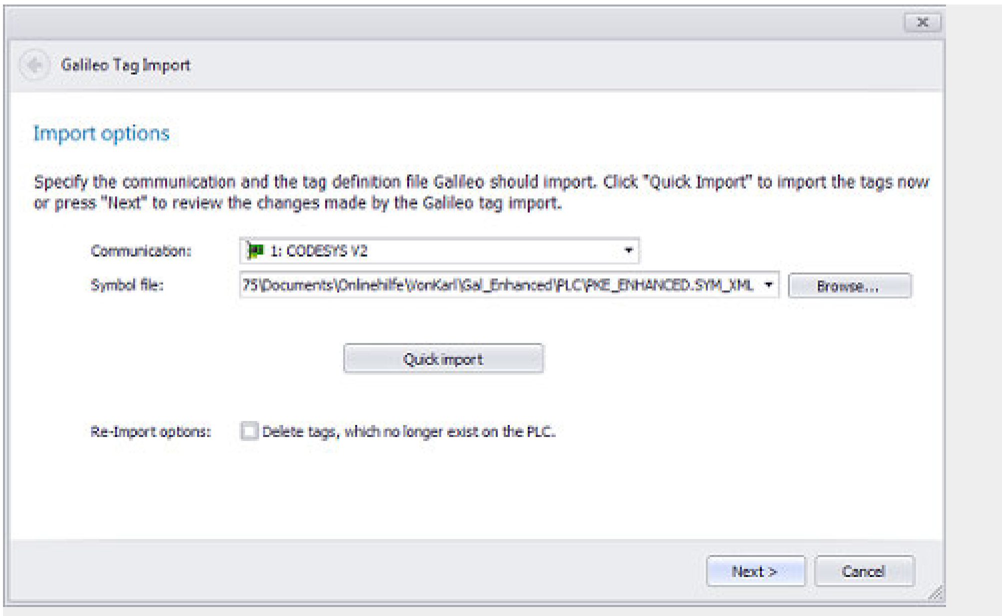
Since the process of importing tags always refers to a PLC, the corresponding communication method must always be defined in the project first.

To Import from Tags proceed as follows:

- Select *Project Organization\Tags\<Communication>*.
- Right-click on it to open the context menu screen.
- Select Import PLC Tags.

The Galileo Tag Import dialog box will appear.

Galileo Tag Import dialog box



Galileo Tag Import

Import options

Specify the communication and the tag definition file Galileo should import. Click "Quick Import" to import the tags now or press "Next" to review the changes made by the Galileo tag import.

Communication: 1: CODESYS V2

Symbol file: 75\Documents\Onlinehilfe\vonKarl\Gal_Enhanced\PLC\PKE_ENHANCED.SYM_XML

Re-Import options: □ Delete tags, which no longer exist on the PLC.

▶ Select the symbol file that contains the tags you want to import by clicking on the Browse... button.

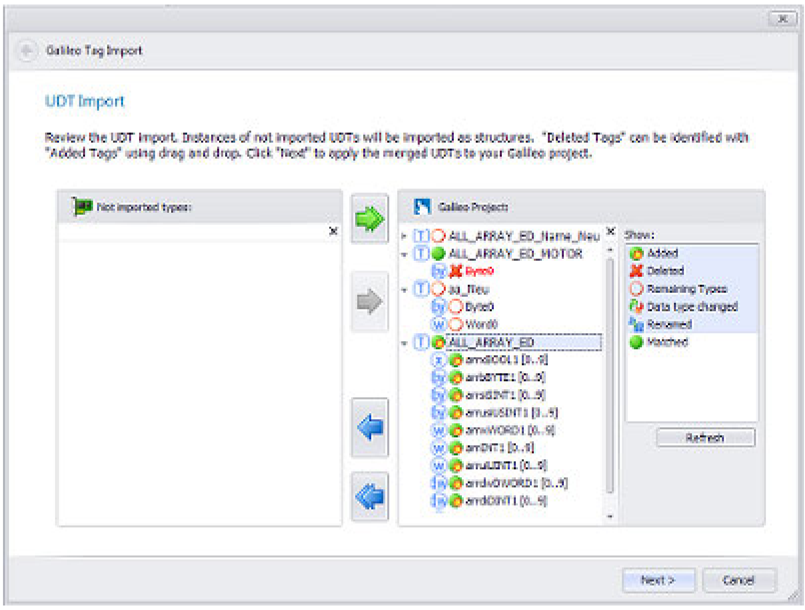
You will be able to select any of the aforementioned file formats.

▶ If you want to import all the tags in the symbol file, click on the Quick import button to import them right away.

If the import file contains user-defined data types, they are also automatically added in GALILEO.

4.3.8 Re-importing tags

Project Tree\Tags\<PLC tag>\Context menu\Import PLC Tags



Galileo Tag Import

UDT Import

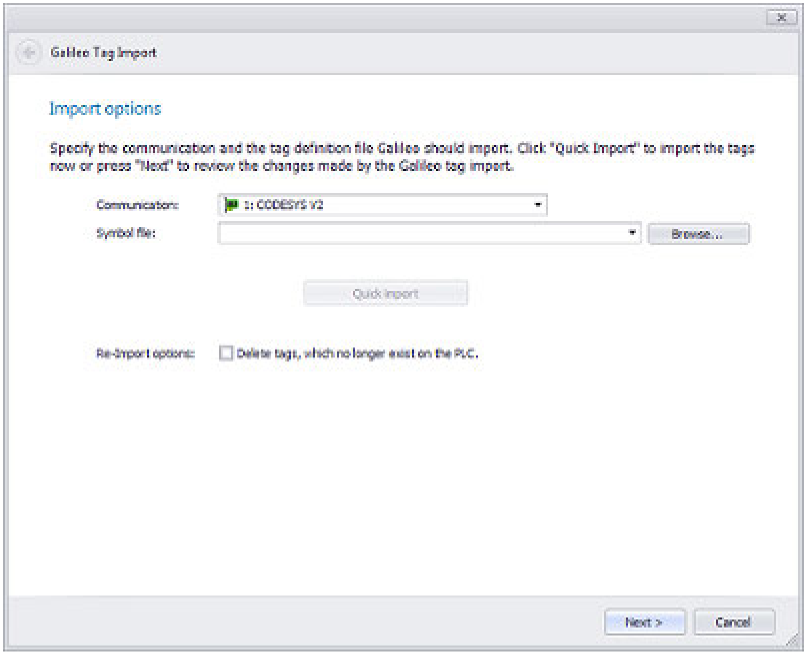
Review the UDT import. Instances of not imported UDTs will be imported as structures. "Deleted Tags" can be identified with "Added Tags" using drag and drop. Click "Next" to apply the merged UDTs to your Galileo project.

Figure 372: Galileo Tag Import dialog box for importing and re-importing tags

To Re-Import from Tags proceed as follows:

- Select *Project Organization\Tags\<Communication>*.
- Right-click on it to open the context menu screen.
- Select <Import PLC Tags>.

The Galileo Tag Import dialog box will appear.



Galileo Tag Import

Import options

Specify the communication and the tag definition file Galileo should import. Click "Quick Import" to import the tags now or press "Next" to review the changes made by the Galileo tag import.

Communication: 1: CODESYS V2

Symbol file:

Re-Import options: □ Delete tags, which no longer exist on the PLC.

Select the symbol file that contains the tags you want to import by clicking on the Browse... button.

You will be able to select any of the aforementioned file formats.

At this point, there will be two different options available for re-importing the tags:

1. Quick import
2. Re-Import in three steps

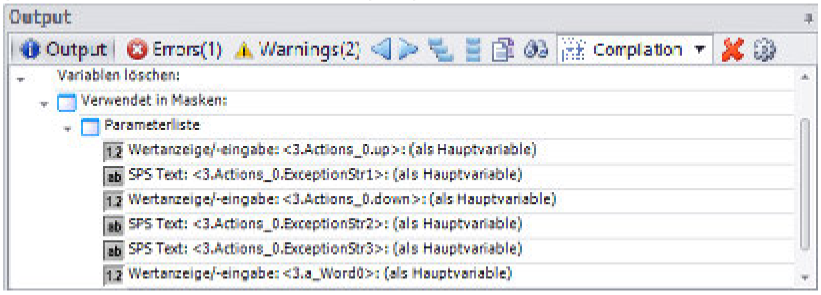
4.3.8.1 Quick import

If you want to import all the tags in the symbol file, simply run the import right away.

- ▶ To do this, click on the Quick import button in the Galileo Tag Import dialog box.

If you use this option, you will not be able to do anything about differences between the entries in the symbol file and the tags already found in the GALILEO project. Accordingly, all tags not found in the project will be imported.

If the <Delete tags, which no longer exist on the PLC> checkbox is enabled, all tags that are found in the GALILEO project but not in the symbol file will be deleted directly with the import. Please note that there will not be any confirmation prompt asking if you are sure! However, the output pane will list the references for the deleted tags.



```
Variablen löschen:
  Verwendet in Masken:
    Parameterliste
      1.2 Wertanzeige/-eingabe: <3.Actions_0.up>: (als Hauptvariable)
      ab SPS Text: <3.Actions_0.ExceptionStr1>: (als Hauptvariable)
      1.2 Wertanzeige/-eingabe: <3.Actions_0.down>: (als Hauptvariable)
      ab SPS Text: <3.Actions_0.ExceptionStr2>: (als Hauptvariable)
      ab SPS Text: <3.Actions_0.ExceptionStr3>: (als Hauptvariable)
      1.2 Wertanzeige/-eingabe: <3.a_Word0>: (als Hauptvariable)
```

4.3.8.2 Re-Import in three steps

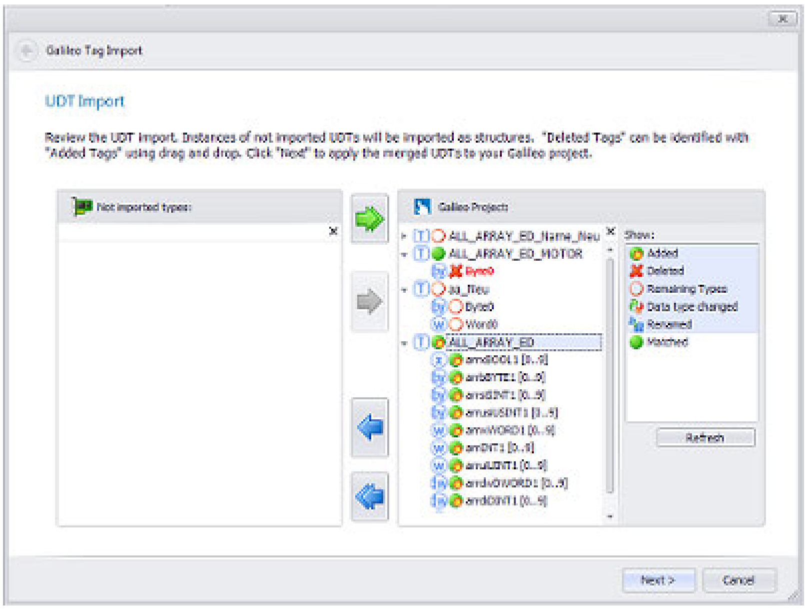
- ▶ Click on Next in the >Galileo Tag Import dialog box.

An overview of the import for custom data types will be shown. At this point, the import has not yet been carried out. The changes that will be made to the project as a result of the import will be shown in the right pane. You can use this step to specify whether data types added to or deleted from the project as a result of the import should be renamed or changed.

If you want to apply all changes, click on Next in the >Galileo Tag Import dialog box.

If there are any changes to the tags in the project that you do not want, select them in the right pane and move them to the pane on the left by clicking on ⬅.

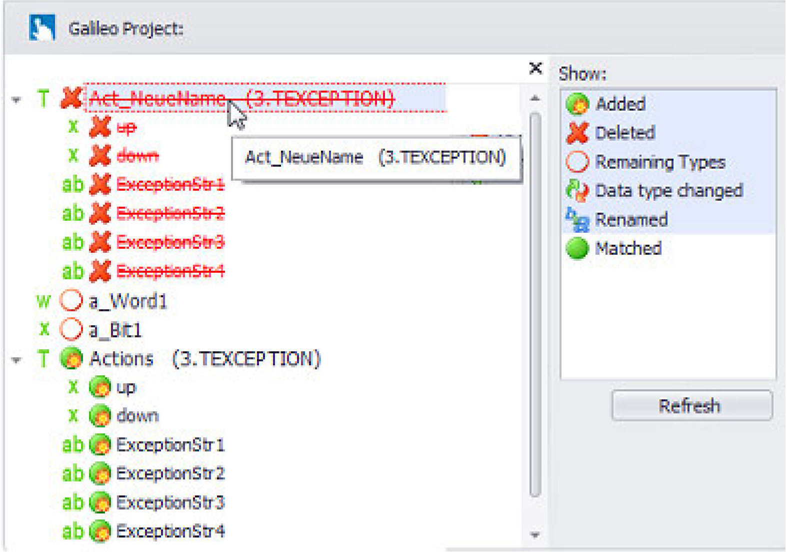
Once the changes you want are in the right pane and the changes you don't want are in the left pane, click on Next>.

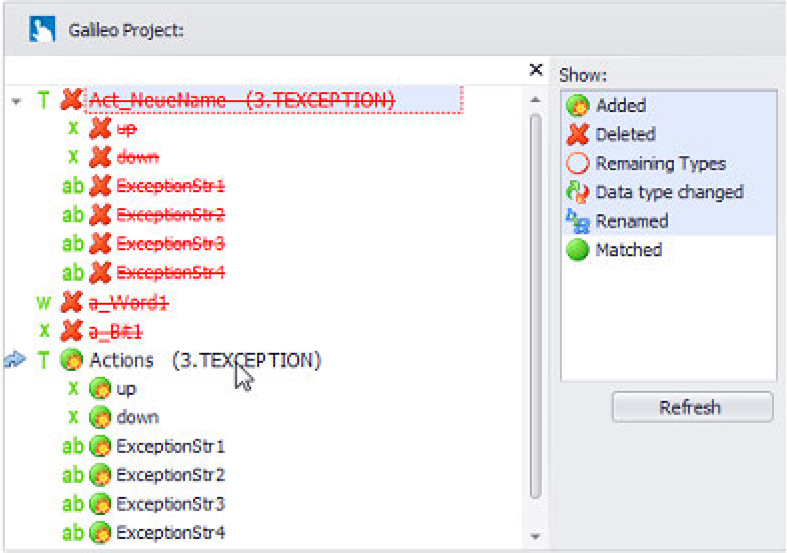


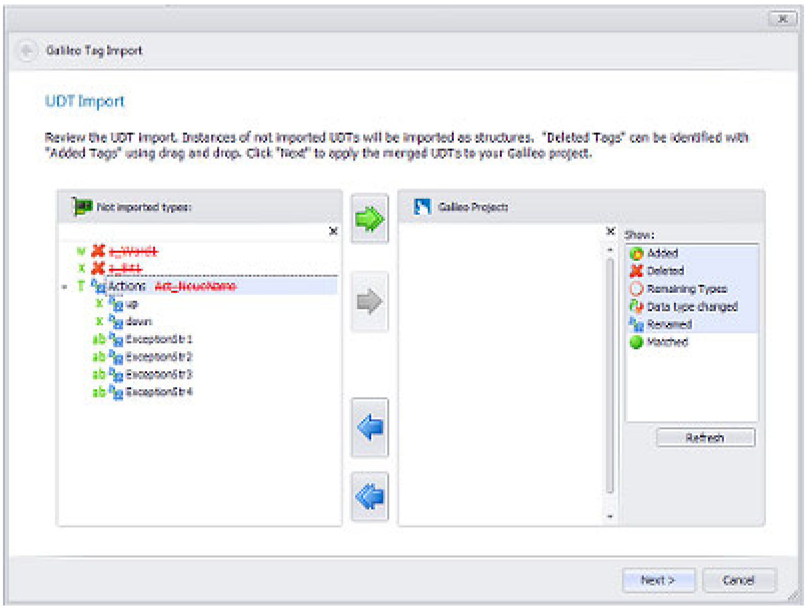
Review the UDT import. Instances of not imported UDTs will be imported as structures. "Deleted Tags" can be identified with "Added Tags" using drag and drop. Click "Next" to apply the merged UDTs to your Galileo project.

Clicking on "Next" will show the import overview for the tags and data type instances. You can use this step to select tags and use the arrow buttons in order to specify whether tags that are not found in the project, have been modified, or have been deleted should be removed from or added to the Galileo project, much like with the custom data types.

In addition, you can also directly join tags for which e.g., the name, address, or data type has changed by dragging the tag marked as deleted and dropping it onto the new tag.



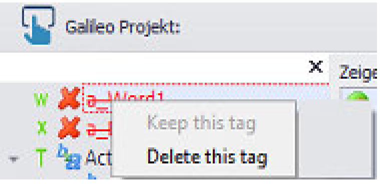



Galileo Tag Import

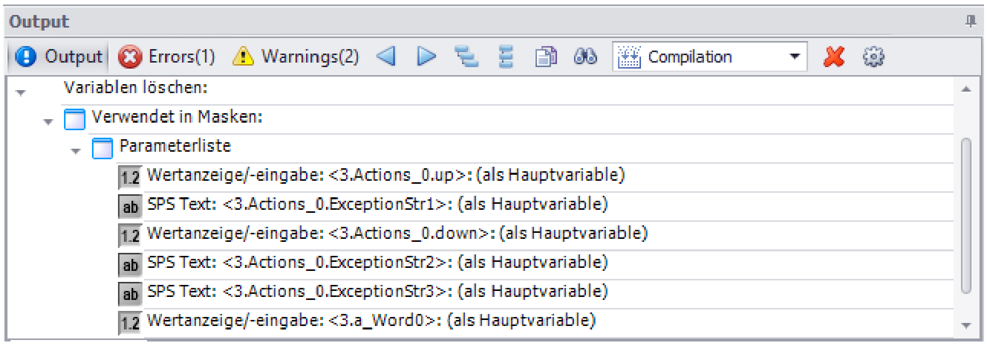
UDT Import

Review the UDT import. Instances of not imported UDTs will be imported as structures. "Deleted Tags" can be identified with "Added Tags" using drag and drop. Click "Next" to apply the merged UDTs to your Galileo project.

It is also possible to keep tags marked as deleted by right-clicking on them to open the context menu and selecting the "Keep this tab" option.



Once you are done making changes, click on "Finish" to re-import the tags as specified. After re-importing the tags, make sure to check the Output pane, which will list the references for the deleted tags.



```
Variablen löschen:
  Verwendet in Masken:
    Parameterliste
      Wertanzeige/-eingabe: <3.Actions_0.up>: (als Hauptvariable)
      SPS Text: <3.Actions_0.ExceptionStr1>: (als Hauptvariable)
      Wertanzeige/-eingabe: <3.Actions_0.down>: (als Hauptvariable)
      SPS Text: <3.Actions_0.ExceptionStr2>: (als Hauptvariable)
      SPS Text: <3.Actions_0.ExceptionStr3>: (als Hauptvariable)
      Wertanzeige/-eingabe: <3.a_Word0>: (als Hauptvariable)
```

Several PLCs

The imported tag addresses are addressed to the PLC selected here. If several PLCs are defined in the project, the import can be executed several times in order to import the tags of different PLCs.

4.3.9 Tag context menu

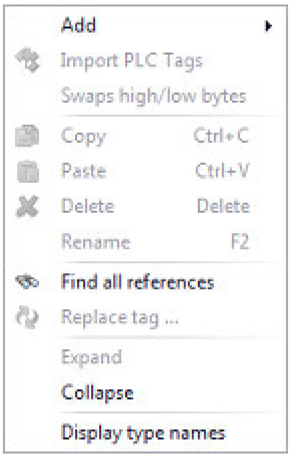


Figure 373: Context menu to *Project Organization\Tags\<Folder>*

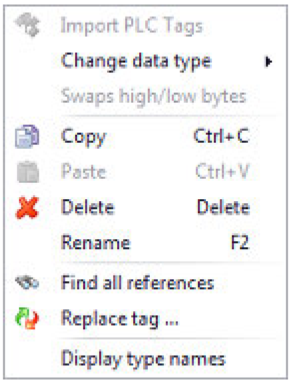


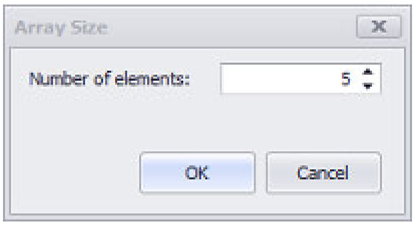
Figure 374: Context menu to <Tag>

New

Available only if a node, an array, or a structure is selected in the tag overview. Depending on the selected node, a new internal tag, PLC tag, or system tag will be added to the project. If an array is selected, you will be able to add a new array element to the array. Finally, if a structure is selected, you will be able to add any data type to it. For detailed information on how to create a new tag of a custom data type, please refer to → section "Creating a custom data type", page 542.

You can enter any tag name as long as it does not exceed 256 characters or use characters that are not permitted. Upper and lower case characters from "a" to "z", numbers and the "_" underscore character are permitted. The tag name must not start with a number.

Setting an array size



Number of elements: 5

Defines the number of elements in an array.

Please note:

In the case of arrays to which addresses are assigned, the number of elements must be adjusted in line with the supported PLC data sizes (pre-defined selection). Available data sizes will depend on the specific PLC being used.
In the case of tags of type Byte, Wort, Double Word, and Float, keep in mind that the same limits apply to all array elements.
All the elements in a bit array will have the same default value.

Import PLC Tags

→ "Importing tags", page 546

Swaps high/low bytes

Swap the bytes of arrays as follows:

Catalog Number	Description
Bit-Array	Replace bits 0...7 with 8...15, 16...23 with 24...31; etc.
Error-Array	Replace bits 0...7 with 8...15, 16...23 with 24...31; etc.
Byte-Array	Swap Byte 0 with 1, Byte 2 with 3; etc.

Cut, Copy, Paste, Delete

The selected tag can be copied, cut, and pasted using the clipboard. If you try to delete or cut a tag and the tag is used by the project, the following dialog box will appear.

If you delete a tag, it will be deleted from the project and the corresponding assignments will be deleted as well. Screen files not currently opened may also be modified under certain circumstances. You can click on to undo a delete operation.

Find all references

This function can be used to find the visualization objects and objects in which the tag is being used. The results will be shown in the Output pane.

Expand / Collapse

This command will only be available when a node, array, or structure is selected. The Expand command will expand the node, array, or structure to show the elements in it.

The Collapse command will collapse the node, array, or structure to hide the elements in it.

Change data type

This menu item is used to change a byte tag into a word tag, without other changes having to be executed in the project. This menu point is only visible and active if the type can also be changed.

Replace tag...

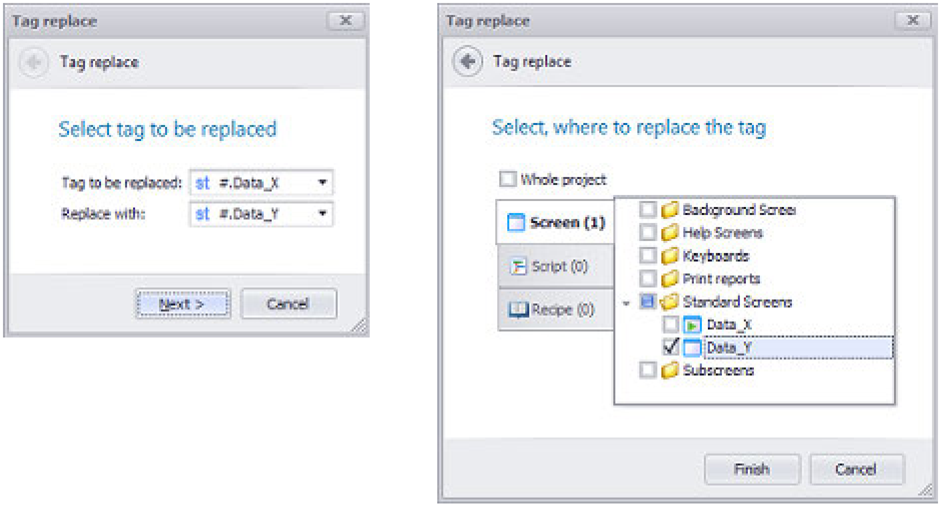
There is a wizard you can use to replace a tag with a different one quickly and easily.

This function is available for individual tags, structures, and arrays.
The tag being replaced and the replacement tag must be of the same type.

- Select the Tag to be replaced.
- In Replace with, select the tag you want from the available existing tags.
- Click on Next
- Select the instance(s) of the tag where you want it to be replaced.

In addition to the whole project, you can instead choose to select only individual project tree sections (e.g., Screens or Scripts). In fact, you can decide to replace the tag at the individual element level.

Table 5: Wizard for replacing tags


Tag replace

Select tag to be replaced

Tag to be replaced: st #.Data_X
Replace with: st #.Data_Y

Select, where to replace the tag

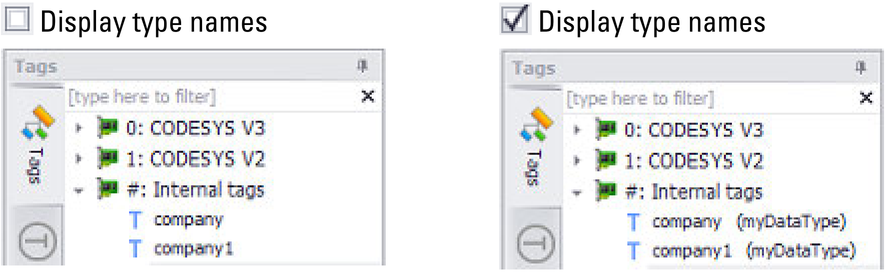
□ Whole project

Screen (1)
Script (0)
Recipe (0)

□ Background Screens
□ Help Screens
□ Keyboards
□ Print reports
▣ Standard Screens
□ Data_X
☑ Data_Y
□ Subscreens

Show data type

For tags of a custom data type, the name of the data type will be shown in addition to the tag name.


□ Display type names

Tags
[type here to filter]
0: CODESYS V3
1: CODESYS V2
#: Internal tags
company
company1

☑ Display type names

Tags
[type here to filter]
0: CODESYS V3
1: CODESYS V2
#: Internal tags
company (myDataType)
company1 (myDataType)

Figure 375: Tags company and company1 of custom data type myDataType.

4.3.10 Create array of controls

Dragging and dropping (left mouse button) tag arrays will open the Create array of controls dialog box for the object. This dialog box can be used to configure/define the area/box for the display.

Dragging and dropping (right mouse button) will open the context menu.

Tags/Internal tags/Add

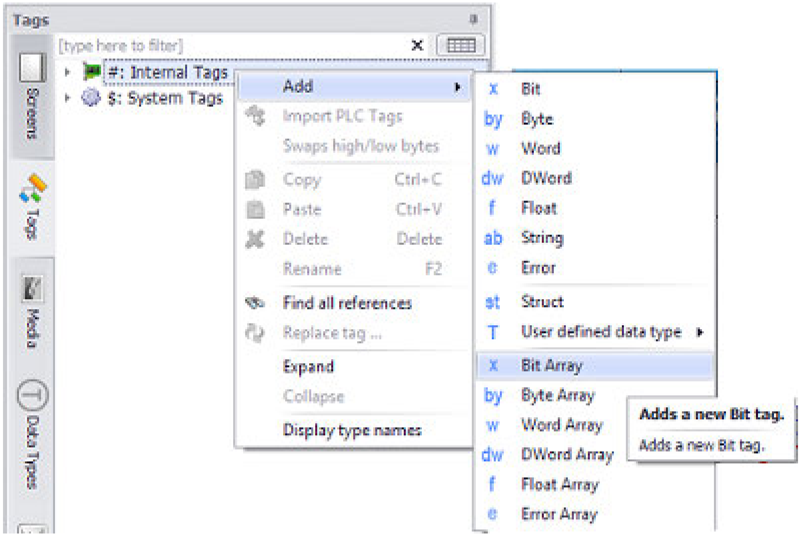


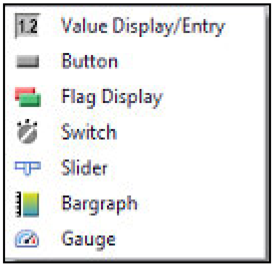
Figure 376: Adding a tag array

For

- Bit tags, a button object
- All other tag types/strings, a value display/entry / PLC text object

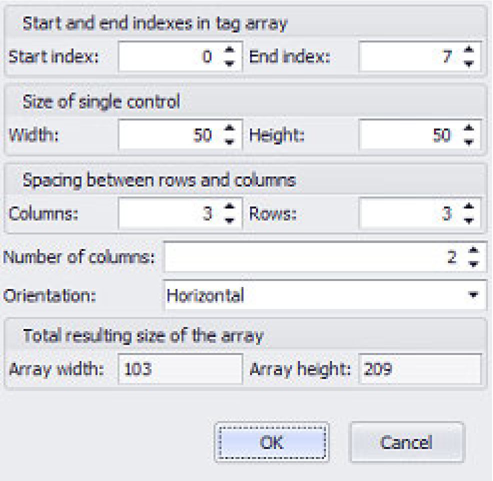
will be generated by default.

If you want to use the drag and drop function to generate a special object, you will need to use the right mouse button instead of the left one.
This will open a context menu such as the following:



If you select an object, the following dialog box will appear when dragging and dropping (left mouse button) onto the screen.

Create array of controls dialog box



Start and end indexes in tag array
Start index: 0 End index: 7

Size of single control
Width: 50 Height: 50

Spacing between rows and columns
Columns: 3 Rows: 3

Number of columns: 2

Orientation: Horizontal

Total resulting size of the array
Array width: 103 Array height: 209

Figure 377: Description

Start and end indexes in tag array	If you only want to create objects for part of the array, you can use these settings to adjust the start and end indexes for the tag array. Only objects with tag elements within the range you specify will be created.
Start index:	First value
End index:	Last value Input values must be representable
Size of single control	Used to enter the area in pixels
Width:	
Height:	
Spacing between rows and columns	Used for alignment on the screen Specified in pixels
Columns:	from 0 to 1240
Rows:	from 0 to 1080
Number of columns	Into which the individual objects will be split with the selected spacing
Alignment	Horizontal or vertical area orientation
Total resulting size of the array	Shows the resulting values based on the settings configured
Array width:	Calculated width
Array:Height	Calculated height

Click on OK.

The matrix for the array will be displayed on the screen.

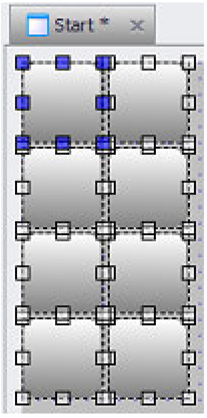


Figure 378: Example of a button array with the matrix configuration above

4.4 Working with the tags table

To open the tag table, click on the ▦ button in the Tags tab.

Tags tab image in the project tree

Project Organization\Tags

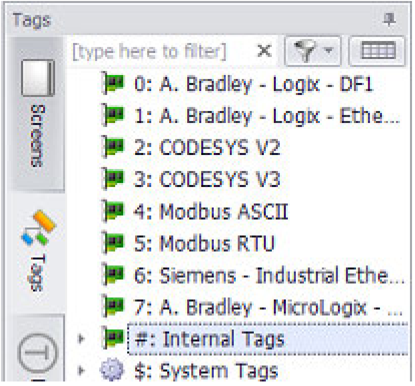
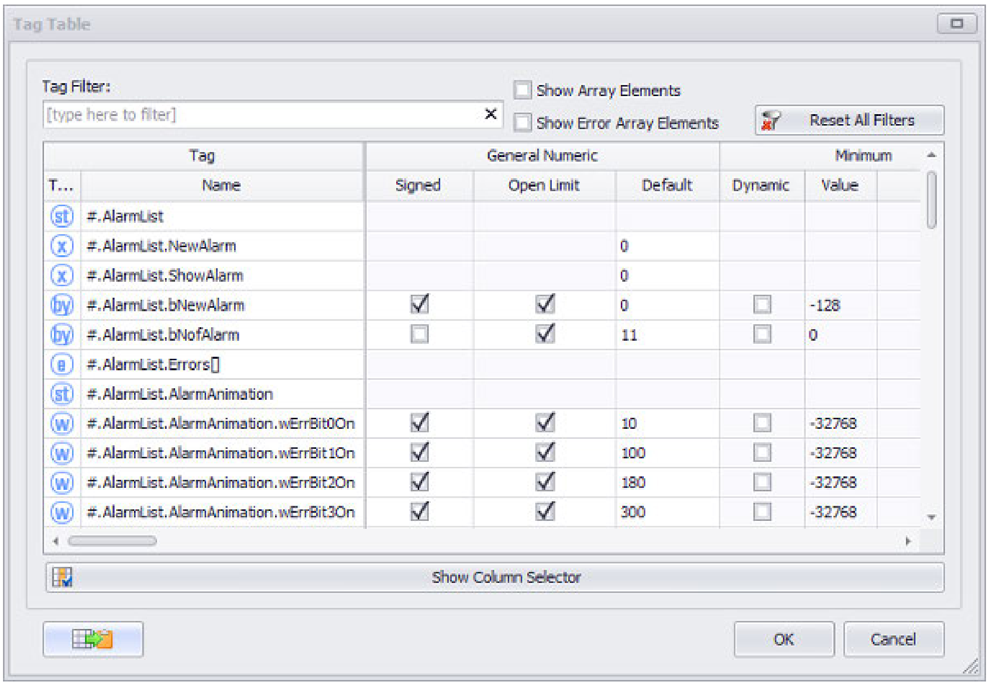

Tags
[type here to filter]
Screens
Tags
0: A. Bradley - Logix - DF1
1: A. Bradley - Logix - Ethe...
2: CODESYS V2
3: CODESYS V3
4: Modbus ASCII
5: Modbus RTU
6: Siemens - Industrial Ethe...
7: A. Bradley - MicroLogix - ...
#: Internal Tags
$: System Tags

Figure 379: Project Tags tab overview

You can use the tags table to see and modify the properties of all tags regardless of data type.

- You can get a direct overview of all the tags' properties.
- You can generate filtered and sorted views of tag properties.
- You can quickly and easily define and copy multiple values.
- You can use the table to do a quality control check of the various tag properties.
- You can define which tags should be logged in the user logger

Editable properties will be indicated by means of a white field background. In contrast, a gray background means that the corresponding property cannot be defined for the relevant data type.



Tag Table

Tag Filter: [type here to filter] ✕

□ Show Array Elements

□ Show Error Array Elements

Reset All Filters

Tag		General Numeric			Minimum	
T...	Name	Signed	Open Limit	Default	Dynamic	Value
st	#.AlarmList					
x	#.AlarmList.NewAlarm			0		
x	#.AlarmList.ShowAlarm			0		
by	#.AlarmList.bNewAlarm	☑	☑	0	□	-128
by	#.AlarmList.bNofAlarm	□	☑	11	□	0
e	#.AlarmList.Errors[]					
st	#.AlarmList.AlarmAnimation					
w	#.AlarmList.AlarmAnimation.wErrBit0On	☑	☑	10	□	-32768
w	#.AlarmList.AlarmAnimation.wErrBit1On	☑	☑	100	□	-32768
w	#.AlarmList.AlarmAnimation.wErrBit2On	☑	☑	180	□	-32768
w	#.AlarmList.AlarmAnimation.wErrBit3On	☑	☑	300	□	-32768

Show Column Selector

OK Cancel

Figure 380: Tag Table dialog box

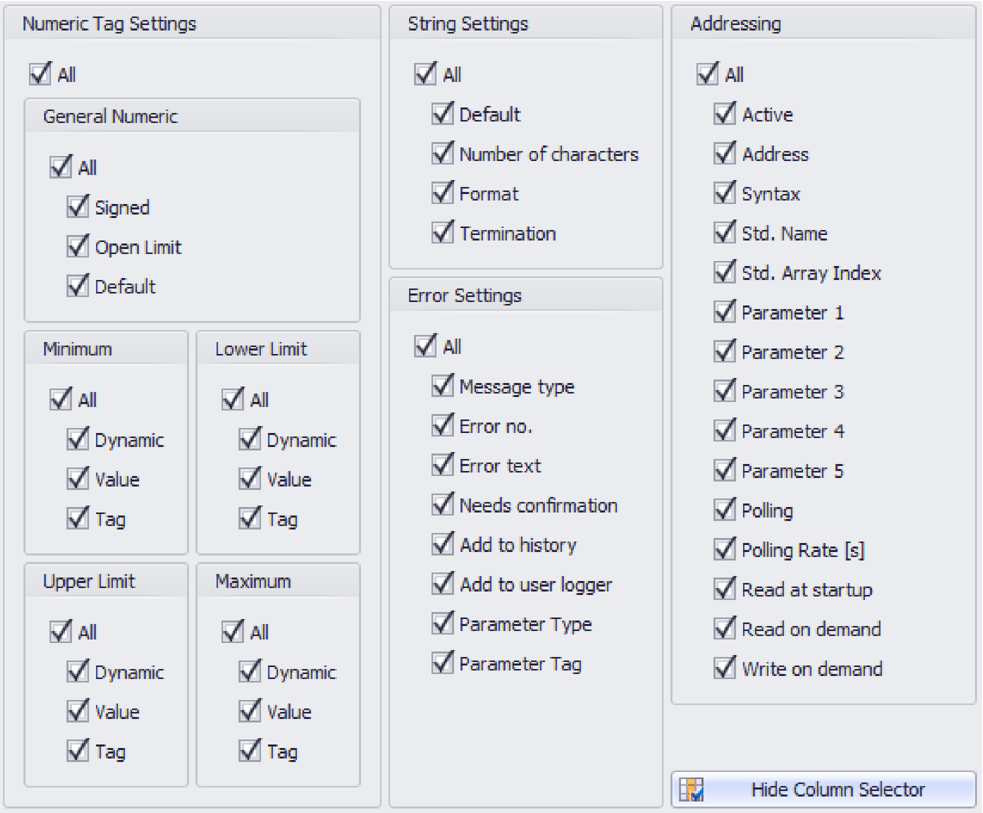
If you want to add tags to the User Logger, you will need to enable ☑ the corresponding checkboxes either individually or as a group in the column. To exclude them, you will need to disable □ the checkbox instead. You can also add or exclude all of them using the Column Selector.

To get a more manageable table, you can click on the "Show Column Selector" button and select which settings should be displayed. The following options will appear:

Column Selector

Clicking on the Show/Hide Column Selector button will show/hide the Column Selector in the lower part of the Tag Table dialog box.

The Column Selector can be used to select which tag properties (such as the address) should be shown or hidden in order to make the table more manageable.



Numeric Tag Settings

☑ All

General Numeric

☑ All
- ☑ Signed
- ☑ Open Limit
- ☑ Default

Minimum

☑ All
- ☑ Dynamic
- ☑ Value
- ☑ Tag

Lower Limit

☑ All
- ☑ Dynamic
- ☑ Value
- ☑ Tag

Upper Limit

☑ All
- ☑ Dynamic
- ☑ Value
- ☑ Tag

Maximum

☑ All
- ☑ Dynamic
- ☑ Value
- ☑ Tag

String Settings

☑ All
- ☑ Default
- ☑ Number of characters
- ☑ Format
- ☑ Termination

Error Settings

☑ All
- ☑ Message type
- ☑ Error no.
- ☑ Error text
- ☑ Needs confirmation
- ☑ Add to history
- ☑ Add to user logger
- ☑ Parameter Type
- ☑ Parameter Tag

Addressing

☑ All
- ☑ Active
- ☑ Address
- ☑ Syntax
- ☑ Std. Name
- ☑ Std. Array Index
- ☑ Parameter 1
- ☑ Parameter 2
- ☑ Parameter 3
- ☑ Parameter 4
- ☑ Parameter 5
- ☑ Polling
- ☑ Polling Rate [s]
- ☑ Read at startup
- ☑ Read on demand
- ☑ Write on demand

Figure 381: Column Selector in Tag Table dialog box

Copy table to clipboard

Clicking on the button will copy the tag table to the clipboard, after which it can be pasted into Microsoft Excel® or Microsoft Word® and worked on there.

- Excel - unformatted copy of cell contents
- Word - tab-separated cell contents

4.5 Tags

You can use the Properties <Tag> pane to configure each individual tag in line with the corresponding tag data type and the intended use of the tag in the project.
In order to be able to do this, you must have first created the tag in the project tree.

The parameters that can be configured will depend on the specific tag type.

- Tag properties can be used to define the limits within which the values of numeric or string tags are allowed to vary.
- Properties can also be used to define and activate the addresses for tags and the type of data communication used.
- Structures can be assigned an address, and the type of data communication used can also be configured for them. Each structure is made up of several structure elements, each of which will be a data type with its own properties.
- Internal tags are used exclusively within GALILEO. Because of this, an address is not assigned to them.

You can enter any tag name as long as it does not exceed 256 characters or use characters that are not permitted. Upper and lower case characters from "a" to "z", numbers and the "_" underscore character are permitted. The tag name must not start with a number.

The following sections describe the individual properties for the various tags.

Project Organization\Tags\PLC-Tags <Value>

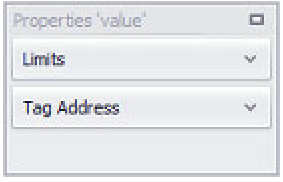


Figure 382: Properties of the "Value" tag

Numeric tag limits

Several limits can be defined for tags of data type BYTE, WORD, DWORD, and FLOAT in GALILEO. Minimum and Maximum are used to define the limits within which the tags' input value can move. These limits will be checked during input as soon as a device operator has the option of entering a value for one of these tags on the device.

If the device is used to enter a value that is less than this minimum, the value will automatically be set to the minimum when the ENTER key is tapped on and the keyboard will not be closed. If the device is used to enter a value that is greater than this maximum, the value will automatically be set to the maximum when the ENTER key is tapped on and the keyboard will not be closed. However, please note that it is completely possible for the tag value to fall outside these limits if is being sent by the PLC.

Bit tags do not have limits. The only value that can be set for them is the default value.
Structures do not have any limits either.

An upper limit and lower limit can also be defined for these tags. If a visualization object is linked to the tags, it will be able to use these limits as warning/notification thresholds, In this case, the way in which the alarm will be output will depend on the specific visualization object. For example, the bargraph visualization object can change colors or flash when a lower limit is fallen below or an upper limit is exceeded – please refer to → " Bargraph", page 356.

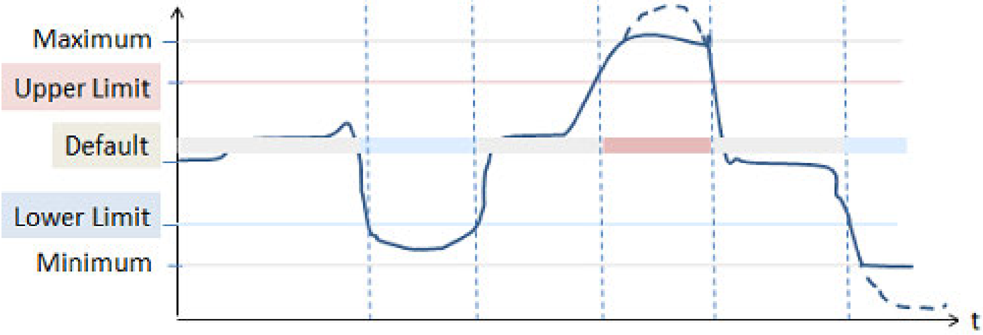


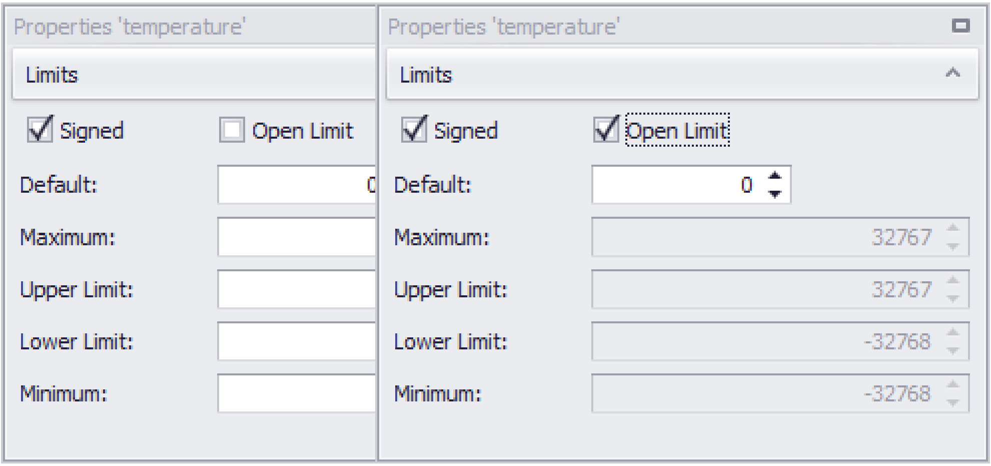
Figure 383: Diagram showing tag limits with different colors when the upper limit is exceeded and the lower limit is fallen below

The following limits apply to Limits:

Maximum ≧ Upper Limit ≧ Lower Limit ≧ Minimum
Maximum ≧ Default ≧ Minimum

If one of these rules is broken, the values entered in the corresponding field in the Properties pane will be highlighted in red. Accordingly, make sure that the limits meet the above rules. During compilation, the rules will be checked and the compiler will output any relevant messages.

Project Organization\Tags



Properties 'temperature'

Limits

☑ Signed □ Open Limit

Default: 0
Maximum:
Upper Limit:
Lower Limit:
Minimum:

Properties 'temperature'

Limits

☑ Signed ☑ Open Limit

Default: 0
Maximum: 32767
Upper Limit: 32767
Lower Limit: -32768
Minimum: -32768

Figure 384: Open Limit disabled; all values entered manually

Figure 385: Open Limit enabled

The limits for the tags, including the minimum and maximum, can be set as constant values, or as dynamic values using tags. To set them as constant values, simply enter the numbers you want into the corresponding fields. To set them as a dynamic limits instead, enter the tags you want. Keep in mind that, if you are using a PLC tag, a configured conversion factor will be applied when calculating the limit. For a description of how value conversion works, please refer to the following section: → "Conversions", page 921.

with sign

The value range will depend on whether the Signed checkbox is enabled, as well as on the tag's data type.

Data type	Unsigned	Signed
Byte	0 to 255	-128 to +127
Word	0 to 65.535	-32.768 to +32.767
DWord	0 to 4.294.967.295	-2.147.483.648 to +2.147.483.647
Float	not possible	-3,40282346638529 *10^{38} to +3,40282346638529 *10^{38}

Open Limits

□ Open Limits

Initially, the lower limit will be set to the minimum and the upper limit to the maximum.
It will be possible to freely define the minimum, maximum, lower limit, and upper limit within the value range for the relevant data type.

☑ Open Limits

The lower limit will be set to the minimum and the upper limit to the maximum.
The minimum and maximum will automatically correspond to the value range limits for the relevant data type.

Value tag default

The default value is used to initialize the visualization object when the device is working offline. This will be the case in the following situations:

- When starting up the device
- When there is no connection to the PLC because the device is working offline
- When displaying internal tags that are not used to communicate with the PLC

The default value will be 0 by default. The value will be shown in the color specified in the States tab

Maximum

This field is used to set the maximum input value. If the device is used to enter a value that is greater than this maximum, the value will automatically be set to the maximum when the ENTER key is tapped on and the keyboard will not be closed. Values entered are always decimal values.

→ A visualization object's minimum and maximum are not the same as the corresponding tag's minimum and maximum. However, the limits for the tag can be used as the initial default values for the visualization object. For detailed information on how to do this, please refer to the properties for the relevant visualization object.

Upper Limit

This field is used to enter the upper limit, which visualization objects will be able to use as a warning/notification threshold. The corresponding color will change as soon as the tag's current value exceeds the upper limit.

Lower Limit

This field is used to enter the lower limit, which visualization objects will be able to use as a warning/notification threshold. The corresponding color will change as soon as the tag's current value falls below the lower limit.

Minimum

This field is used to set the minimum input value. If the device is used to enter a value that is less than this minimum, the value will automatically be set to the minimum when the ENTER key is tapped on and the keyboard will not be closed. Values entered are always decimal values.

→ A visualization object's minimum and maximum are not the same as the corresponding tag's minimum and maximum. However, the limits for the tag can be used as the initial default values for the visualization object.

Defining dynamic values with tags

Instead of entering a fixed value, you can link the field to a tag, so that the value can be changed dynamically. To link to a tag, follow the steps below:

- Place the cursor inside the field.
- Right-click to open the context menu and select the Use Tag option.

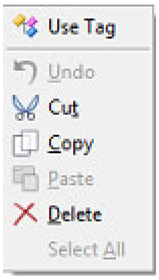
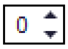
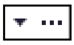


Figure 386: Context menu with Use Tag option

The field will go from looking like this:  to looking like this: .

- ▶ Click on the drop-down menu arrow and select the tag that should represent the value.
- ▶ Clicking on the ... icon will open the Properties <tag name> dialog box. Make sure that the settings for the tag are correct, e.g, Default, Maximum. If necessary, you can also use tags here in order to define limits. During compilation, the system will check whether there are recursive tag references and, if so, will output an error message.

→ All value tags are always declared in decimal format regardless of the selected display format (decimal, hex, binary, octal, BCD)!

→ All values are PLC values. They will be transmitted to the PLC the way they are, i.e., without being converted.

String tag limits ab

A string tag is used to display a changeable string (i.e., a changeable sequence of characters) on the device, with the text being transmitted to the PLC.

The value display/entry visualization object is configurable, and it can be assigned a created tag. If you assign it a numeric tag, the corresponding properties will be shown – please refer to → section " Value Display/Entry", page 506.
If you assign it a string tag instead, the corresponding properties will be shown.

In real-life applications, string tags are used to make it possible to display texts from the PLC in a flexible manner, among other things. For instance: if an automation component that outputs error messages exclusively in text format is installed in a system, the corresponding text can be assigned to a PLC Text object and displayed accordingly. PLC system texts can also be displayed on the device screen with the use of PLC Text objects.

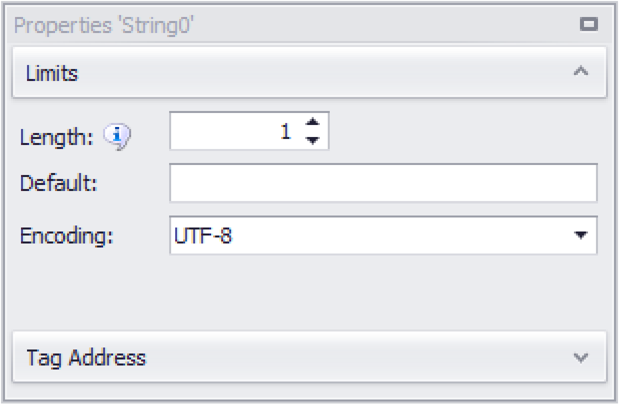
The corresponding text can be changed both on the device and on the PLC.
The string can be entered using alphanumeric keyboards.

The maximum text length is limited to 256 characters, i.e., the number of characters that a tag of data type STRING with 256 bytes can show.
The string tag can be defined in *Project Tree\Tags\<Tags>\Properties*, where it is also possible to specify whether a character will be represented by one or two bytes and whether a terminator character will be used. When a terminator character is used, one or two bytes must be subtracted accordingly. The actual maximum text length will depend on all these settings. If, however, the tag is instead defined as an array of chars with a smaller number of characters in the PLC, this will be the decisive limiting factor for the text length.

→ It is important to make sure that the number of characters defined in GALILEO and in the PLC is the same.

It is not possible to define different states for string tags.

Project Organization\Tags\



Properties 'String0'

Limits

Length: 1

Default:

Encoding: UTF-8

Tag Address

Figure 387: Properties 'String' with default value

Length

The length specifies the size of the string container measured in code units. The size of the container, in bytes, depends on the selected encoding. In the figure above, the container will have a size of 1 byte based on the selected encoding.

See also e.g. de.wikipedia.org/wiki/Unicode

or en.wikipedia.org/wiki/Character_encoding#Terminology

The size of the container (in bytes) must match the size of the container (in bytes) in the PLC.

String tag default

The default string will be used to initialize the visualization object when the device is working offline. This will be the case in the following situations:

- When there is no connection to the PLC because the device is working offline
- When displaying internal tags that are not used to communicate with the PLC

The default string will be an empty string by default.

Encoding

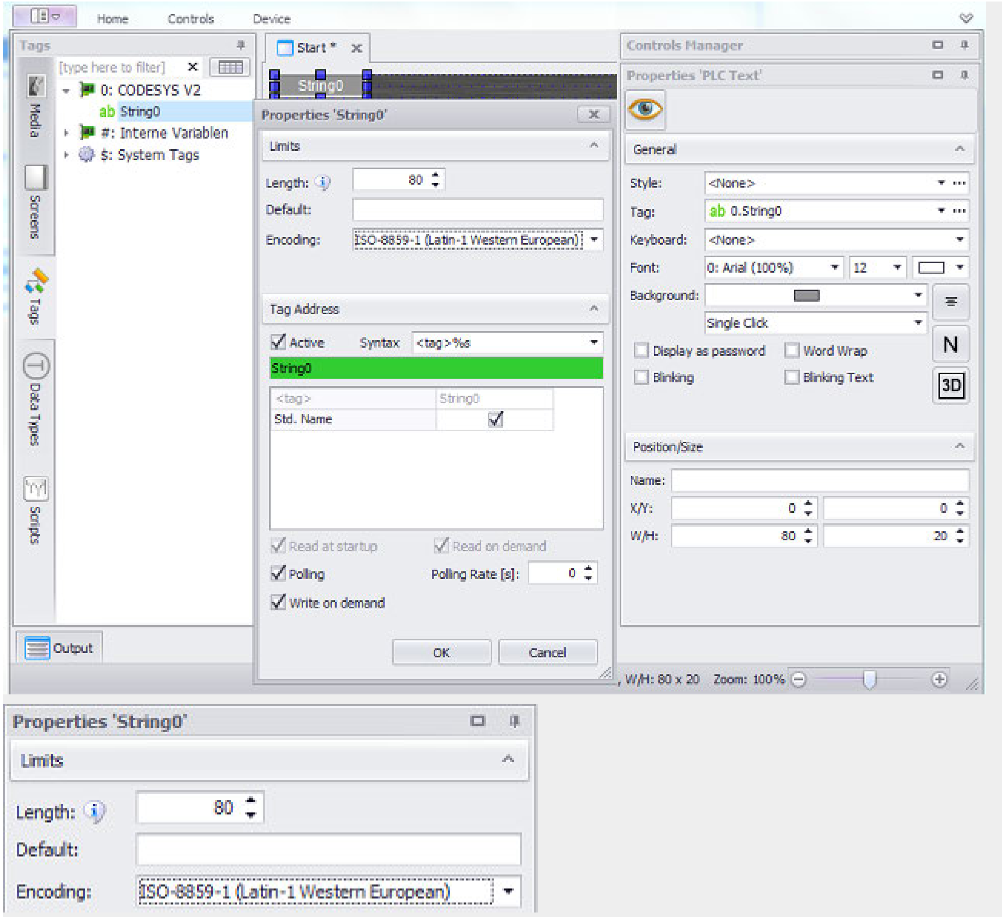
The selected encoding defines the code point mapping between the character set display for the user and the format that the PLC uses to store strings in a byte array. It is not a visualization property.

You can set the default encoding in the PLC Selection & Configuration dialog box.

Described in section:

→ section "Settings PLC Configuration", page 585

String tag example - CODESYS 2



Tags: [type here to filter]
- 0: CODESYS V2
  - ab String0
- #: Interne Variablen
- $: System Tags

Properties 'String0'

Limits

Length: 80
Default:
Encoding: ISO-8859-1 (Latin-1 Western European)

Tag Address

☑ Active Syntax <tag>%s

String0

<tag>	String0
Std. Name	☑

☑ Read at startup ☑ Read on demand
☑ Polling Polling Rate [s]: 0
☑ Write on demand

Style: <None>
Tag: ab 0.String0
Keyboard: <None>
Font: 0: Arial (100%) 12
Background:
Single Click
□ Display as password □ Word Wrap
□ Blinking □ Blinking Text

Name:
X/Y: 0 0
W/H: 80 20

W/H: 80 x 20 Zoom: 100%

Properties 'String0'

Limits

Length: 80
Default:
Encoding: ISO-8859-1 (Latin-1 Western European)

Figure 388: Default setting with a length of 80 characters and encoding that allows Ä, Ö, Ü, etc.

Tag address

Every tag with an address is transmitted in its own data block. Tag addresses can be direct addresses or symbolic addresses. These addresses are used as addresses for memory areas for tags in the PLC.

Depending on the specific communication method being used, the Syntax drop-down menu will have various address types available for selection, e.g., register range, pointer range, inputs, outputs. These address types will also depend on the PLC being used and the tag's data type.

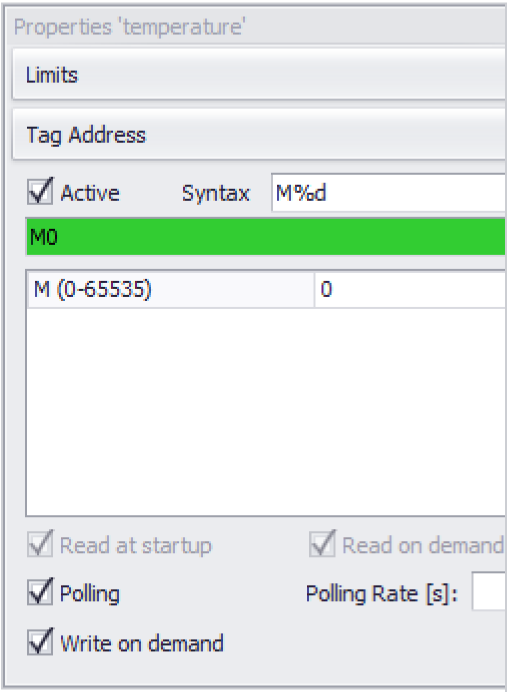
String tags defined as read/write tags (data direction) can either be read continuously with a polling rate or can be read in a defined manner controlled with the Control system tag.

PLCs with absolute addressing

Pointer / register ranges can be entered in the input fields. These fields only accept numeric values. The fields on the right of the editable fields will show the permitted number ranges.

PLCs with symbolic addressing

Uppercase letters, lowercase letters, numbers, underscore characters ("_"), and points (".") are permitted as text references.



Properties 'temperature'

Limits

Tag Address

☑ Active Syntax M%d

M0

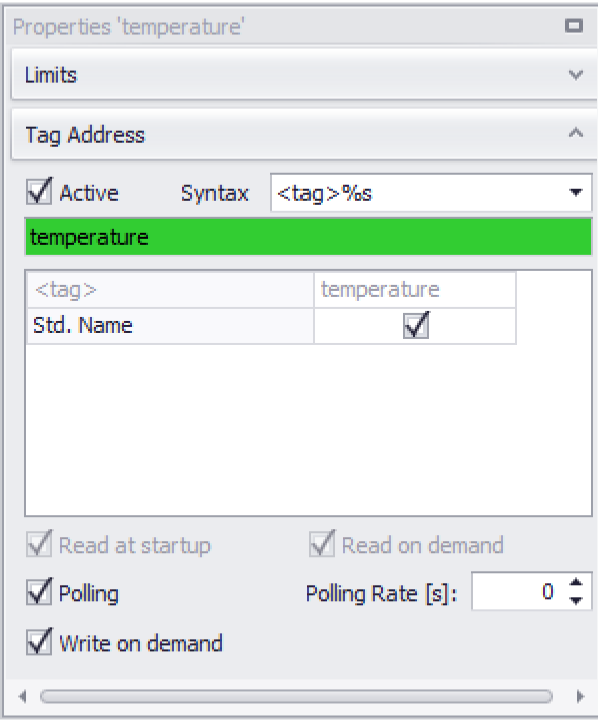
M (0-65535)	0

☑ Read at startup ☑ Read on demand

☑ Polling Polling Rate [s]:

☑ Write on demand

Figure 389: Pane for PLC with absolute addressing



Properties 'temperature'

Limits

Tag Address

☑ Active Syntax <tag>%s

temperature

<tag>	temperature
Std. Name	☑

☑ Read at startup ☑ Read on demand

☑ Polling Polling Rate [s]: 0

☑ Write on demand

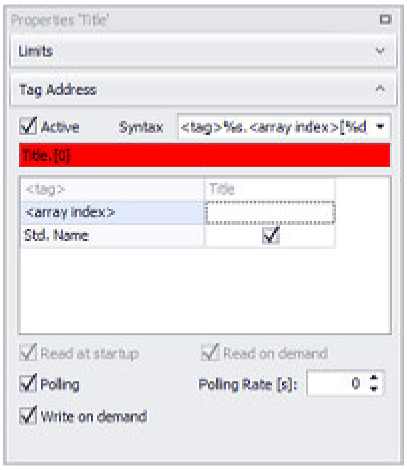
Figure 390: Pane for PLC with symbolic addressing

If you want tags to be transmitted to a PLC, the tags must be assigned the tag addresses in the corresponding PLC. This will be done automatically if a new PLC tag is created for a selected communication system in the *Project Tree\Tags* pane. For example: in Project Tree\Tags\ 0:CoDeSys XSOFT-CoDeSys2.

Accordingly, the Tag Address property will only appear for PLC tags, and not for internal tags or system tags.

The tag addresses for tags for different communication systems can be connected to each other using the gateway functionality – please refer to → "Using the gateway functionality to communicate with multiple PLCs", page 596.

→ Invalid addresses will result in communication errors on the device.



Properties 'Title'

Limits

Tag Address

☑ Active Syntax <tag>%s.<array index>[%d

Title.[0]

<tag>	Title
<array index>	
Std. Name	☑

☑ Read at startup ☑ Read on demand

☑ Polling Polling Rate [s]: 0

☑ Write on demand

Figure 391: Addressing for PLCs with symbolic addressing

Active

If this checkbox is enabled, this means that the tag has an active address. In other words, that the tag address has been assigned to the tag and is active. Once these two conditions are met, the tag will start being transmitted to the PLC in a data block. As for whether the tag will be transmitted with polling or in an event-based manner, this will depend on the additional settings configured with the checkboxes underneath.

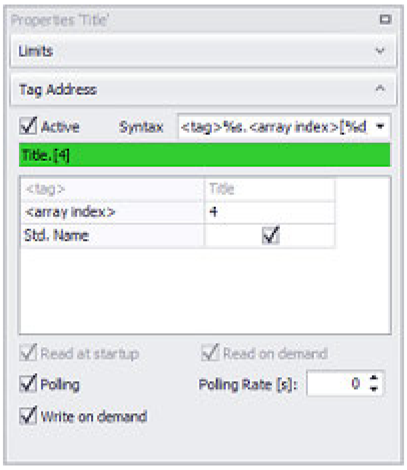
If the checkbox is disabled, this means that the tag does not have an active address. Even if the tag has been assigned a tag address, it will not be transmitted at runtime.

Syntax / <tag> / Title

Different PLCs use different syntaxes for data communication. The syntaxes available for selection in the drop-down menu will depend on the PLC communication systems selected for the project and on the communication system to which the relevant PLC tag has been assigned in the Tags pane. While the syntax specifies what the tag address must look like, there are other settings required for the tag address. These settings must be configured in the list underneath. Once they are configured, the tag address will be put together based on the selected syntax. This tag address will be shown in the field underneath.

Title.[0]

Missing data communication information GALILEO will be indicated by highlighting the tag address in read. In the example above, the Title tag has been assigned the symbolic address for an array. The array's index is required for this purpose. The highlighting will change to green only until a value for the index has been entered, e.g., "<array index> = 4".



Properties 'Title'

Limits

Tag Address

☑ Active Syntax <tag>%s.<array index>[%d

Title.[4]

<tag>	Title
<array index>	4
Std. Name	☑

☑ Read at startup ☑ Read on demand

☑ Polling Polling Rate [s]: 0

☑ Write on demand

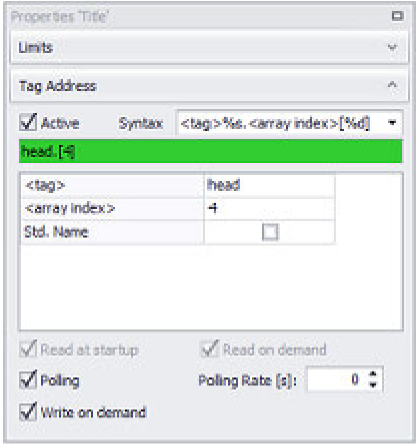
Figure 392: Address for PLC with string reference

Std. Name

If this checkbox is enabled, the name of the tag or structure will be used as the name for the symbolic tag. This is the usual approach. This option is only available with PLCs that use symbolic addressing.

If there are data communications with multiple communication systems, it is completely possible for it to be necessary to link a tag to a specific tag address that has a different name. In this case, this checkbox can simply be disabled. The name of the <tag> will then need to be entered into the corresponding field.

Project Tree\Tags\Context menu\Properties



Properties 'Title'

Limits

Tag Address

☑ Active Syntax <tag>%s.<array index>[%d]

head.[4]

<tag>	head
<array index>	4
Std. Name	□

☑ Read at startup ☑ Read on demand

☑ Polling Polling Rate [s]: 0

☑ Write on demand

For more information, please refer to the descriptions for the various communication systems. These descriptions list the syntax for tags of different data types for each PLC.

Read at startup

If this checkbox is enabled, the device will start reading at startup, provided the tag is relevant to the display.

If the checkbox is disabled, the device will not read the tag.

Write at startup

If this checkbox is enabled, the device will write the tag to the PLC at startup.

If the checkbox is disabled, the device will not write the tag to the PLC.

Read on demand

If this checkbox is enabled, the device will read the tag from the assigned communication system when an event occurs.

An Event can be:

- Save recipe
- Data synchronization during operation
- A request from the PLC in the control structure

If the checkbox is disabled, the device will not read the tag.

→ This setting is not used to display tag values! Tags must always be polled for this purpose instead.

→ section "Data communication during operation", page 594

Write on demand

Within this context, "demand" means event-based data communication.

If this checkbox is enabled, the device will write the tag to the assigned communication system when an event occurs.

An Event can be:

- value entry
- Data synchronization during operation
- A request from the PLC in the control structure

If the checkbox is disabled, the device will not write the tag.

Polling

If this checkbox is enabled, the tag will be cyclically read from the PLC at the interval set in Polling Rate.

In order for this reading to be carried out, the tag must already have been assigned a tag address and must be active.

If the checkbox is disabled, the PLC will not read it cyclically.

Regardless of whether data blocks are cyclically transmitted with the Polling property, the tag's value will also be transmitted in an event-based manner.

Please note that there are cases in which tags will always be polled – even if this property is not enabled. This will apply in the following cases:

- The tag is being used in graph blocks.
- The tag is being used in the gateway and transmitted between multiple PLCs.

Polling Rate [s]

Used to set the interval at which data will be read from the PLC. The polling rate range is from 0 to 6500 seconds.

If data changes slowly, a slower polling rate should be set so that other data can be read more frequently. If the rate is set to <0 s>, the data will be read as fast as possible depending on the amount of data being transferred, the PLC protocol being used, and the cycle time for the device and the PLC. If you are not sure which polling rate to set, it is recommended to set it to a value of <0 s> or <1 s>.

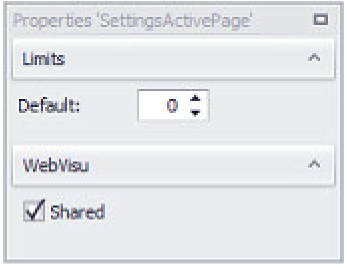
Internal tags

Internal tags are used exclusively within GALILEO. Because of this, an address is not assigned to them.

All internal tags have a WebVisu property regardless of their data type.

These internal tags are used to show how many users are logged in, for example.

Project Tree\Tags\Internal Tags\<x SettingActive>



Properties 'SettingsActivePage'

Limits

Default: 0

WebVisu

☑ Shared

Figure 393: Example showing internal bit tag SettingActive

Shared

If this option is enabled, there will only be one instance of the corresponding internal tag. If the internal tag's value is changed locally or through a web client, the exact same internal tag instance will be the one being changed. Moreover, the change in value will take effect immediately on all web clients and locally. If an internal tag is shared, it will be shown in green in the Tags pane.

If the option is disabled instead, the local application and the web clients will be using an internal tag with the exact same name. However, each web client, as well as the local application on the device, will be using its own instance of this tag. In this case, changes made to the tag's value will apply only to the instance in which the change is made.

When it comes to shared internal tags, there are certain limitations regarding their use for user management purposes.

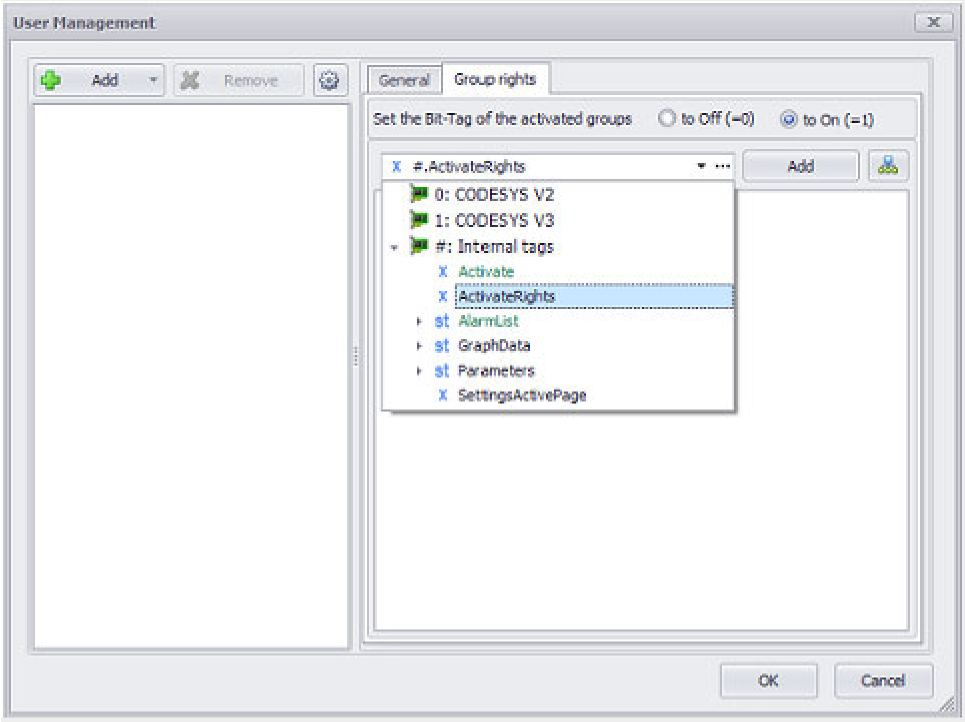
More specifically, the shared property must be disabled for the following specific user management scenarios:

1. Configuring user group rights
2. Saving user names
3. Saving group IDs

→ Do not use shared internal tags for user management purposes. If you do, the compiler will output an error message.

Example showing how to configure user group rights

Home\User Management\Group rights tab



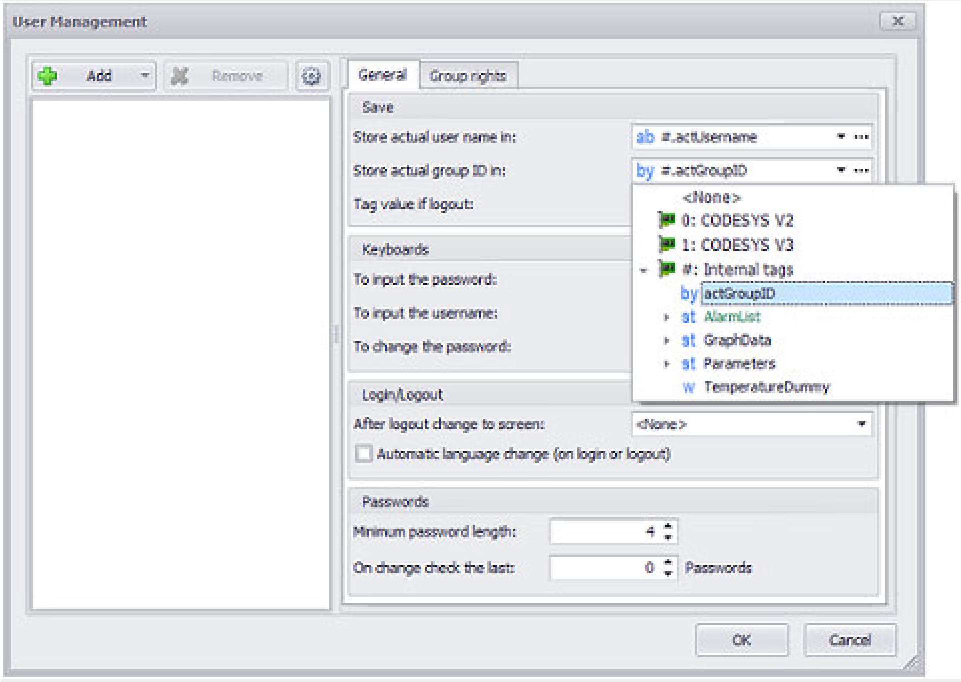
Set the Bit-Tag of the activated groups ○ to Off (=0) ● to On (=1)

#.ActivateRights

Figure 394: Example showing how to use the ActiveRights internal bit tag in order to configure group rights in the User Management dialog box

Example showing how to save user names and group IDs

Home\User Management\General tab



User Management

Save

Store actual user name in: ab #.actUsername

Store actual group ID in: by #.actGroupID

Tag value if logout:

Keyboards

To input the password:

To input the username:

To change the password:

Login/Logout

After logout change to screen: <None>

□ Automatic language change (on login or logout)

Passwords

Minimum password length: 4

On change check the last: 0 Passwords

Figure 395: Example showing how to use internal tags for user management purposes: actUsername in order to store the user name and actGroupID in order to store the group ID

4.6 System tags

System Tagsare identified with the sy data type and are a required component of every GALILEO project. These tags can be found in *Project Tree\ Tags* ⚙ *$: System Tags*.
System tags contain device information such as <HardwareVersion>.
This system tag makes it possible to configure the behavior of a GALILEO application differently for web clients and for the local application on the device.
Please note that system tags can be read, but cannot be written to or deleted.
The only exceptions are the sy Control, sy Status, and sy Date_Time system tags, which play an important role in data communications.

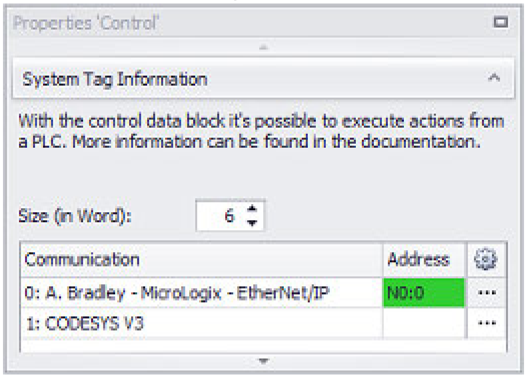
For a description of the three system tags, please refer to the → section "Control structure", page 603 , → section "Status data block", page 607, and → section "Synchronizing the time and date", page 612 sections.

The following section goes over the properties that can be set for the tags.

4.6.1 Data communication system tags

Control

The Control system tag is always polled with the "fast" speed. Tags that are polled are always read when the device is started and on demand. This applies to Control as well, which is why Control does not have any configurable read and write options, in contrast to other tags.



Properties 'Control'

System Tag Information

With the control data block it's possible to execute actions from a PLC. More information can be found in the documentation.

Size (in Word): 6

Communication	Address	⚙
0: A. Bradley - MicroLogix - EtherNet/IP	N0:0	...
1: CODESYS V3		...

Figure 396: Example showing the Control system tag when an A.Bradley communication system is being used

Size

This setting is used to define the size of the data block used to synchronize Control between the PLC and the device. The block size will be between 1 and 6 data words. For a more detailed description of the data block's format, please refer to → section "Control structure", page 603.

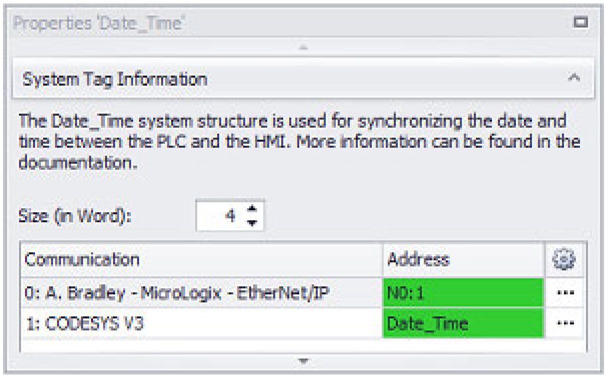
Date_Time

Date_Time is always polled with the "fast" speed. Tags that are polled are always read when the device is started and on demand. This applies to Date_Time as well, which is why Date_Time does not have any configurable read and write options, in contrast to other tags.

For a description of the Date_Time system tag and the data block's format, please refer to the → section "Synchronizing the time and date", page 612 section.

Size

This setting is used to define the size of the data block used to synchronize the date and time between the PLC and the device. The block size will be between 1 and 4 data words. For a description of the data block's format, please refer to the → section "Data format of Date_Time system tag", page 613 section.



Properties 'Date_Time'

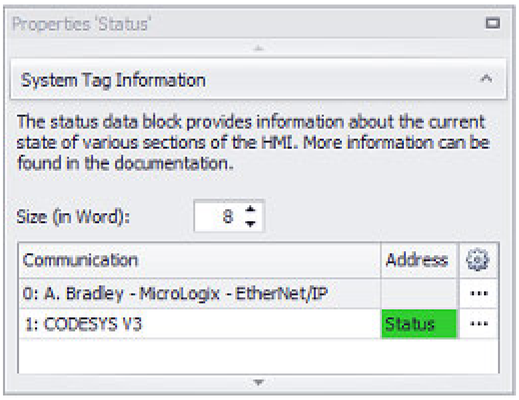
System Tag Information

The Date_Time system structure is used for synchronizing the date and time between the PLC and the HMI. More information can be found in the documentation.

Size (in Word): 4

Communication	Address	
0: A. Bradley - MicroLogix - EtherNet/IP	N0:1	...
1: CODESYS V3	Date_Time	...

State



Properties 'Status'

System Tag Information

The status data block provides information about the current state of various sections of the HMI. More information can be found in the documentation.

Size (in Word): 8

Communication	Address	
0: A. Bradley - MicroLogix - EtherNet/IP		...
1: CODESYS V3	Status	...

Size

This setting is used to define the size of the data block used to synchronize the state between the PLC and the device. The block size will be between 1 and 8 data words. For a detailed description of the data block's format, please refer to the → section "Status data block", page 607 section.

4.6.2 System tags for web visualization

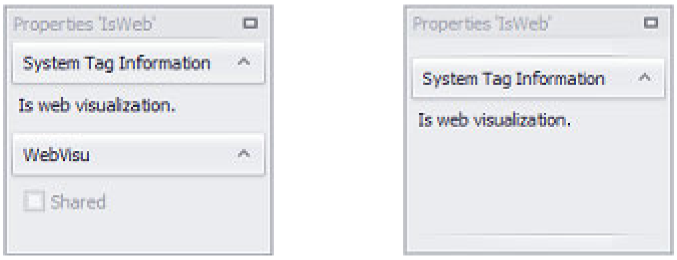
IsWeb

The IsWeb system tag will be enabled if you select the <Web only> or <Web and local> visualization mode in *Home\Settings\WebVisu tab*. An additional drop-down menu called WebVisu will show that the system tag is enabled. The system tag will not be set until at runtime.

If the IsWeb system tag is enabled, it will be instantiated for the local application on the device and for each web client. The IsWeb system tag instance for the local application on the device will be set to a value of <0>, while the instances for every web client application will be set to a value of <1>.

If the IsWeb system tag is not enabled, the IsWeb system tag instances will be set to a value of <0> for the local application on the device and for every web client application.

This system tag makes it possible to configure the behavior of a GALILEO application differently for web clients and for the local application on the device. For a description of how to use different screen navigation configurations, please refer to the → section "Different screen navigation configurations for web clients and for the local application on the device", page 1078 section.



Properties 'IsWeb'

System Tag Information

Is web visualization.

WebVisu

□ Shared

Properties 'IsWeb'

System Tag Information

Is web visualization.

Figure 397: left: IsWeb system tag when the Web only or Web and local visualization mode is selected
right: IsWeb system tag when the Local only visualization mode is selected

Shared

This property cannot be modified. It is shown only in order to indicate that an enabled IsWeb tag cannot be shared, as the system tag will be set to <0> locally on the device and to <1> on the web clients at runtime.

5. Data communication principles

The purpose of the Automatization Software GALILEO program is to enable users to easily, conveniently, and quickly implement human–machine interfaces. To this end, the relevant automation equipment is used to output and visualize the current data from a PLC or machine so that operators can see it. Likewise, operators are provided with input options so that they can directly access the PLC and control it as necessary. This makes communication between the device and the PLC an essential pillar in how the application works.

Because of this, GRS is subdivided into two components. The first component is responsible for visualization, while the second is responsible for data communication with PLCs. Both components work independently from each other and ensure fast response times.

The following data communication types are common and are implemented by GALILEO:

- Data polling
- Event-based data communication

Data polling

The term "polling" refers to the cyclical reading of data from a PLC.

A polling rate can be set for each tag in GALILEO. This polling rate is the interval at which the device will read data from the PLC and can be set to a value between 0 and 5600 seconds. A value of 0 seconds means that the tag will be polled as fast as possible. This polling rate can be set in the tag properties, in the Polling Rate field.

Event-based data communication

The term "event-based communication" can refer to read or write operations. As the name implies, however, these operations will only be carried out when an event occurs. Examples of events include the device starting, a function being triggered by the device operator tapping a button on the device screen, and a function being triggered by the PLC.

The following pages contain detailed information explaining how these types of data communication are implemented in GALILEO and what needs to be done in order to ensure that data communication will work properly.

5.1 Select Communication

The term "communication" or "communication method" refers to data communication with a PLC from a specific manufacturer or a machine via a defined hardware interface on the touch panel using a specific data protocol.

The number and type of hardware interfaces will depend on the device model on which the application is being run. Regardless, data communications can only be established with a maximum of eight PLCs.

Data communication with PLCs from a variety of manufacturers is supported, including Allen-Bradley, Siemens, and Beckhoff.

Various data protocols are supported, including TCP/IP, PROFIBUS DP, and CoDeSys.

Projects created in GALILEO can be configured for various PLCs.

→ The PLC should be selected at least before the data blocks are addressed since the address syntax required depends on the PLC used.

Existing projects can be reconfigured for a different PLC by selecting the new PLC in the PLC Communication dialog box opened by clicking on PLC Communication.

The tags need to be checked after making changes to the communication method.

Only PLCs can be selected that have the same data block size (data blocks). Once the PLC is changed, an address needs to be assigned again to each tag if a different syntax is required.

The tag addresses available later on for assigning addresses to tags will depend on the driver selected.

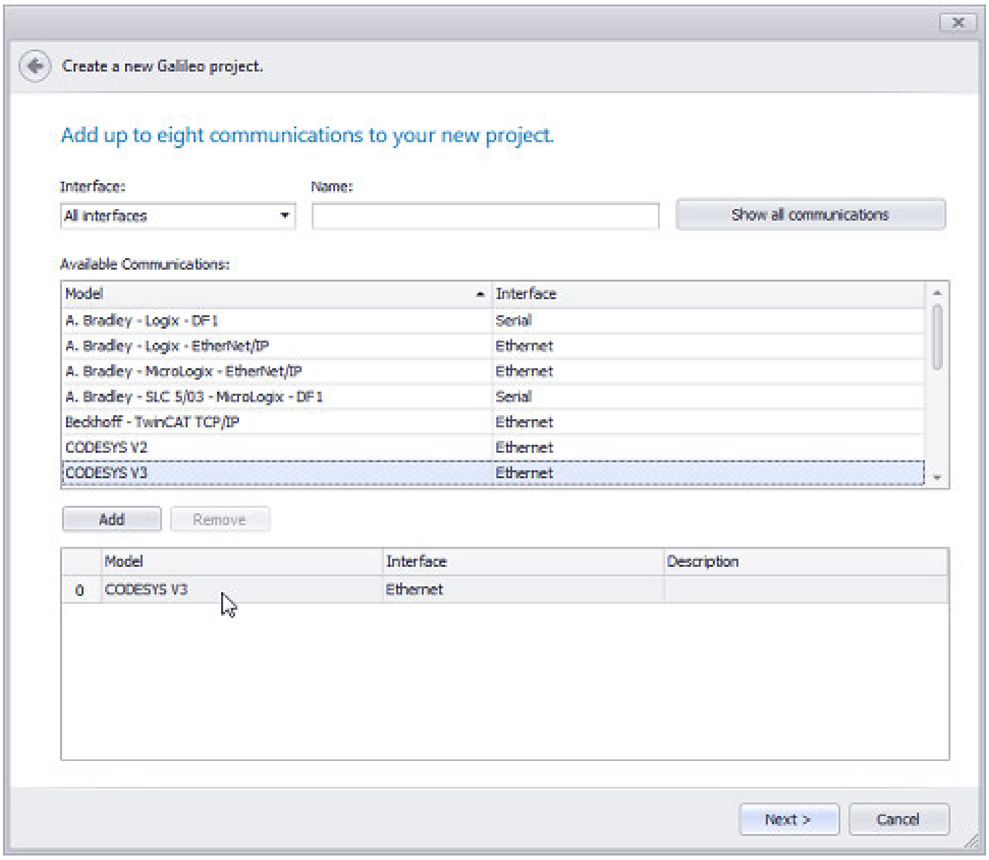
5.1.1 Selecting a communication system while creating a project

When creating a new project, the wizard will ask you to select the communication systems you want for it; for more information, please refer to → section "Substep 3. Selecting communication methods", page 162

Use the <Add up to eight communications to your new project> dialog box to select one or more communication systems for the project. What exactly is a "communication system"?

The term "communication" or "communication method" refers to data communication with a PLC from a specific manufacturer or a machine via a defined hardware interface on the touch panel using a specific data protocol.

Project Management\ New Project ...



Create a new Galileo project.

Add up to eight communications to your new project.

Interface: All interfaces

Name:

Show all communications

Available Communications:

Model	Interface
A. Bradley - Logix - DF1	Serial
A. Bradley - Logix - EtherNet/IP	Ethernet
A. Bradley - MicroLogix - EtherNet/IP	Ethernet
A. Bradley - SLC 5/03 - MicroLogix - DF1	Serial
Beckhoff - TwinCAT TCP/IP	Ethernet
CODESYS V2	Ethernet
CODESYS V3	Ethernet

Add Remove

	Model	Interface	Description
0	CODESYS V3	Ethernet	

Next > Cancel

Figure 398: New Project Wizard, step 3 – adding communication methods to the project

The list contains the configured PLC protocols for this project.

The Interface field can be used to open the drop-down menu containing all available interfaces. The number and type of interfaces will depend on the selected device model and are described in greater detail in the → section " Select panel", page 221 section.

Interfaces
Used to filter the list of communication systems shown under "Available Communications" by physical interface.

Name
Used to filter the list of communication systems by name.

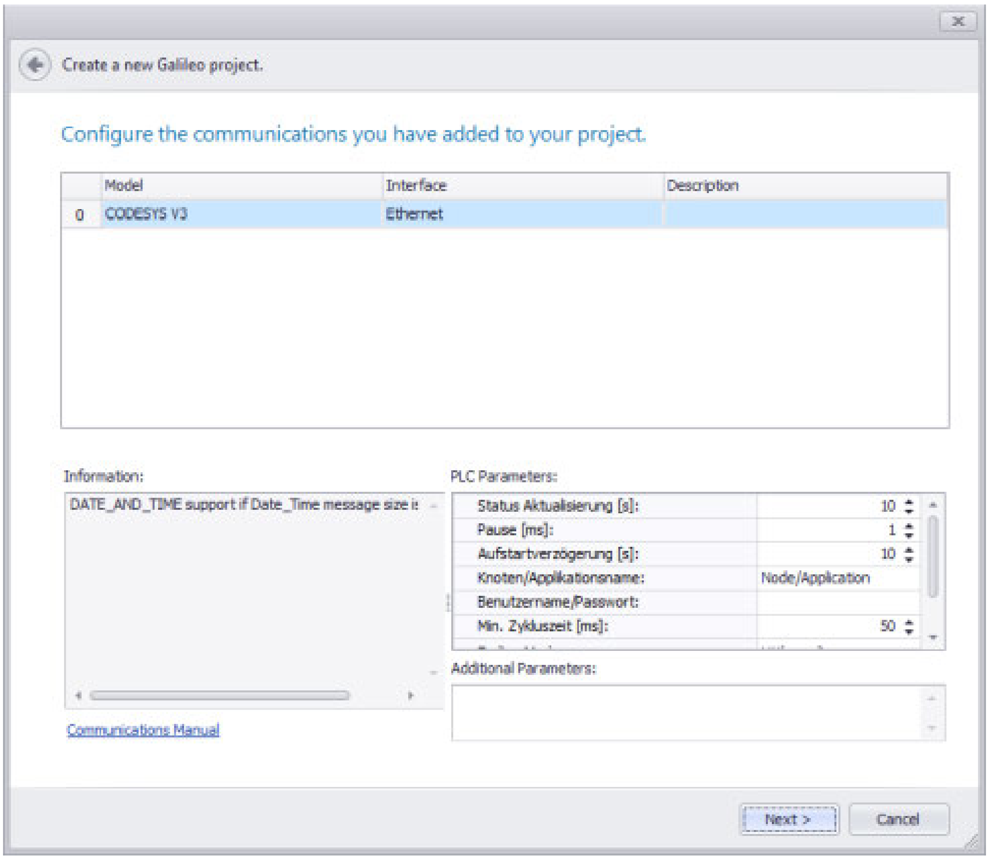
Show all communications
Clicking on this button will reset the Interface and Name filters.

Available Communications
Selection list showing all available communication systems or the communication systems that meet the filter criteria defined above.

To add communication systems to the lower pane, select a communication system and click on Add (the added communication systems will be numbered 0 to 7).
To remove them, select a communication system and click on Remove.

The added communication systems can already be configured while creating a project, i.e., the corresponding PLC parameters can be configured during the process.

Project Management\ New Project ...



Create a new Galileo project.

Configure the communications you have added to your project.

	Model	Interface	Description
0	CODESYS V3	Ethernet	

Information:

DATE_AND_TIME support if Date_Time message size is

Communications Manual

PLC Parameters:

Status Aktualisierung [s]:	10
Pause [ms]:	1
Aufstartverzögerung [s]:	10
Knoten/Applikationsname:	Node/Application
Benutzername/Passwort:	
Min. Zykluszeit [ms]:	50

Additional Parameters:

Figure 399: New Project Wizard, step 4 – configuring communication methods

Information

This pane will show a brief summary of the relevant information for the selected communication method and point out any special characteristics that should be taken into account.
This information is only available in English.

For further reading on the various communication systems, please refer to the → "Further usage information", page 1143 appendix or to the system descriptions provided by the relevant manufacturer.

PLC Parameters

The PLC parameters shown will depend on the communication method selected. The fields will only be editable if the selected protocol requires the corresponding setting to be configured.

MPI/Profibu- This field is used to specify the station number of the touch panel in a

s Station No.
network.

Send buffer
This field defines the size of the send block (MMI → PLC) including protocol header in the number bytes.

Refresh State [s]
Defines the maximum refresh time for the state structure, in seconds. If there are no device state changes within this refresh time, the state structure will still be sent to the PLC. The maximum status refresh time is 255 seconds. The shorter the refresh time, the greater the load on data communication between the device and the PLC. This can result in the state structure being the only data transmitted.

→ Rule:
Since only one data block is normally transmitted per PLC cycle, the selected refresh time should be at least five times greater than the PLC's cycle time.

Startup Delay [s]
When runtime starts, all communication messages will be suppressed for the time set here. This may be necessary if the touch panel starts faster than the PLC, which would result in a large number of error messages being generated.

Endian Mode
This property is used to define the bit order – when bit numbers are counted in ascending order from the least significant bit to the most significant bit, the order is referred to as "little-endian format."
In contrast, when bit numbers are counted in ascending order from the most significant bit to the least significant bit, the order is referred to as "big-endian format."

For example for the number 132

Bit order								
Little Endian	7	6	5	4	3	2	1	0
Big Endian	0	1	2	3	4	5	6	7
Bit order	1	0	0	0	1	0	0	0
Decimal $_{value}$	128	64	32	16	8	4	2	1

Baud Rate
Select the permissible transmission rates between device and PLC.

Memory alignment inside structures

Receive Buffer
This field can be used to define the size of the receive block (PLC → MMI), including the protocol header, in bytes.

→ For a description of the required settings for the PLC parameters not listed here, please refer to the documentation for the protocol or communication card being used.

5.2 Settings PLC Configuration

PLC Selection & Configuration

The PLC Selection & Configuration dialog box can be used to configure an existing project for additional and/or other PLCs.

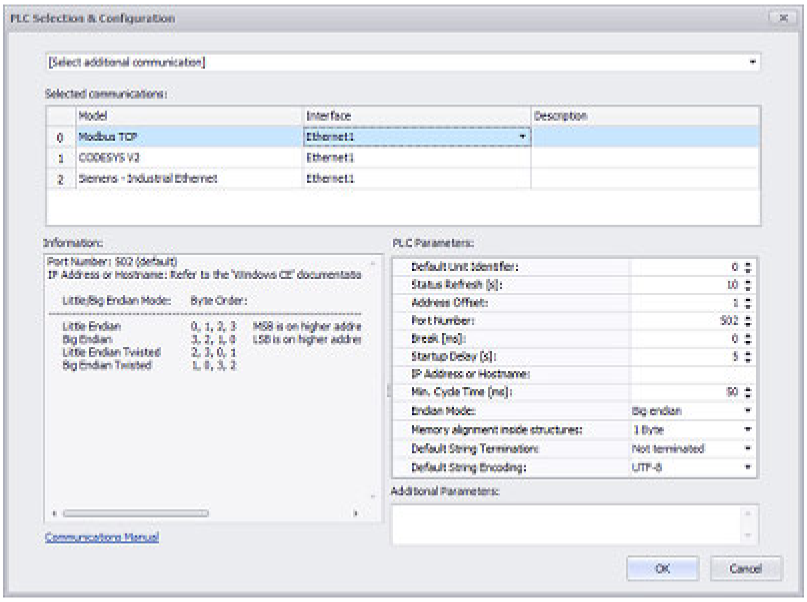
The term communication or "communication method" refers to data communication with a PLC from a specific manufacturer or a machine via a defined hardware interface on the device using a specific data protocol.

Every project can be configured for up to eight different communication methods via PLC interfaces.

The OPC Unified Architecture communication protocol is available in GALILEO Version 10.5 and higher.

The PLC should be selected at least before the addresses are assigned to the tags, as the address syntax required varies from PLC to PLC. This means that, once addresses have been assigned, you will only be able to switch between PLCs that have the same data blocks. Otherwise, you will need to change the addresses.

Device tab\ PLC Communication



PLC Selection & Configuration

[Select additional communication]

Selected communications:

	Model	Interface	Description
0	Modbus TCP	Ethernet1	
1	CODESYS V2	Ethernet1	
2	Siemens - Industrial Ethernet	Ethernet1	

Information:

```
Port Number: 502 (default)
IP Address or Hostname: Refer to the 'Windows CE' documentatio

   Little/Big Endian Mode:   Byte Order:
 ------------------------------------------------
   Little Endian             0, 1, 2, 3   MSB is on higher addre
   Big Endian                3, 2, 1, 0   LSB is on higher addre
   Little Endian Twisted     2, 3, 0, 1
   Big Endian Twisted        1, 0, 3, 2
```

Communications Manual

PLC Parameters:

Default Unit Identifier:	0
Status Refresh [s]:	10
Address Offset:	1
Port Number:	502
Break [ms]:	0
Startup Delay [s]:	5
IP Address or Hostname:	
Min. Cycle Time [ms]:	50
Endian Mode:	Big endian
Memory alignment inside structures:	1 Byte
Default String Termination:	Not terminated
Default String Encoding:	UTF-8

Additional Parameters:

Figure 400: PLC Selection & Configuration dialog box

[Select additional communication]	Used to select a communication system The drop-down menu will show all the communication systems supported by the selected panel model. Clicking on a communication system will add it to the Selected communications list.

Selected communications (0..7)	The list will show the PLC communication systems that have already been set up and are active – if you want, you can add a description for each of them. Moreover, you can make changes to the individual communication systems by changing the corresponding values in the individual Model and Interface cells. For example: serial interfaces, RS485 (for panel) and COM x (for GalileoOpen) If you have selected (0..7) communication systems, a prompt to this effect will appear. Please note that you can enter any text you want into the Description field.
⊗	Used to indicate conflicts, e.g., a communication system that has already been created but is not enabled
Info:	This pane will show a brief summary of the relevant information for the selected communication method and point out any special characteristics that should be taken into account. This information is only available in English.
PLC Parameters	The PLC parameters shown will depend on the communication method selected. The fields will only be editable if the selected protocol requires the corresponding setting to be configured.
Additional parameters	This box can be used to enter specific parameters for the communication system.

The tags need to be checked after making changes to the communication method.

The Communications Manual link will take you to the manual for the communication systems.

You can open the Communications User Manual directly with ▤ \ ❓ Help and Tools as a .PDF document in English.
For further usage information on the various communication systems, please refer to the system descriptions provided by the relevant manufacturer.

5.3 Transmitting tags

GALILEO takes tags and intelligently groups them into data packets so that communication can be as efficient as possible. The specific way in which the data packets are put together will depend on the protocol being used.

It is usually considered a good idea to group tags used within a screen into arrays or structures. In contrast to older GALILEO versions, this approach is only recommended for specific cases.

In order to be able to implement event-based data communication, a control data block, status data block, and date/time are transmitted in addition to the normal data blocks. Later sections explain how these system tags are used.

Which tags are transmitted?

GALILEO transmits data in the following blocks:

- **Tags** of data type bit, byte, word, double word, float, array, structure, custom data type
- **Control structure**; fast polling always
- **State structure**; event-based writing, cyclical writing
- **Date/Time**; polling / event-based; read/write

If there are multiple PLCs connected to the device, all of them must have the same control structure and state structure sizes.

When are tags transmitted?

From device to PLC: Generally speaking, the device will only write tags in an event-based manner. For more information on which events are used for this purpose, please refer to → "Data communication during operation", page 594

From PLC to device: The device will read tags from the PLC if polling has been configured, or on demand if the PLC triggers an event.

The polling rate use to read data from the PLC can be configured using the properties for the various tags. By default, all tags to which an address has been assigned will be configured with <fast> polling. This means that the Polling Rate setting in their properties will be set to <0> seconds. When the polling rate is set to this value, the device will attempt to read the data as fast as possible.

Generally speaking, the more tags configured with <fast> polling, the more resources this will take, resulting in a slower cyclical rate for reading individual tags. This effect will not be identical for all communication systems. In fact, the actual cyclical read rate will heavily depend on the protocol being used.

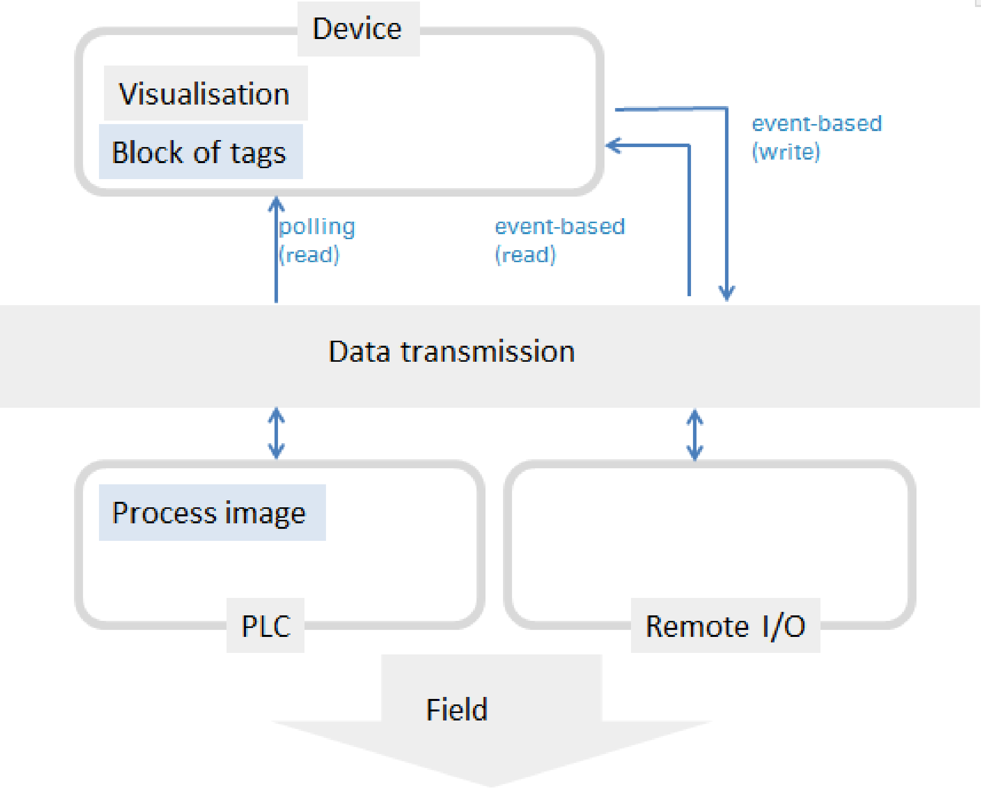
The GRS will use the tags used in the current screen to determine which tags have to be read.

You can also set a specific polling rate (1 second, 10 seconds, etc.) for every tag with the "read" or "write" data direction. This is primarily required for tags that only change slowly, e.g., water levels, temperatures.

This has the advantage that all other Tags of the current screen with the type <fast> can be read accordingly faster.

Follow the general rules outlined below:

- Only use polling when necessary.
- Set a Polling Rate = <0> for tags that change frequently (e.g., unit counter).
- If you are not sure what the cycle time should be, use a Polling Rate = <1>.



5.4 Data Synchronization

The term "data synchronization" refers to ensuring that all modules connected to the device have the same data (presetting). Data that should not be lost when the system is switched off should always be stored in the PLC's non-volatile memory.

If data synchronization is completed successfully, bit 1 of the 1st data word in the state structure will be set to 1. Please refer to → section "1. Data Word", page 608 as well.

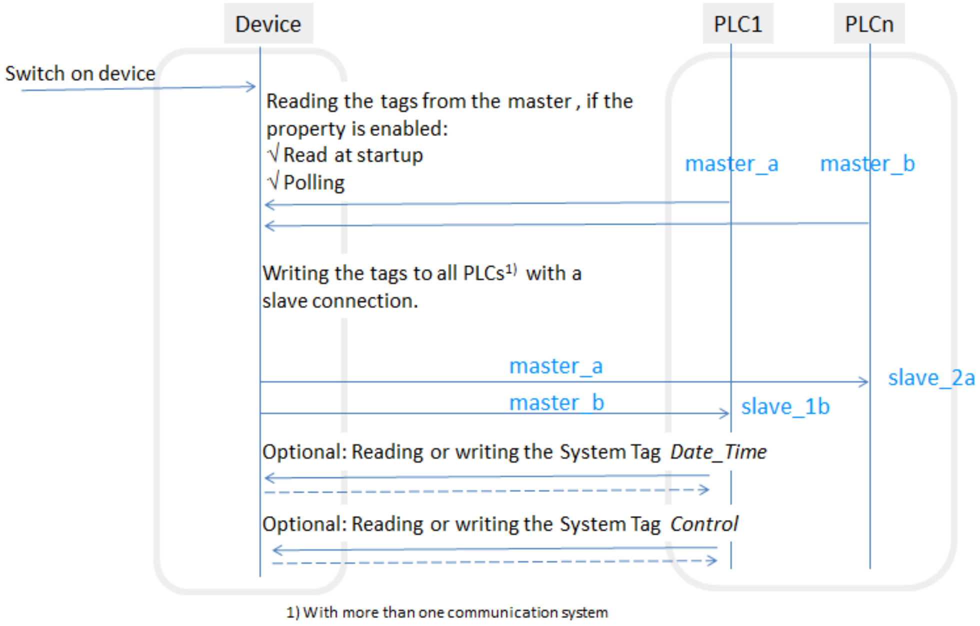
→ In scripts the state of this bit can be read with the function → "OnPresetOK", page 992.

5.4.1 Data synchronization when the device is switched on

Once the device is switched on, the data synchronization process will be automatically started during the startup phase. During this process, the device will read all the tags for which the Read at startup property is enabled once from the PLC. The corresponding data direction is specified in the properties for the tags as well. If the Polling property is enabled, the Read at startup property will be automatically enabled as well.

The PLC that will serve as the master is set already when tags are created.

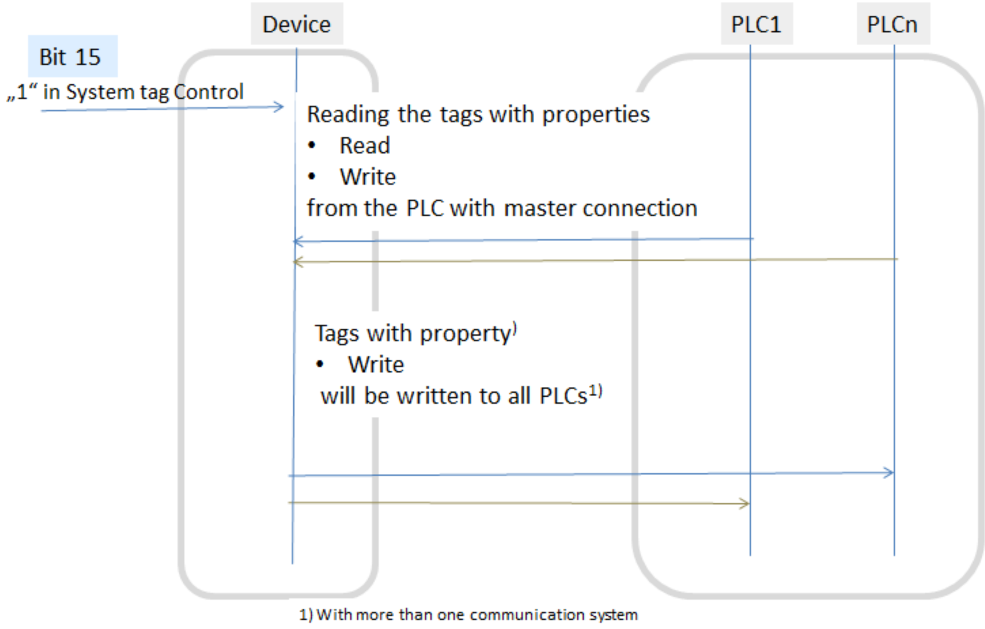
If more than one communication system is being used, the tags from a master can be transmitted to other communication systems' tags using the gateway functionality. In this case, the tags in the other communication systems will be referred to as "slaves." During data synchronization, the tags in the master that have the Read at startup property enabled will be read from the master and written to all other PLCs with a slave connection. The properties of the tags in these slave PLCs are irrelevant within this context.


Device
PLC1
PLCn
Switch on device
Reading the tags from the master , if the property is enabled:
√ Read at startup
√ Polling
master_a
master_b
Writing the tags to all PLCs1) with a slave connection.
master_a
master_b
slave_2a
slave_1b
Optional: Reading or writing the System Tag Date_Time
Optional: Reading or writing the System Tag Control
1) With more than one communication system

5.4.2 Event-based data synchronization

An event can be used to trigger the data synchronization process during normal operation. This event is triggered by setting bit 15 in the Control system tag's 1st data word. During this data synchronization process, the device will read the tags with the Read at startup property enabled from the PLC. If more than one communication system is being used, all tags with the Read at startup property will be read from the master PLC and transmitted to the slave PLC(s). The properties of the tags in these slave PLCs are irrelevant within this context.

Every single communication system has its own Control system tag, meaning any of them can trigger the data synchronization event.



Reading the tags with properties

- Read
- Write

from the PLC with master connection

Tags with property)

- Write

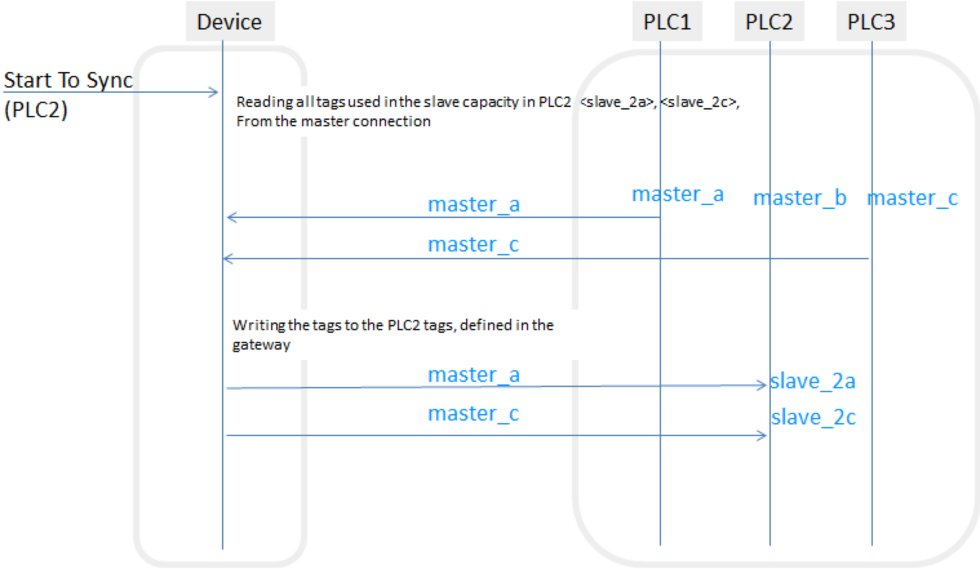
will be written to all PLCs[1)]

1) With more than one communication system

5.4.3 Slave data synchronization

The Start Sync to Connection function can be used to trigger a slave data synchronization process for a specific communication system. This means that all the tags used in a slave capacity in the selected communication system will be read from the masters and transmitted to the slave. Reading

Example



Reading all tags used in the slave capacity in PLC2 <slave_2a>,<slave_2c>,
From the master connection

Writing the tags to the PLC2 tags, defined in the gateway

5.4.4 Events which cause Data Synchronization

Event-based data synchronization will only be carried out if the <Read on demand> property is enabled for the corresponding tags.

The following events can trigger a data synchronization process:

- The device switching to a screen that runs a start script that uses tags.

5.5 Data polling

The easiest and most reliable communication method for both design engineers and users is tag polling. It is an easy method to use because it requires virtually no planning for specific communication systems, and a reliable method to use because it ensures that data will be updated constantly so that it does not have any discrepancies relative to PLC data.

Polling should only be avoided in cases in which the data throughput required for the user cannot be maintained or in which the device is unable to meet the increased communication requirements. The reason for this is usually that each individual tag (bit, byte, word, double word) is polled individually by the PLC.

→ In the case of PLCs that work with an image table, it is advisable to have the tags polled on a screen be closely located to each other on the PLC's physical memory. This will make it possible to read the data with fewer operations. In most of these cases, whether the tags are declared as arrays, structures, or individual tags in GALILEO will be completely irrelevant.

This means that GRS is designed to provide maximum performance precisely when it comes to these communication requirements, providing design engineers with the freedom to design the projects they want without having to worry excessively about how communication will be implemented.

In addition, GRS is able to communicate with various PLCs or bus systems via a maximum of eight channels.

Modifying a GALILEO-project for a different PLC that uses the same communication protocol can be done in a manner of minutes. If, however, a different protocol is used, the addresses for the tags may need to be changed in certain circumstances.

When will be read?

Tags will be polled in the following cases, provided the <Polling> property checkbox has been enabled for them:

- The tag is used in the screen currently being displayed.
- Tag is used in a graph data block and the graph block is started.
- The tag has addresses on two or more communication systems (achieved with the use of the gateway functionality)
- The <On demand> property is enabled and a recipe is being saved

→ "Transmitting tags", page 587

5.6 Data communication during operation

After data is successfully synchronized, tags will normally be polled during ongoing operation and, if the device operator enters input on the device, the device will transmit the modified tags to one or more PLCs.

It is also possible to run a data synchronization process during ongoing operation – please refer to → "Event-based data synchronization", page 590.

Polling

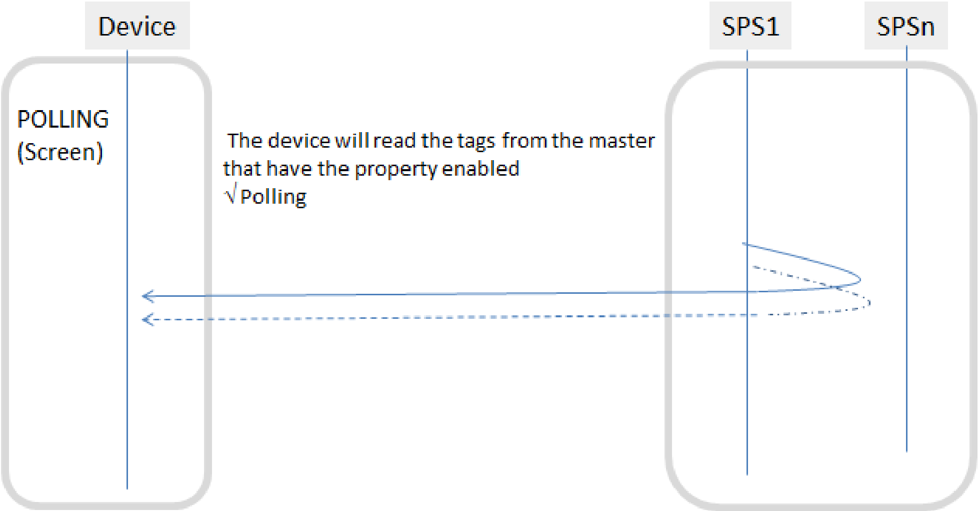
In order for polling to be used, the <Polling> property needs to be enabled for the corresponding tag.

To keep data communication down to the required minimum, tags will only be polled if they are currently relevant. Within this context, "currently relevant" refers to tags that are currently being visualized with visualization objects on the current screen on the device. There is one exception: tags in a graph block that are visualized with a graph visualization object for which the graph has been started. These tags will be polled even if the graph is not linked to the current screen. Please note that there are visualization objects that can be placed on a screen but that are defined as not being visible! This means that there can be tags that are not being shown on the screen but that will still be polled.

Tags will be polled in the following cases, provided the checkbox for the <Polling> property is enabled:

- The tag is used in the screen currently being displayed.
- Tag is used in a graph data block and the graph block is started.

If multiple communication systems are being used and tags are linked to multiple systems using the gateway functionality, the device will read the tags that have the <Polling> property enabled from the master and automatically write them to all tags linked in the slave role in the gateway. The properties for these slave tags are irrelevant within this context.


Device
SPS1
SPSn
POLLING
(Screen)
The device will read the tags from the master
that have the property enabled
√ Polling

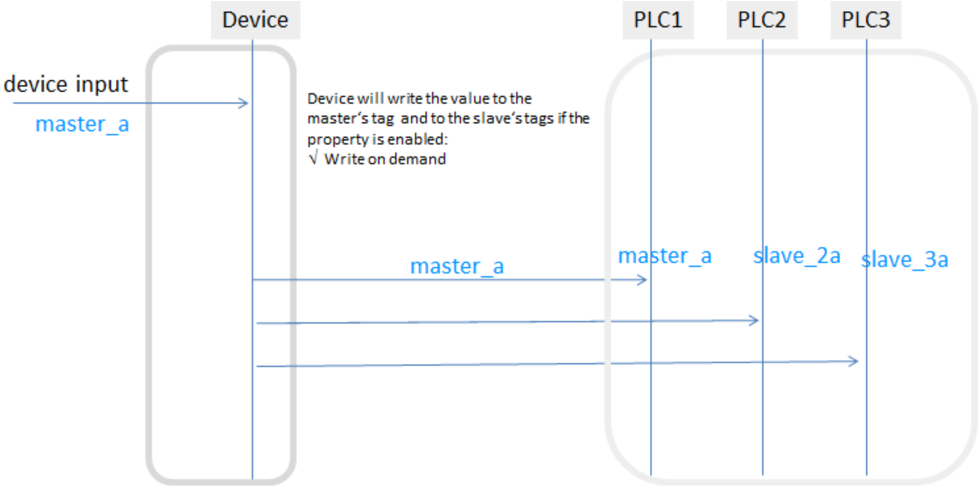
Device input

In addition to polling, event-based data communication is also used during ongoing operation. One possible event is input provided by an operator on the device. This means that if an operator uses a visualization object to enter a value on the device, the device will write the value to the PLC if the Write on demand property is enabled for the corresponding tag.

If multiple communication systems are being used and a tag is linked to tags in other systems using the gateway functionality, the device will write the value to the master's tag and to the slaves' tags. The properties for the slaves' tags are irrelevant within this context. If, for example, tag <master_a> is modified as a result of the device operator entering input, the device will write the tag to master PLC1 and to the tags linked as slave tags in the gateway, i.e., <slave_2a> in PLC2 and <slave_3a> in PLC3, provided that the Write on demand property for the <master_a> tag is enabled.

Tags will be polled in the following cases, provided the Polling property is enabled:

- The checkbox for the tag's <Polling> property is enabled
- The tag is used in the screen currently being displayed.
- Tag is used in a graph data block and the graph block is started.
- The tag has addresses on two or more communication systems (achieved with the use of the gateway functionality) (bridge)


Device
PLC1
PLC2
PLC3
device input
master_a
Device will write the value to the master's tag and to the slave's tags if the property is enabled:
√ Write on demand
master_a
master_a
slave_2a
slave_3a

For a description of additional events, please refer to the → section "Data Synchronization", page 589 section.

→ "Tag address", page 569
→ "Data Synchronization", page 589

5.7 Using the gateway functionality to communicate with multiple PLCs

If multiple PLCs are involved in the overall system's data communication, the gateway functionality can be used to define the data communication direction for the various tags. The following applies within this context:

The device will write master tags to slave tags.

Every tag in a project is either declared as <Internal> or assigned to a communication system so that it is automatically a master tag. The gateway functionality can be used to define whether master tags should be written to tags in other PLCs. In this case, the tags in the other PLCs will be slave tags.

The links between master tags and slave tags in the gateway will be taken into account with every data synchronization, regardless of whether it is polling-based or event-based. This ensures that the data on the modules connected to the device will match.

Master

This cell is used to define the master for the tag. It will only be relevant if multiple communication systems are being used, in which case it will define the PLC from which the device will read the data when the latter is being synchronized. For a data synchronization description, please refer to the → section "Data Synchronization", page 589 section

slave

These cells are used to define one or more slaves for the master tag. The cells are only relevant when using multiple PLC connections, and define the PLCs to which the device will write the master tag when synchronizing data.

Example

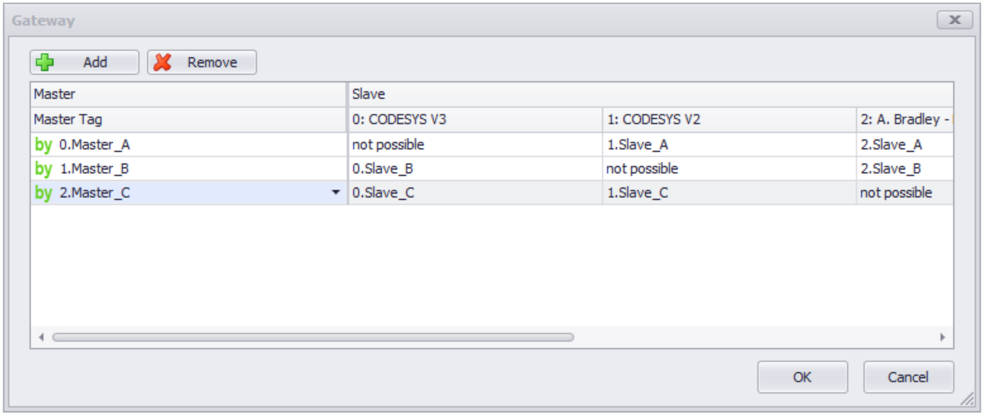
Say that you have some tags linked as shown in the example below. This means that:

The device will read tag Master_A from the 0:CODESYS V2 communication system and write it to tag 1.Slave_A in communication system 0:CODESYS V2 and tag 2.Slave_A in communication system 2:A. Bradley.

The device will read tag Master_B from the 1:CODESYS V2 communication system and write it to tag 0.Slave_B in communication system 0:CODESYS V2 and tag 2.Slave_B in communication system 2:A. Bradley.

The device will read tag Master_C from the 2:A. Bradley communication system and write it to tag 0.Slave_C in communication system 0:CODESYS V2 and tag 1.Slave_C in communication system 1:CODESYS V2.

Home\Gateway



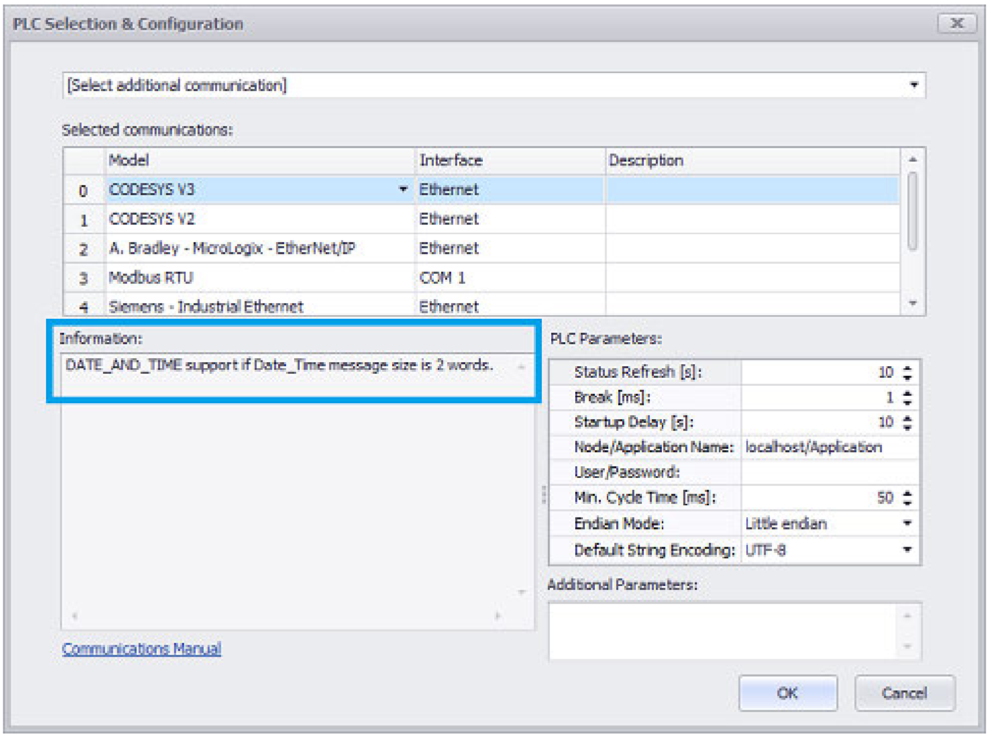
Master	Slave		
Master Tag	0: CODESYS V3	1: CODESYS V2	2: A. Bradley -
by 0.Master_A	not possible	1.Slave_A	2.Slave_A
by 1.Master_B	0.Slave_B	not possible	2.Slave_B
by 2.Master_C	0.Slave_C	1.Slave_C	not possible

Figure 401: Gateway dialog box with three communication systems CODESYS V2, CODESYS V2, A. Bradley

5.7.1 Brief summary of relevant information for the communication system

This pane will show a brief summary of the relevant information for the selected communication method and point out any special characteristics that should be taken into account.
This information is only available in English.

In the example below, the pane points out that the date and time can be synchronized and that a block size of 2 data words must be configured for the Date_ Time system tag.


PLC Selection & Configuration
[Select additional communication]
Communications Manual
OK
Cancel

Selected communications:

	Model	Interface	Description
0	CODESYS V3	Ethernet	
1	CODESYS V2	Ethernet	
2	A. Bradley - MicroLogix - EtherNet/IP	Ethernet	
3	Modbus RTU	COM 1	
4	Siemens - Industrial Ethernet	Ethernet	

Information:

DATE_AND_TIME support if Date_Time message size is 2 words.

PLC Parameters:

Status Refresh [s]:	10
Break [ms]:	1
Startup Delay [s]:	10
Node/Application Name:	localhost/Application
User/Password:	
Min. Cycle Time [ms]:	50
Endian Mode:	Little endian
Default String Encoding:	UTF-8

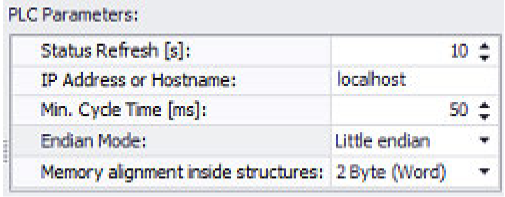
Additional Parameters:

5.7.2 PLC Parameters

The number and type of PLC parameters will vary based on the selected communication system. The settings in the PLC Parameters pane can be used to configure the required communication parameters for the protocol being used. The required settings can be gathered from the description for the corresponding communication system.

The following PLC parameters can be configured for all communication systems:

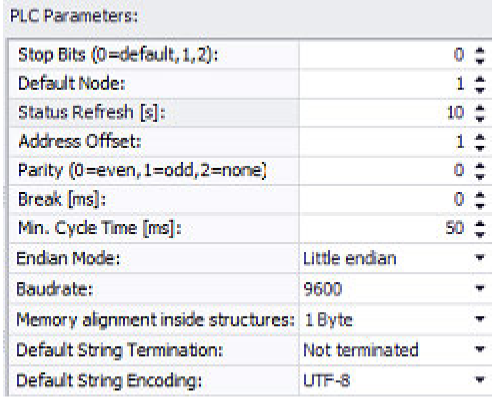
- Refresh State [s]
- IP address or hostname
- Min. Cycle Time [ms]
- Endian Mode
- Memory Alignment



PLC Parameters:

Status Refresh [s]:	10
IP Address or Hostname:	localhost
Min. Cycle Time [ms]:	50
Endian Mode:	Little endian
Memory alignment inside structures:	2 Byte (Word)

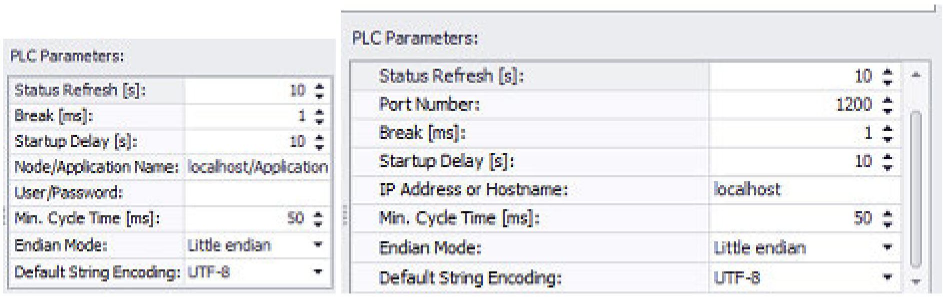
Figure 402: PLC parameters for the A.Bradley - Logix - EtherNet/IP communication system



PLC Parameters:

Stop Bits (0=default, 1, 2):	0
Default Node:	1
Status Refresh [s]:	10
Address Offset:	1
Parity (0=even, 1=odd, 2=none)	0
Break [ms]:	0
Min. Cycle Time [ms]:	50
Endian Mode:	Little endian
Baudrate:	9600
Memory alignment inside structures:	1 Byte
Default String Termination:	Not terminated
Default String Encoding:	UTF-8

Figure 403: PLC parameters for the Modbus TCP communication system



PLC Parameters:

Status Refresh [s]:	10
Break [ms]:	1
Startup Delay [s]:	10
Node/Application Name:	localhost/Application
User/Password:	
Min. Cycle Time [ms]:	50
Endian Mode:	Little endian
Default String Encoding:	UTF-8

PLC Parameters:

Status Refresh [s]:	10
Port Number:	1200
Break [ms]:	1
Startup Delay [s]:	10
IP Address or Hostname:	localhost
Min. Cycle Time [ms]:	50
Endian Mode:	Little endian
Default String Encoding:	UTF-8

Figure 404: PLC parameters for the Codesys V3 and Codesys V2 communication systems

Status Refresh [s]

Defines the maximum refresh time for the state structure, in seconds. If there are no device state changes within this refresh time, the state structure will still be sent to the PLC. The maximum status refresh time is 255 seconds. The shorter the refresh time, the greater the load on data communication between the device and the PLC. This can result in the state structure being the only data transmitted.

→ Please note:
The refresh time should be at least five times greater than the PLC's cycle time.

Break

This PLC parameter defines a pause time, in milliseconds, that will follow every cyclical read operation. Setting a pause can be a good idea when attempting to reduce the load on the PLC in the case of read operations that use fast polling.

IP address or hostname

Either a set IP address or a network name can be entered for most communication systems, e.g., for CODESYS V2.

- IP addresses are used when the device uses a set IP address to communicate with the host.
- Network addresses are used when the DHCP standard protocol is used by the host to assign a dynamic IP address to the device and this dynamic IP address is then used by the host to communicate with the device.
- You can use localhost as a standardized IP address if the host and the GALILEO application are found on the same device and communicate locally.

Min. Cycle Time [ms]

This PLC parameter is used to define a minimum cycle time for the device, in milliseconds. If the device reads relevant tags faster than this minimum cycle time, a corresponding pause will be forced. Minimum cycle times can be a good idea when attempting to reduce the load on the PLC in the case of read operations that use fast polling.

Endian Mode

This property is used to define the order in which bytes will be written to the communication bus. Some PLCs use processors that transmit data in big-endian format or in little-endian format. The options provided by this PLC parameter setting enable the device to transmit data of data type WORD and DWORD with either one of these formats without a problem.

The term "big-endian" means that the most significant byte will be transmitted first.

The term "little-endian" means that the least significant byte will be transmitted first.

Example using number 8342$_{decimal}$ = 0010 0000 1001 0111

Little Endian

Byte order	
Byte 0	1001 0111
Byte 1	0010 0000

Big Endian

Byte order	
Byte 1	0010 0000
Byte 0	1001 0111

Other

Baud Rate

This field is only available if the selected protocol requires a setting here.

Select the permissible transmission rates between device and PLC.

MMI Station No.

This field is only available if the selected protocol requires a setting here.

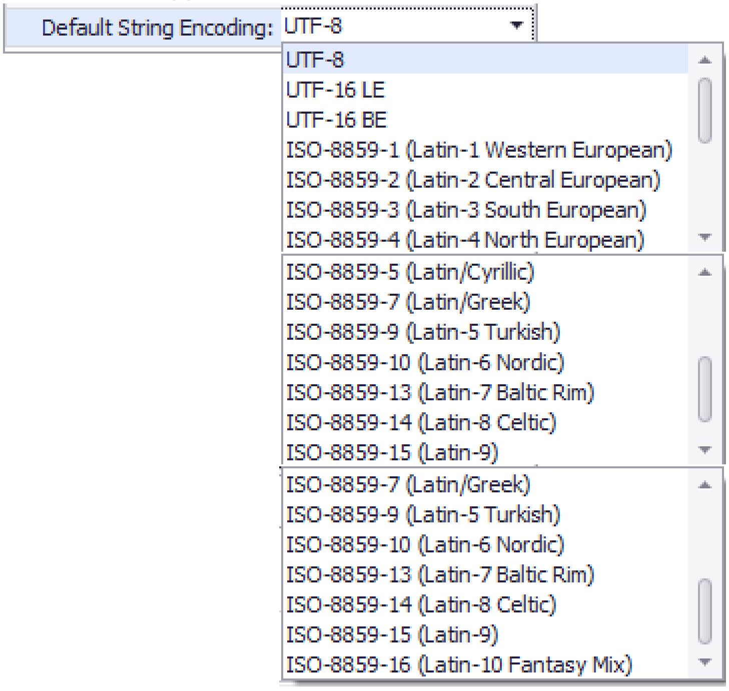
This field is used to specify the Station No. of the device in a network.

Startup Delay [s]

All communication messages are inhibited at the start of the runtime during this time. This can be necessary if the panel starts faster than the PLC and therefore many communication error messages would be generated.

Default String Encoding

Used to select a defined character set. As of this writing, the following standardized formats are supported:



Default String Encoding: UTF-8

UTF-8
UTF-16 LE
UTF-16 BE
ISO-8859-1 (Latin-1 Western European)
ISO-8859-2 (Latin-2 Central European)
ISO-8859-3 (Latin-3 South European)
ISO-8859-4 (Latin-4 North European)
ISO-8859-5 (Latin/Cyrillic)
ISO-8859-7 (Latin/Greek)
ISO-8859-9 (Latin-5 Turkish)
ISO-8859-10 (Latin-6 Nordic)
ISO-8859-13 (Latin-7 Baltic Rim)
ISO-8859-14 (Latin-8 Celtic)
ISO-8859-15 (Latin-9)
ISO-8859-7 (Latin/Greek)
ISO-8859-9 (Latin-5 Turkish)
ISO-8859-10 (Latin-6 Nordic)
ISO-8859-13 (Latin-7 Baltic Rim)
ISO-8859-14 (Latin-8 Celtic)
ISO-8859-15 (Latin-9)
ISO-8859-16 (Latin-10 Fantasy Mix)

Figure 405: Communication system PLC parameters

5.7.3 Compiler warning

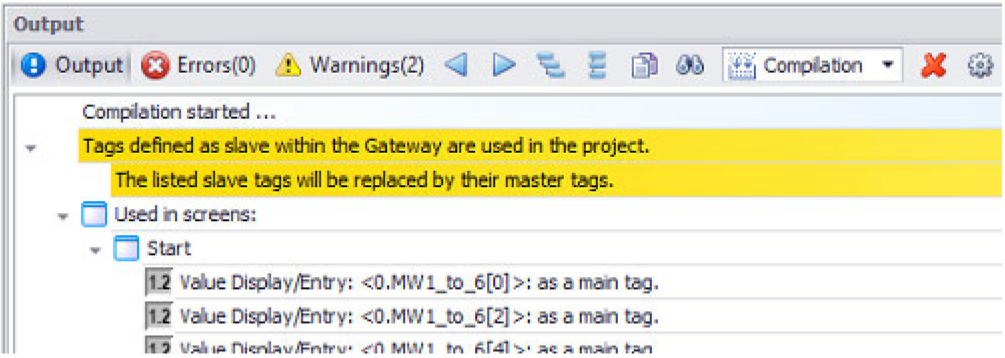
If a tag that was defined as a slave in the gateway is used in the project, a compiler warning will be shown.

This warning is used to indicate that the master tag will be used instead of the slave tag everywhere that this slave tag would have been used.

→ Slave tags cannot be used on screens, in scripts, in recipes, etc., as this does not conform to the definition of a slave tag.

Slave tags always get their value from the corresponding master tags. This means, for example, that a button writing operation to a slave tag is a design error.

The compiler warning looks as follows:



Compilation started ...
Tags defined as slave within the Gateway are used in the project.
The listed slave tags will be replaced by their master tags.
Used in screens:
Start
1.2 Value Display/Entry: <0.MW1_to_6[0]>: as a main tag.
1.2 Value Display/Entry: <0.MW1_to_6[2]>: as a main tag.
1.2 Value Display/Entry: <0.MW1_to_6[4]>: as a main tag.

Figure 406: Compiler warning example

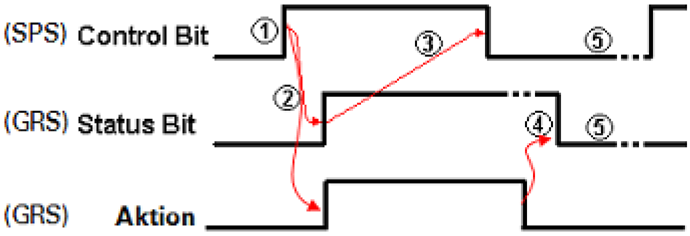
The compiler warning can be disabled in the Build and Deploy settings dialog box, under the Settings tab and Build.

5.8 Controlling functions with the PLC

The procedure for having the PLC control functions is always identical and is described below. For a detailed description of the structure of the control data block with control bits and the status data block with status bits, please refer to the → section "Control structure", page 603 and → section "Status data block", page 607 sections.

→ Control bit = Control structure control bit
Status bit = State structure status bit

1. PLC sets control bit.
2. As soon as GRS detects the control bit in the control data, the bit is copied to the corresponding status bit and the state structure is transmitted to the PLC.
3. GRS starts running the function.
4. As soon as the PLC detects the status bit in the State system tag, it must clear the control bit.
5. The status bit will only be cleared when the following two conditions are met:
   - The function's execution has been completed.
   - GRS detects that the control bit has been cleared.
6. The function will not be considered completed until the PLC detects that the status bit has been cleared.
   The same control bit is only allowed to be set again if the status bit has already been cleared.



→ The person programming the PLC must make sure that the control bit is cleared as soon as the status bit has a state of 1.

Different functions can be enabled at the same time or in succession.

Multiple PLCs can send requests with their own set control bit.

→ Control bits in the Control system tag must not be set statically, as GRS will only run functions based on status changes!
Individual exceptions are explicitly covered in this documentation.

5.9 Control structure

System Tagsare identified with the sy data type and are a required component of every GALILEO project. These tags can be found in *Project Tree\ Tags\ $: System Tags*.
System tags contain device information such as <HardwareVersion>.
This system tag makes it possible to configure the behavior of a GALILEO application differently for web clients and for the local application on the device.
Please note that system tags can be read, but cannot be written to or deleted.
The only exceptions are the sy Control, sy Status, and sy Date_Time system tags, which play an important role in data communications.

Control structure with fast polling

The Control system tag can be used to implement event-based data communication. Within this context, the PLC can trigger various functions by setting defined bits. Accordingly, these bits are also referred to as "control bits." The Control system tag will always be polled regardless of the screen that is currently being shown. The fast <Polling Rate> will always be used for it.

The Control data block contains a maximum of 8 data words (16 data bytes). The size for the Control system tag can be configured in its properties using the Size (in Word) field. If this size is smaller than 8 data words, the structure size will be reduced accordingly and only the functions for the data words used in the structure will be available.

If several PLCs are connected to the device all of them must have the same magnitude of status data block.

The control bits should not be statically set to 1, as If there are cases in which the control bit can be set statically, this will be mentioned explicitly. For a description of how the PLC can correctly trigger functions with the Control data block, please refer to the → section "Controlling functions with the PLC", page 602 section.

→ Notice:
All reserved ranges of the control data block must be set to 0!

Following is a description, for each data word, of the individual control bits and their functions when they have a state of 1.

1. Data Word

Control bit	
0…4	–
5, 6, 7	→ "Refreshingerrors", page 616 If one of these control bits is set to 1, the error tags will be updated. These control bits must only be set to 1 if control bits 5, 6, 7 of the state structure's 1st data word have a value of 0. Due to reasons of compatibility with older versions, all three bits have the same function!
8	–
9	–
10	→ "Synchronizing the time and date", page 612 The device will read the date and time from the PLC using the Date_Time tag and adjust the device's internal clock.
11	→ "Synchronizing the time and date", page 612 The device will write the date and time from the device to the PLC using the Date_ Time tag. → section "Synchronizing the time and date", page 612
12	External Screen Change The device will show the screen with the number stored in the 2nd data word.
13	–
14	–
15	→ section "Event-based data synchronization", page 590 If this control bit is set to 1, data will be synchronized. All tags will be read from the PLC.

2. Data Word

Number of target screen for an external screen change triggered by the PLC. The target screen's number needs to be entered before the screen change is triggered using control bit 12 in the 1st data word. The numbers for the screens can be found and configured using the <Screen No.> field in the screens' properties – please refer to → section "Screen types", page 263.

3. Data Word

Control bit	
0	–
1	Disable Touch If this control bit is set to 1, touch functionality will be disabled. This means that all input will be disabled accordingly. Touch functionality can be re-enabled by having the PLC set the control bit to 0. → section "Touch disable with Timeout [min]", page 264
2..5	–
6, 7	Load recipe entry If one of these control bits is set to 1, the device will read the recipe entry specified in the 6th data word with its tag values from the recipe file. The tag values are always read, even if the currently loaded recipe entry is specified in the 6th data word, see also → section "Example: Loading a recipe entry from the PLC", page 615
8	–
9..10	–
11	Read String Tags If this control bit is set to 1, the device will read all tags of data type STRING from the PLC.
12..13	–
14	Target state special function Defines the desired target state for a function triggered with control bit 15, e.g., 1 = ON 0 = OFF The Start/Stop Graph function is a function with an ON/OFF state; please refer to → section "Start/Stop Graph Block", page 732 as well.
15	Run special function If this control bit is set to 1. the special function with the function number in the 4th data word will be run. If the special function needs an argument, the argument must be saved in the 5th data word. If the special function needs a target state (ON or OFF) when called, the state of control bit 14 must correspond to this target state.

4. Data Word

Select Special Function

Number of the special function that should be triggered with control bit 15 in the 3rd data word. The number for the special function must be entered before the function is triggered with control bits 14 and 15. In addition, certain special functions require arguments, which must be entered into the 5th data word in advance as well.

5. Data Word

Configure special function

Used to provide the argument for the special functions triggered with control bit 15 in the 3rd data word.

Example:

Special function → section "Start Sync to Connection", page 626

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Communication	–								7	6	5	4	3	2	1	0

Example:
Special Function Start/Stop Graph 0…99

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Communication	Graph block ID															

6. Data Word

This data word is required in order to carry out recipe changes and read recipes. Setting control bit 7 in the 3rd data word will carry out the recipe change. Control bit 6 is used to read recipe entries. The differences are described in the → section "Example: Loading a recipe entry from the PLC", page 615 section.

The special functions in the Recipe group require the ID for the recipe object and an index for the recipe entry.

Both settings are stored in this data word as follows:

Control bit	
0...3	These bits define the ID for the target recipe object. In GALILEO, these objects can be assigned values ranging between 0 and 15.
4...15	These bits are used to define the index for the recipe entry. It is necessary to make sure that the recipe object has at least the number of recipe entries corresponding to the index. Only the first 4096 entries in a recipe object can be loaded this way.

7. and 8. Data Word

These data words are reserved for future functions.

5.10 Status data block

System Tagsare identified with the sy data type and are a required component of every GALILEO project. These tags can be found in *Project Tree\ Tags\ $: System Tags*.
System tags contain device information such as <HardwareVersion>.
This system tag makes it possible to configure the behavior of a GALILEO application differently for web clients and for the local application on the device.
Please note that system tags can be read, but cannot be written to or deleted.
The only exceptions are the sy Control, sy Status, and syDate_Time system tags, which play an important role in data communications.

The status data block is the system tag required for the handshake used for event-based data communication. It is the response to the control data block. For a more detailed description of the interaction between the status data block and the control data block, please refer to → "Controlling functions with the PLC", page 602.

→ The status structure follows the control structure.

The state structure contains 8 data words (16 data bytes). If the selected PLC supports less than 8 data words as the maximum block length, the structure size will be reduced accordingly and only the functions for the data words used in the structure will be available.

If several PLCs are connected to the device, all of them must have the same magnitude of state data block.

Event-based state structure

The status structure is transmitted to the PLC in an event-based manner and does not depend on the screen being shown. The following events will cause the status structure to be transmitted to the PLC:

- The state refresh time expiring;
  please refer to → "Status Refresh [s]", page 598 as well.
- Changes to the control structure; this frequently means changes in the device's status as well, e.g., screen changes.

The individual bits and their functions are described below.

1. Data Word

Status bit		
0	Device Connection 0 = No connection to device 1 = Connection to device is OK	This status bit always has a value of 1 and accordingly can be used to monitor the connection between the device and the PLC. To do this, use the status refresh time as a timeout on the PLC – please refer to→ section "Status Refresh [s]", page 598 as well.
1	Data synchronization 0 = The data synchronization could not be executed 1 = Data synchronization completed successfully on all connections	See also, → "Data Synchronization", page 589
2	Device Connection 0 = One or more connections between the device and PLCs are not OK 1 = All connections between the device and PLCs are OK	
3	–	
4	Error tags This status bit will have a value of 1 as long as there is still at least one error message in an alarm list because the error is still active or has not yet been confirmed. Only error tags that have the Needs confirmation property enabled will be considered.	
5, 6, 7	Update error tags This status bit will have a value of 1 as long as GRS is busy updating error tags.	Due to reasons of compatibility with older versions, all three bits have the same function! See also: Help Screen Help Text Help Picture → section "Refreshingerrors", page 616.
8, 9	–	
10, 11	Time and Date Synchronization This status bit will have a value of 1 as long as GRS is busy synchronizing the date and time. If the bit's value changes to 0, this means that the synchronization process has been completed. The internal clock will then be set to match the data in the Date_Time system tag or the current time in the internal clock will be written to the Date_Time system tag. See also: Help Screen Help Text Help Picture → section "Synchronizing the time and date", page 612	
12	Screen Change This status bit will have a value of 1 as long as GRS is carrying out the screen change. If the bit's value changes to 0, this means that the external screen change has been completed.	

13, 14	–	
15	Data synchronization This status bit will have a value of 1 as long as GRS is running the new data synchronization process. If the bit's value changes to 0, this means that the data synchronization process has been completed.	

→ Connections disabled with the "Comm. on Connection x ON/OFF" or "Connection x persistent ON/OFF" function will not be taken into account for status bits 1 and 2. If a connection is set to OFF with the "Quiet on Connection x ON/OFF" function, this connection will be considered OK for purposes of status bits 1 and 2.

2. Data Word

Status bit		
0 up to 15	Actual mask number This data word always contains the number of the current screen to be activated. The numbers of the screens concerned are shown in chapter Screen No..	

3. Data Word

Status bit		
0	–	
1	Disable Touch "Touch Disable" mode has been enabled and all touch input is accordingly locked.	
2	Printer connection 0 = Connections between device and printer not OK 1 = Printer active.	
3	Printer is not ready.	
4	User logger is active When event scripts or the user make tag changes or run certain functions, this will be logged.	
5	History is full, from now on the oldest data will be overwritten.	
6	Read recipe entry This status bit will have a value of 1 as long as GRS is busy reading the recipe entry. If the tag's value changes to 0, this means that the process of reading the recipe entry, including the transmission of data to the PLC, has been completed. This status bit will only be set if the recipe function is triggered based on control bit 6 in the Control data block's 3rd data word.	See also: Help Screen Help Text Help Picture → "Example: Loading a recipe entry from the PLC", page 615
7	Run recipe function This status bit will have a value of 1 as long as GRS is busy running the recipe function. If the tag's value changes to 0, this means that the recipe function, including the transmission of data to the PLC, has been completed. This bit will be set when a recipe entry is loaded by the user on the device (e.g., with the Load Recipe Entry function) or with control bit 7 in the control structure's 3rd data word.	See also: Help Screen Help Text Help Picture → "Example: Loading a recipe entry from the PLC", page 615

8, 9, 10	–
11	Read tags of data type STRING This status bit will have a value of 1 as long as GRS is reading all string tags from the PLC. If the tag's value changes to 0, this means that the read operation is complete.
12	This status bit will have a value of 1 as long as GRS is carrying out the screen change. If the bit's value changes to 0, this means that the external screen change has been completed.
13	–
14	State (1 = ON or 0 = Off) of function after processing (see status bit 15).
15	Execute function This status bit will have a value of 1 as long as the function with the number in the 4th data word is being executed. If the bit's value changes to 0, this means that the function has been carried out. The final state (On or Off) for the function will be registered using status bit 14. If the function needs an argument, the argument must be saved in the 5th data word.

4. Data Word

Status bit	
0 up to 15	Function Number of the Function This data word is used enter the function number of the function currently being processed (please refer to control bits 14 and 15 in the 3rd data word as well).

5. Data Word

Status bit	
0 up to 15	Function Parameter entered Used to provide the argument used for the function (please refer to the 4th data word and to status bits 14 and 15 in the 3rd data word as well).

6. Data Word

Status bit	
0 up to 15	Recipe object and recipe entry This data word is used to store the information for changing or reading a recipe (please refer to status bits 6 and 7 in the 3rd data word as well). • Control bits 0 – 3 Recipe object number • Status bits 4 – 15 New recipe entry number for the recipe object defined above

7. Data Word

Status bit	
0 up to 7	Data synchronization State of connections 0 to 7

	• 0 = The data synchronization could not be executed on this connection or a communication error has occurred • 1 = The data synchronization completed successfully on this connection and no communication error has occurred[1].	
8 up to 15	–	

1) The status bit will also have a value of 1 if the connection has been disabled with one of the following functions:
•"Comm. on Connection x ON/OFF "
•"Connection x persistent ON/OFF "
•"Quiet on Connection x ON/OFF"

8. Data Word

Status bit		
0 up to 15	–	

5.11 Synchronizing the time and date

System Tagsare identified with the sy data type and are a required component of every GALILEO project. These tags can be found in *Project Tree\ Tags\ $: System Tags*.

System tags contain device information such as <HardwareVersion>.

This system tag makes it possible to configure the behavior of a GALILEO application differently for web clients and for the local application on the device.

Please note that system tags can be read, but cannot be written to or deleted.

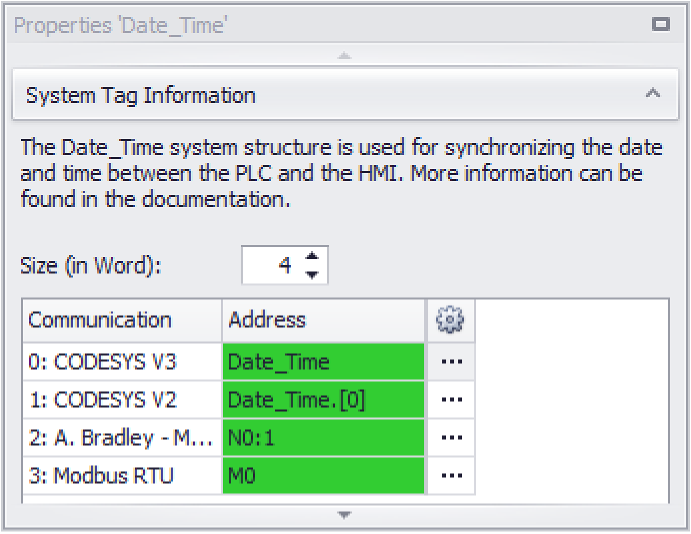
The only exceptions are the sy Control, sy Status, and sy Date_Time system tags, which play an important role in data communications.

The Date_Time system tag contains the date and time data that needs to be synchronized between the device and one or more PLCs. The tag has its own separate data block with a block size of up to 4 data words and can be read by and written to any PLC.

There are no configuration options for setting the time when this system tag will be synchronized. Moreover, it is not polled. The data is synchronized exclusively with functions or with the Control system tag.

→ In order for the Date_Time system tag to work properly, it must be assigned an address.

Project Tree\Tags\$:System Tags\Date_Time



Properties 'Date_Time'

System Tag Information

The Date_Time system structure is used for synchronizing the date and time between the PLC and the HMI. More information can be found in the documentation.

Size (in Word): 4

Communication	Address	⚙
0: CODESYS V3	Date_Time	...
1: CODESYS V2	Date_Time.[0]	...
2: A. Bradley - M...	N0:1	...
3: Modbus RTU	M0	...

Figure 407: Properties for the Date_Time system tag linked to the addresses for four communication systems

Write

- Writing with theWritePLCDateTime function
  The device will write the Date_Time system tag to the communication system once when the function is called. The number for the communication system is

passed as an argument when the function is called.

- Writing with the Control system tag
  The device will write the Date_Time system tag once to the PLC that triggers the function with the Control system tag, 1st data word, bit 11.

Read

- Reading with the ReadPLCDateTime function
  The device will read the Date_Time system tag from the communication system once when the function is called. The number for the communication system is passed as an argument when the function is called. If there are multiple PLCs, the contents of Date_Time will then be written to the other configured PLC address (es) in order to keep the date and time synchronized.
- Reading with the Control system tag
  The device will read the Date_Time system tag once from the PLC that triggers the function with the Control system tag, 1st data word, bit 10. If there are multiple PLCs, the contents of Date_Time will then be written to the other configured PLC address(es) in order to keep the date and time synchronized.

Bit 11 in the Control system tag's 1st data word is allowed to always have a value of 1. In this case, the Date_Time system tag will be transmitted to the PLC at the interval set with the status refresh time. This means that the date and time will be continuously written to the PLC. This status refresh time can be defined as a PLC parameter.

For a detailed description of how the synchronization process is supposed to run, please refer to → "Controlling functions with the PLC", page 602.

Data format of Date_Time system tag

The Date_Time system tag is used to synchronize the time and date between the PLC and the device. This system tag is transmitted in a data block with a size that is configured in the properties for the tag. The block size will be between 1 and 4 data words. Finally, the data block's format is defined by the block size, and looks as follows:

Block size of 1 data word

Bit	15 14 13 12 11	10 9 8 7 6 5	4 3 2 1 0
	Hours (0..23)	minutes (0..59)	Seconds/2 (0..29)

Block size of 2 data words

1th data word Bit	15 14 13 12 11	10 9 8 7 6 5	4 3 2 1 0
	Hours (0..23)	minutes (0..59)	Seconds/2 (0..29)
2th data word Bit	15 14 13 12 11 10 9	8 7 6 5	4 3 2 1 0
	Year (0..99)	Month (1..12)	Day (1..31)

A block size of 3 data words is not permitted

Block size of 4 data words

Byte	Most significant	Least significant
1th data word	minutes (0..59)	seconds (0..59)
2th data word	not used	Hours (0..23)
3th data word	Month (1..12)	Day (1..31)
4th data word	Year (e.g. 2006)	

Specific formats may be used for different communication protocols. Please consult the GALILEO*Information:* pane for the selected communication system in the dialog box opened with → section "Brief summary of relevant information for the communication system", page 597.

5.12 Example: Loading a recipe entry from the PLC

There are two options for loading a recipe entry with the PLC and displaying it on the device screen, with both of these options triggering an event with the control structure: Either the PLC needs to trigger the Load Recipe Entry function by setting bit 15 or one of the special control bits used by the device to read a recipe entry needs to be set.

1. Control structure: 3rd data word – set control bit 15 to 1; Load Recipe Entry function
2. Control structure: 3rd data word – set control bit 6 or 7 to 1

In both cases, the recipe object's ID and the index for the recipe entry you want must be specified in the control structure's 6th data word.

In GALILEO, the recipe object that should be modified with the PLC can be assigned an ID between 0 and 15. If the recipe entry's index is greater than the maximum number of recipe entries, an error message will appear and the operation will be canceled, i.e., no new tag values will be read. Only the first 4096 recipe entries can be loaded this way.

Differences between the two options:

Control data block 3th data word Bit 6, 7	If one of these control bits is set to 1, the device will read the recipe entry, with the corresponding tag values, from the recipe file specified in the 6th data word. The tag values are always read, even if the currently loaded recipe entry is specified in the 6th data word.
Function Load Recipe Entry:	Recipe entries can be loaded with a function from the <Recipe> group, Load Recipe Entry. The function number for Load Recipe Entry is 2050 (802Hex). This function number must be stored in the 4th data word. The function is triggered using control bit 15 in the control structure's 3rd data word. Bits 6 and 7 of the 3rd data word must be 0! The tag values of the new recipe entry are read. They are read, even if the currently loaded recipe entry is specified again in the 6th data word. See also → "3. Data Word", page 605 → "4. Data Word", page 605 → section "6. Data Word", page 606

→ The device cycle will be interrupted in order to load the recipe entry. It will not be resumed until all the recipe's tags have also been transmitted to the PLC as part of the data synchronization process!

For a description of the functions available for recipes, please refer to the → "Recipe functions", page 764 section.

For a description of how recipes are managed, please refer to the → "Using recipes to control processes", page 738 section.

5.13 Refreshingerrors

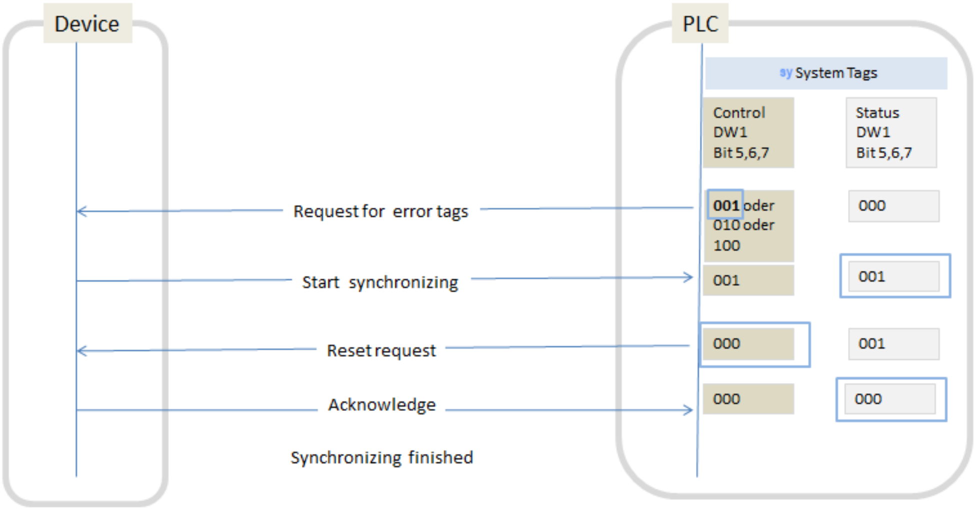
If an error occurs in the system, the PLC will set bit 5, 6, or 7 in theControl system tag's 1st data word to 1. Since the Control system tag is polled continuously, the device will detect this state change immediately and will update all error tags.
To do so, the error tags will be read from the PLC and will then be shown in the alarm list and added to the error history.

The error tags themselves should not be polled, as this will place an excessively large load on communication.

The following must be taken into account when updating error data.

- The Error tags are only read from the PLC if one of these bits is set to 1.
- The update bits in the Control system tag should only be set to 1 if the update bits in the State system tag have a value of 0!
- Due to reasons of compatibility with older GALILEO versions, bits 5, 6, and 7 in the Control system tag and in the State system tag have the same function.

For a description of the correct procedure, please refer to the → "Controlling functions with the PLC", page 602 section.



Error tags with polling

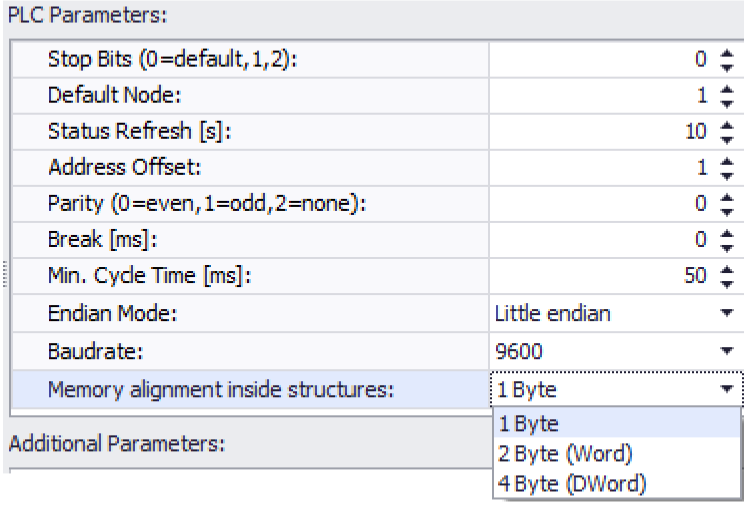
As mentioned above, you should avoid using error tag polling in order not to overload communications. Instead, you should work exclusively with the Control/Status system tags when it comes to error refreshing.

If, however, you need to update error tags using the polling mechanism, make sure to choose an appropriate polling rate that will ensure that the load on communications is not excessive.

→ If there is a large number of error tags that need to be polled as quickly as possible, it is a good idea to define these error tags on a second communication channel.
However, please note that this is only possible with Ethernet-based communications.

5.14 Data structure alignment

The data structure alignment defines where the individual tags in a structure are located in a PLC's memory. This makes it possible to achieve the maximum possible flexibility for the various PLC communication systems. Please note that not all PLC communication systems feature this option. To select a communication system, click on *Home\PLC Communication*. The PLC Parameters pane will show the various available parameters for the selected communication system. If the data structure alignment is one of them (it will be shown as "Memory alignment inside structures"), you can configure it there.



PLC Parameters:

Stop Bits (0=default,1,2):	0
Default Node:	1
Status Refresh [s]:	10
Address Offset:	1
Parity (0=even,1=odd,2=none):	0
Break [ms]:	0
Min. Cycle Time [ms]:	50
Endian Mode:	Little endian
Baudrate:	9600
Memory alignment inside structures:	1 Byte

Additional Parameters:

Figure 408: Data structure alignment using Modbus RTU as an example

The memory alignment is illustrated using the following example of a Moeller PLC and two Siemens PLCs.

A structure with the following elements was made for the example:

Structure:	**Data types:**
st test	
x button1	x BIT
by value1	by BYTE
x button2	x BIT
w value2	w WORD
dw value3	dw DWORD

The structure's start addresses have been set to ST2:MB0.0 for Siemens S7 and 0 for Modbus RTU.

Using different data structure alignments, this yields the following PLC addresses for the individual structure elements (refer to the memory layouts underneath as well):

PLC:	Elements:	Memory alignment:		
		1 byte	**2 Bytes**	**4 Bytes**

Siemens S7	button1	ST2:MB0.0	ST2:MB0.0	ST2:MB0.0
	value1	ST2:MB1.0	ST2:MB1.0	ST2:MB1.0
	button2	ST2:MB2.0	ST2:MB2.0	ST2:MB2.0
	value2	ST2:MB3.0	ST2:MB4.0	ST2:MB4.0
	value3	ST2:MB5.0	ST2:MB6.0	ST2:MB8.0
Modbus RTU	button1	M0	M0	M0
	value1	M8	M8	M8
	button2	M16	M16	M16
	value2	M.24	M32	M32
	value3	M40	M48	M64

Example: Memory Alignment

Following are graphic representations of the memory layouts for the above Modbus RTU example with the various data structure alignments. Memory cells not being used are shown in gray:

Alignment: 1 Byte

Address	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
M0								button1
M8	value1							
M16								button2
M24	value2							
M32	value2							
M40	value3							
M40	value3							
M40	value3							
M40	value3							

Alignment: 2 Bytes

Address	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
M0								button1
M8	value1							
M16								button2
M24								
M32	value2							
M40	value2							
M48	value3							
M56	value3							
M64	value3							
M72	value3							

Alignment: 4 Bytes

Address	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
M0								button1
M8	value2							
M16								button2
M24								
M32	value2							
M40	value2							
M48								
M56								
M64	value3							
M72	value3							
M80	value3							
M88	value3							

If you are unsure of which data structure alignment applies, you can verify the PLC addresses for the structure elements in the address dialog box; please refer to → section "Assigning addresses to PLC tags", page 535

5.15 Start/stop Graph Blocks from the PLCs

There are two ways to start and stop plotting for a graph from the PLC.

1. Enable the *Start if Tag:* checkbox in Home\Graph Blocks\General tab and specify a tag, e.g., <xStartBit>. To start and stop graph plotting, the PLC must set the tag to the specified value.
2. Control/Status structure
   a. Control structure: 4th data word - set 3099$_{dec}$; Start/stop Graph Block 0…99 function.
   b. Control structure: 5th data word - <Graph Blocks ID>
   c. Control structure: 3rd data word – set control bit 14 to 1; Target state = ON
   d. Control structure: 3rd data word – set control bit 15 from 0 to 1; function is activated

→ Please note that the control bit will be set back to 0 after the function is run.

The Start/Stop Graph 0…99 function will run if the following condition is met:

Status structure: 3rd data word - state bit 15 equal to 1.

For a description of the functions available for graph blocks, please refer to the → section "Graph group functions", page 721 section.

For a description of how recipes are managed, please refer to the → section "Plotting and displaying tag values using Graph Blocks", page 663 section

5.16 Communication test - Comm test

To check that the addresses assigned to the tags defined in GALILEO are correct, you can use the Communication test tool.
This tool can be used to test all the tags used in the project in terms of read access and output all the tags that were not read successfully in a list.
This makes it easier to identify tags with wrong addresses and tags that are no longer valid.

Running the communication test

Prerequisites
There must be a GALILEO project with tags with assigned addresses for one or more PLCs.

- To run the communication test on the same computer as the Automatization Software GALILEO, select Simulator as the target system:

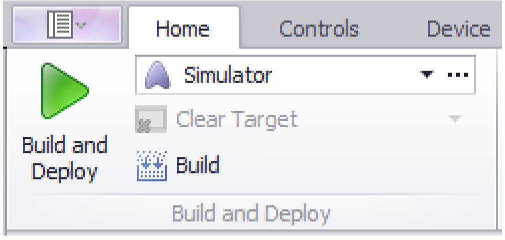


Figure 409:

- To run the communication test on a panel, select the corresponding panel with its FTP target address ("MyTestFtp" in the example):

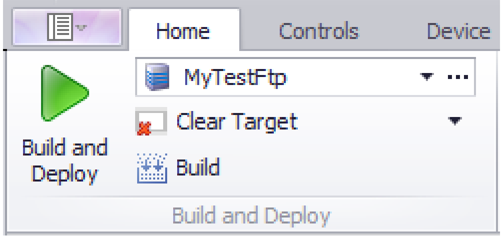


Figure 410:

- To run the test, click on the icon in the quick access toolbar in the title bar.

If the project has not yet been compiled in its current state, a prompt to this effect will appear.
Confirm the prompt.

The communication test will start.

5.16.1 Comm-Test tool

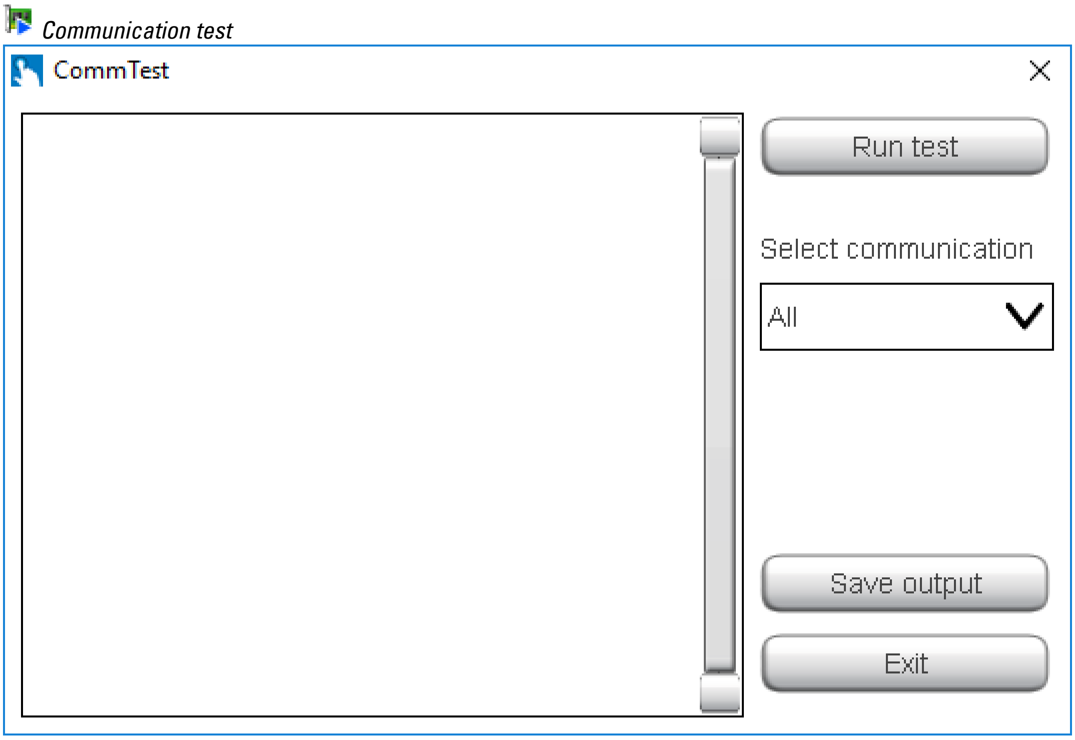


Figure 411: CommTest start screen

Select communication

1. Select the communication interface you want to test.

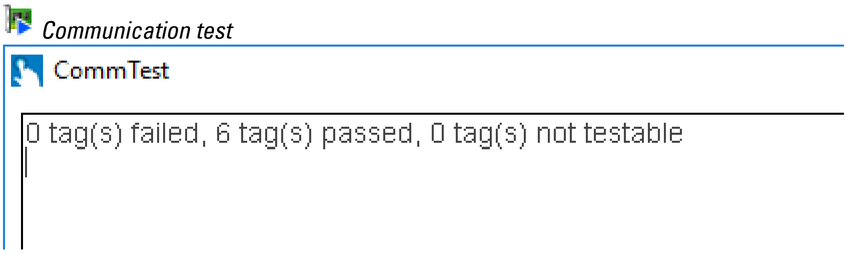
If you select All, all configured interfaces will be selected.

Run test

2. Clicking on Run test will run the test.

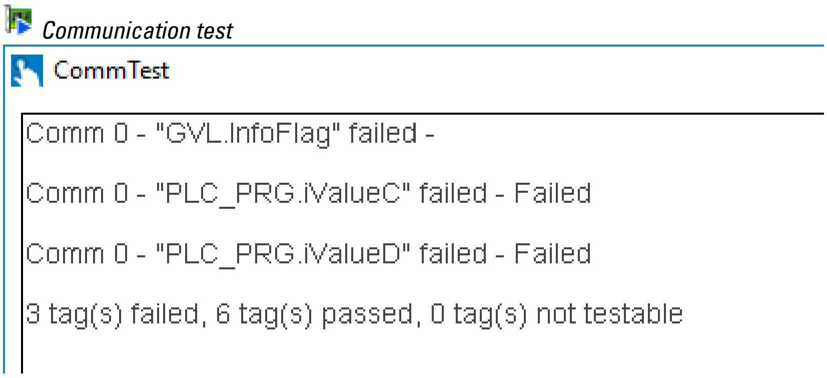
This test may take a while depending on the number of tags being tested.

The test result will be shown in the tool's left pane:



```
0 tag(s) failed, 6 tag(s) passed, 0 tag(s) not testable
```

Figure 412: Test passed



```
Comm 0 - "GVL.InfoFlag" failed -

Comm 0 - "PLC_PRG.iValueC" failed - Failed

Comm 0 - "PLC_PRG.iValueD" failed - Failed

3 tag(s) failed, 6 tag(s) passed, 0 tag(s) not testable
```

Figure 413: Test failed

You can save the result in a text file by clicking on Save output.

→ Only the tags that are actually used somewhere at runtime will be tested.
Unused tags will be ignored.

Click on Exit to finish the communication test.

5.17 Communication functions (communication)

5.17.1 Comm. on Connection x ON/OFF

This function can be used to enable and disable communication at the specified interface number. If the device is restarted, communication will be enabled for all interfaces.

Number of the Function	Interfaces 0: 2816 (B00h)
	Interfaces 1: 2817 (B01h)
	Interfaces 2: 2818 (B02h)
	Interfaces 3: 2819 (B03h)
	Interfaces 4: 2850 (B22h)
	Interfaces 5: 2851 (B23h)
	Interfaces 6: 2852 (B24h)
	Interfaces 7: 2853 (B25h)
Executable from the PLC	Yes
Function with ON/OFF state	Yes
Function available in standard screens	Yes
Function available with keyboards	No
Function available in script	Yes
Function available in application entry script	Yes

5.17.2 Connection x persistent ON/OFF

With this function the communication on the corresponding slot can be enabled or disabled. The selected state will be retained even after the device is restarted.

Number of the Function	Slot 0: 2842 (B1Ah)
	Slot 1: 2843 (B1Bh)
	Slot 2: 2844 (B1Ch)
	Slot 3: 2845 (B1Dh)
	Slot 4: 2846 (B1Eh)
	Slot 5: 2847 (B1Fh)
	Slot 6: 2848 (B20h)
	Slot 7: 2849 (B21h)
Executable from the PLC	Yes
Function with ON/OFF state	Yes
Function available in standard screens	Yes
Function available with keyboards	No
Function available in script	Yes
Function available in application entry script	Yes

5.17.3 GPI - Communic. Dialog

This function to displays the dialog Communication Setup. This function can only be used with the GPI.

Number of the Function	2820 (B04h)
Executable from the PLC	No
Function with ON/OFF state	No
Function available in standard screens	Yes
Function available with keyboards	No
Function available in script	No
Function available in application entry script	No

5.17.4 Quiet on Connection x ON/OFF

In the ON state all communication system messages are disabled on the connection concerned. In normal mode the function is in OFF state, that means all system messages are displayed and stored in the system history if necessary. The communication at the corresponding slot is always executed regardless of the state of the function.

After the device is restarted, the function is in the OFF state, i.e. communication error messages are displayed at all slots.

Number of the Function	Slot 0: 2880 (B40h)
	Slot 1: 2881 (B41h)
	Slot 2: 2882 (B42h)
	Slot 3: 2883 (B43h)
	Slot 4: 2884 (B44h)
	Slot 5: 2885 (B45h)
	Slot 6: 2886 (B46h)
	Slot 7: 2887 (B47h)
Executable from the PLC	Yes
Function with ON/OFF state	Yes
Function available in standard screens	Yes
Function available with keyboards	No
Function available in script	Yes
Function available in application entry script	Yes*)

*) If the function is used in an application entry script, it is necessary to ensure that the "startup delay" parameter is set to 0 for the corresponding communication.
If it is not, the function will remain without effect, since the following sequence will occur (if, e.g., the "startup delay" parameter is set to 5 s):

1. Galileo starts
2. Galileo determines that communication x has a start delay of 5 seconds. In other words, this communication is not enabled yet.
3. The application entry script is run. The "Quiet on Connection" remains without effect, since communication is not yet active.
4. After 5 seconds (and long after the application entry script is done), Galileo activates the communications. The previously triggered "Quiet on Connection" function remains without effect.

5.17.5 Start Sync to Connection

Slave data synchronization

If this function is triggered, the selected communication systems (as passed in the arguments) will be synchronized again. In other words, the master tags corresponding to all the tags for which the selected communication system is defined as a slave will be read and then transmitted to the slave tags.

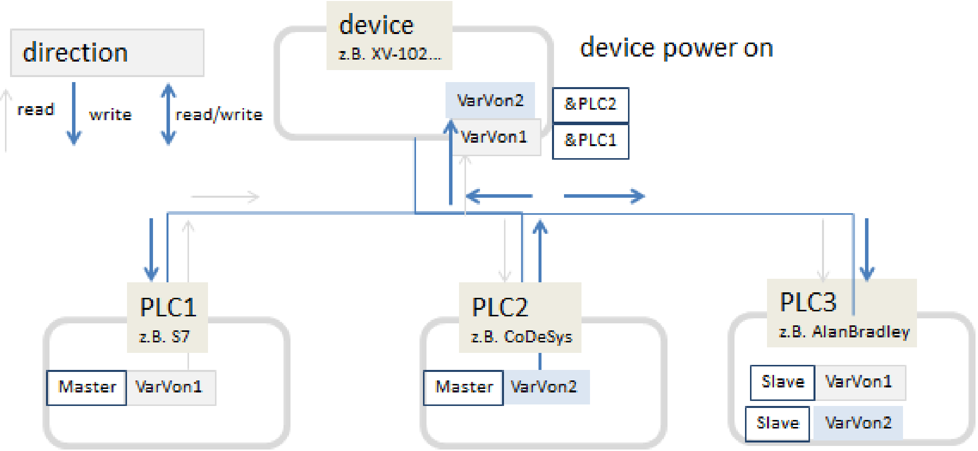
Multiple communication systems can be synchronized simultaneously.

Number of the Function	2874 (B3Ah)
Executable from the PLC	Yes
Function with ON/OFF state	No
Parameters (Control system tag: 5th data word)	Word parameter in the 5th data word in the control data block: Bit 0: Communication system 0 Bit 1: Communication system 1 ... Bit 7: Communication system 7 Bit 8...15: not used
Function available in standard screens	No
Function available with keyboards	No
Function available in script	No
Function available in application entry script	No

Overview picture example

Say communication system PLC3 is selected. This communication system is defined as a slave for tags VarVon1 and VarVon2. When the Start Sync to Connection function is called,

the device will read these two tags from the communication system defined as a master: VarVon1 will be read from PLC1 and VarVon2 from PLC2. The current tag values will then be sent to PLC3.



5.17.6 DisconnectAllClients

This function is used together with the web visualization.
When this function is run, all currently connected web clients are disconnected.

Number of the Function	6144 (1800h)
Executable from the PLC	No

Function with ON/OFF state	No
Function available in standard screens	Yes
Function available with keyboards	No
Function available in script	Yes
Function available in application entry script	Yes

5.17.7 Flush persistent tag storage values

This function can be used together with Persistent Tag Storage pseudo communication.

When this function is run, all Persistent Tag Storage communication values will be stored persistently.

Number of the Function	2892 (0b4ch)
Executable from the PLC	No
Function with ON/OFF state	No
Function available in standard screens	Yes
Function available with keyboards	No
Function available in script	Yes
Function available in application entry script	Yes

5.17.8 CAN SDO Node x ON/OFF

This function can be used when CANopen communications are being used.

When this function is run, CANopen SDO communications with the corresponding CAN node will be enabled/disabled.

Number of the Function	2888 (0b48h)
Executable from the PLC	Yes
Function with ON/OFF state	Yes
Parameter 1	CAN Node Number
Function available in standard screens	Yes
Function available with keyboards	No
Function available in script	Yes
Function available in application entry script	Yes

5.17.9 CAN SDO Node x Persistent ON/OFF

This function can be used when CANopen communications are being used.
When this function is run, CANopen SDO communications with the corresponding CAN node will be enabled/disabled.
The selected state will be retained even after the device is restarted.

Number of the Function	2889 (0b49h)
Executable from the PLC	Yes
Function with ON/OFF state	Yes
Parameter 1	CAN Node Number
Function available in standard screens	Yes

Function available with keyboards	No
Function available in script	Yes
Function available in application entry script	Yes

5.17.10 CAN PDO ON/OFF

This function can be used when CANopen communications are being used. When this function is run, CANopen PDO communication will be enabled/disabled.

Number of the Function	2890 (0b4ah)
Executable from the PLC	Yes
Function with ON/OFF state	Yes
Function available in standard screens	Yes
Function available with keyboards	No
Function available in script	Yes
Function available in application entry script	Yes

5.17.11 CAN PDO Persistent ON/OFF

This function can be used when CANopen communications are being used. When this function is run, CANopen PDO communication will be enabled/disabled. The selected state will be retained even after the device is restarted.

Number of the Function	2891 (0b4bh)
Executable from the PLC	Yes
Function with ON/OFF state	Yes
Function available in standard screens	Yes
Function available with keyboards	No
Function available in script	Yes
Function available in application entry script	Yes

5.17.12 Modbus Station No ON/OFF

This function can be used when Modbus communications are being used.

When this function is run, communications with the specified Modbus station number will be enabled/disabled.

Number of the Function	2893 (0b4dh)
Executable from the PLC	Yes
Function with ON/OFF state	Yes
Parameter 1	Galileo communication channel
Parameter 2	Modbus station no.
Function available in standard screens	Yes
Function available with keyboards	No
Function available in script	Yes
Function available in application entry script	Yes

6. Languages and texts

In the Automatization Software program, texts are used for a variety of purposes. For instance, they are used to add labels to screens and buttons, as well as to set up error messages and help information. In fact, when a visualization object is first added to a screen, it will not have a label. This means that if you want to add a label to it so that device users will later know what the object does, you can place an additional text visualization object over the original object.

If you are planning to use the same label (<TURN ON>, for example) for multiple visualization objects, you can simply create the text once and then reuse it as many times as necessary. This is tremendously useful when it comes to translating user interfaces, as the texts for labels that are used in multiple objects only need to be translated once. This also comes in handy when you need to change a label. For example, say you wanted to rename the <TURN ON> button and change it to <ON> instead. In this case, all you would have to do is change the text once, after which the program would automatically change the label for all buttons with the <TURN ON> text to <ON>.

One limitation is that a text cannot be used both for labels and error messages, i.e., it can be used for one or the other, but not both. In fact, the various texts you create will be assigned to one of the following three text categories in GALILEO:

Standard text	Texts used as labels for buttons, fields, screens, etc.
Error text	Texts shown as error messages in alarm list visualization objects
Help Text	Texts shown in help information objects on help screens in order to provide assistance

Common features for all text types

All text genres are language-dependent.

Error texts are not standard texts. However, they are subject to the same rules as standard texts:

All texts are language-dependent. They are Unicode-compatible and support characters for all languages. Moreover, the number of characters in a text has no limit. Just like standard texts, these texts will be grouped under a tab in the Text Editor, which can also be used to create new texts. To get to this tab, go to *Home\Texts\Error Texts tab*.
A different section explains how to work with texts, including error texts, in detail – please refer to → "Working with the Text Editor ", page 640.

As soon as the text is longer than the defined display area on the device screen, it will be truncated on the right.

Texts can be of any length and can have any number of lines. Moreover, they can be larger than the object in which they are being displayed. The font, font size, color, etc. used for a text are object-specific, i.e., they are configured using the properties for the specific visualization object in which the text is being displayed.

The same text can be used several times.

→ GALILEO is UNICODE compatible and supports the characters for all languages.

All text types are categorized by type in one of the lists under the following tabs: Standard Texts, Help Texts, Error Texts. These lists of texts will be available for selection for the relevant visualization objects and are used for context-sensitive text input.

The text list can be used to find existing texts, as well as to edit them if necessary.

In addition, it can be used to create texts directly and edit them.
The text list can be used to find existing texts, as well as to edit them if necessary.

Default Texts

A standard text is created as soon as you create a static text visualization object. Standard texts are also created when you use the Text as foreground property for a state in a flag display, function key, switch, slider, or button visualization object and enter the corresponding text.

Help Texts

Help texts are used together with the help button and help information visualization objects and the tag help object in order to provide device operators with assistance. For a description of how and when help texts are displayed, please refer to the → "Providing device users with help", page 932 section.

Error Texts

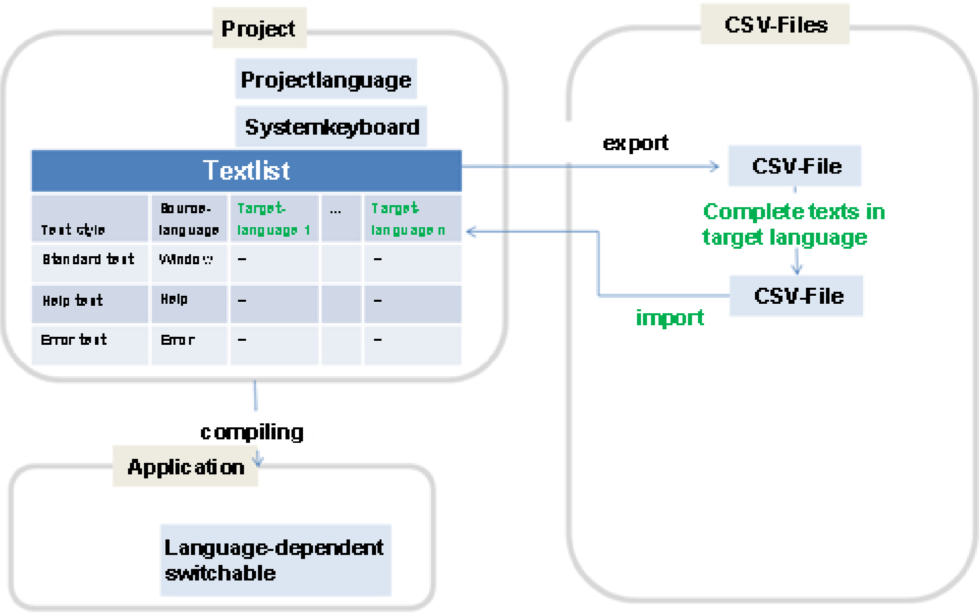
Error texts can only be linked to error tags, and are displayed inside alarm lists. These error texts can be shown on the device screen or printed out.

Error texts can be printed out using the → "Print History", page 1033, etc. functions.

All created texts are stored in the project, meaning that they are project-specific texts and need to be created again for other projects or carried over from another project with the Export/Import function.
→ section "Importing/Exporting texts", page 653

The longer the texts are, the smaller the font must be for the alarm list so that the entire error text can appear on one line.

The following pages contain an example showing how to work with texts.



Text style	Source-language	Target-language 1	...	Target-language n
Standard text	Window	–		–
Help text	Help	–		–
Error text	Error	–		–

6.1 Languages

Setting up a language selection control in the project

GALILEO projects can include up to a maximum of 100 project languages, ensuring that multi-language applications can be easily and conveniently developed. Simply click on Home\Languages to create the project languages you want. You can then create the corresponding source texts in the Text Editor by clicking on Home\Texts.

If the source texts shown in the Text Editor have been translated to all the project languages for all text types, the project will be compiled with all the project languages. The application, in turn, can be designed in such a way as to enable device operators to switch between languages with the button for a function key visualization object and/or to allow for the PLC to trigger language changes.

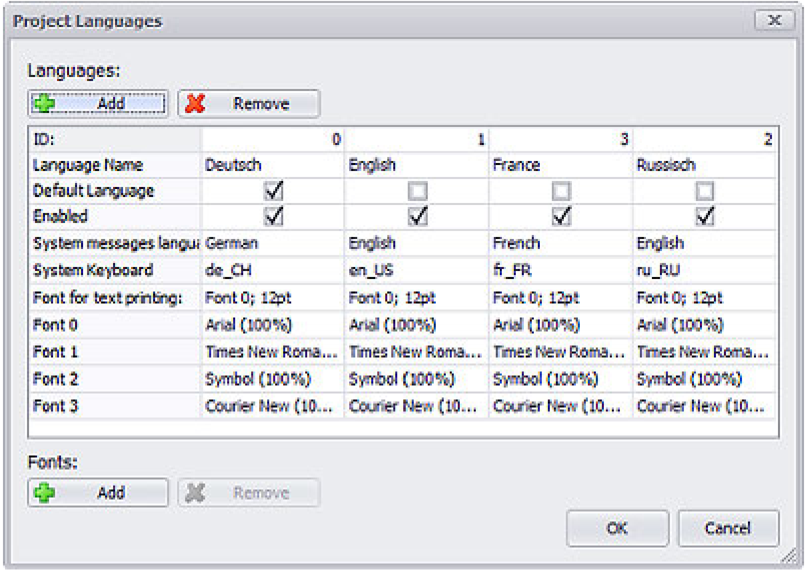
The Change to language function enables device operators to switch between languages. It is also possible for every user to save their current project language so that the application will automatically switch to it when they log in. For a description of how to enable users to set a project language, please refer to the → "Properties User management", page 894

A separate keyboard layout can be selected for each projected language. So users of different countries can work with their accustomed keyboard layout.

The following points must also be remembered with regard to language changes:

- The setting of the selected language is retained also after the device is switched off.
- When a user logs in, the application will switch to the language that was last used by them. If the user has never logged in before, the default language will be loaded. For a description of how to set the default language, please refer to → section "Set Project language", page 634.
- The Default Language is loaded if a project is recompiled and transmitted onto the unit when the unit is started.
- Default Language is loaded when a new project is transmitted to the unit and the unit is started.

Home\ Languages tab



ID:	0	1	3	2
Language Name	Deutsch	English	France	Russisch
Default Language	☑	□	□	□
Enabled	☑	☑	☑	☑
System messages langu	German	English	French	English
System Keyboard	de_CH	en_US	fr_FR	ru_RU
Font for text printing:	Font 0; 12pt	Font 0; 12pt	Font 0; 12pt	Font 0; 12pt
Font 0	Arial (100%)	Arial (100%)	Arial (100%)	Arial (100%)
Font 1	Times New Roma...	Times New Roma...	Times New Roma...	Times New Roma...
Font 2	Symbol (100%)	Symbol (100%)	Symbol (100%)	Symbol (100%)
Font 3	Courier New (10...	Courier New (10...	Courier New (10...	Courier New (10...

Figure 414: Project Languages dialog box with German as the default language and English and Chinese as translation languages.

When the application switches to a different language, all the texts for that language, as well as the font table for it, will be loaded. Al TTF fonts installed on the computer will be available for selection.

For example:
If the language is changed from <German> to<Chinese>, all the texts configured with <Courier New> will be displayed using the <MS Song> font instead.

The setting used to define the font size as a percentage (140% in the example) defines the size that will be used when displaying texts relative to the size configured for the default language. In other words, Chinese fonts would have a size that would be 140% of the default language's size in the example above. This makes East Asian scripts in particular easier to read.

→ To use texts that are exactly the same for all languages, use the exact same font for all the languages, e.g., <Arial>.
In addition, make sure that the text itself is identical for all the languages.

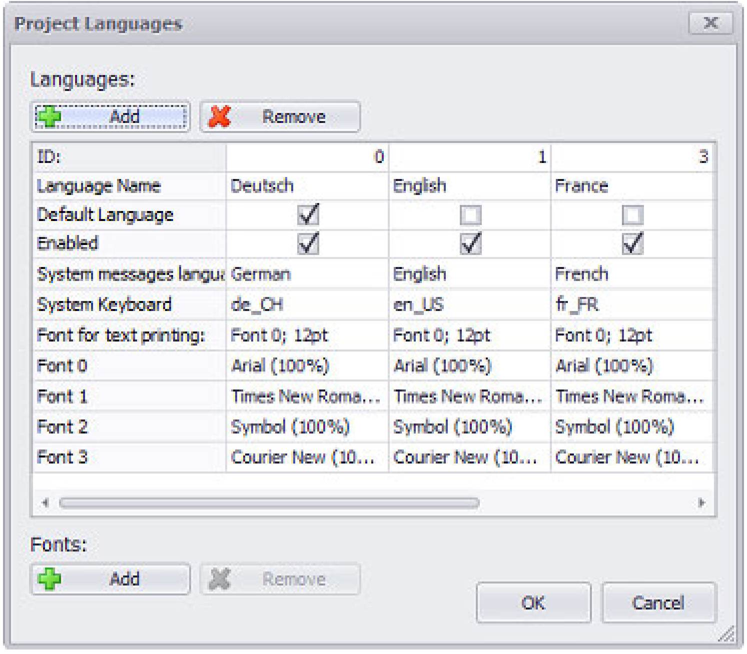
The following steps show how to create an application with multiple languages and set one of them (English, for example) as the default language.

Follow the steps below to create a function key that enables operators to switch between languages

1. **Set Project language**
   - ▶ Click on *Home\Languages* to open the Project Languages dialog box.
   - ▶ Click on the Languages: Add button and add two new columns.

- In the Language Name field, enter the names for the languages, e.g., English and French.
- Select a system keyboard layout for each language, e.g., en_US for English.
  Let's assume that the application will also have texts that remain the same regardless of the language selected (e.g., proper nouns). Accordingly, add another font (e.g., Brand).
- To do this, click on Fonts Add. Then select the same font for all the languages, e.g., <Arial>.
- Check to make sure that the checkbox in the Default Language row is enabled under the German column.
- Click on OK.

Home\ Languages tab



ID:	0	1	3
Language Name	Deutsch	English	France
Default Language	☑	☐	☐
Enabled	☑	☑	☑
System messages langu	German	English	French
System Keyboard	de_CH	en_US	fr_FR
Font for text printing:	Font 0; 12pt	Font 0; 12pt	Font 0; 12pt
Font 0	Arial (100%)	Arial (100%)	Arial (100%)
Font 1	Times New Roma...	Times New Roma...	Times New Roma...
Font 2	Symbol (100%)	Symbol (100%)	Symbol (100%)
Font 3	Courier New (10...	Courier New (10...	Courier New (10...

Figure 415: Project Languages dialog box with German as the default language and English and French as additional languages

Clicking on Add will create space in the text list for all the texts already defined in the project.
The block for the language you just created will be added under its own ID to the right of the language that you added last.

The text field properties will be filled out with the basic settings and the fonts that have already been created.

ID Consecutive ascending numbering used for identification purposes

Language Name

To change the name for a language, simply click inside the corresponding field, such as Language 0.

Default Language

If there is more than one project language, you can use the checkbox ☑ to define which one will be the default language.

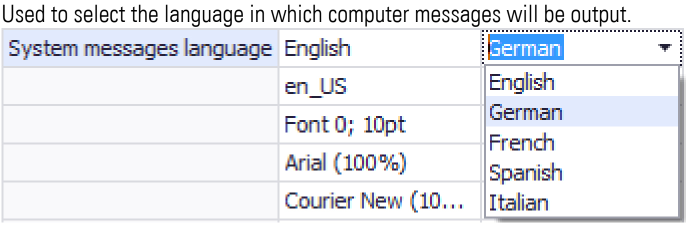
New texts you create will be automatically assigned to this project language

Enabled

Enabling this checkbox ☑ makes the corresponding language be available in the project

System messages language

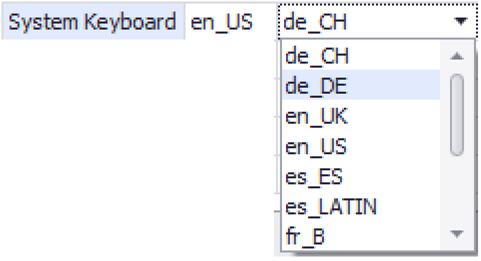
Used to select the language in which computer messages will be output.



System messages language	English	German
	en_US	English
	Font 0; 10pt	German
	Arial (100%)	French Spanish
	Courier New (10...	Italian

System Keyboard

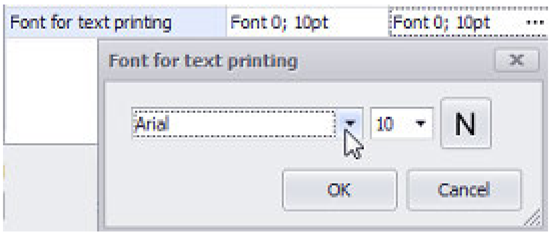
Used to specify the keyboard used for input purposes in the application



System Keyboard en_US de_CH

Font for text printing:

Font used to print GALILEO pages.
The fonts available for selection will be the same as those available in the project.



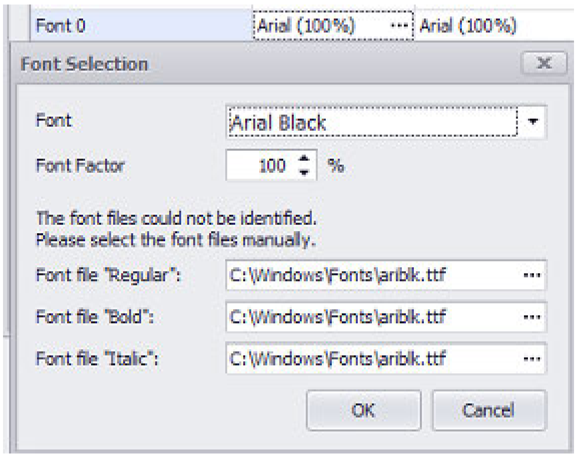
Font for text printing Font 0; 10pt Font 0; 10pt ...

Font for text printing

Arial 10

Font 0

The font used in the project.



Font 0	Arial (100%) ...	Arial (100%)

Font Selection

Font: Arial Black

Font Factor: 100 %

The font files could not be identified.
Please select the font files manually.

Font file "Regular": C:\Windows\Fonts\ariblk.ttf

Font file "Bold": C:\Windows\Fonts\ariblk.ttf

Font file "Italic": C:\Windows\Fonts\ariblk.ttf

Once you are done, you can rename and configure the project language you have created.

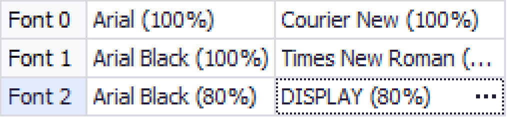
The text fields generated with the other languages will be empty.

You can copy and edit text entries by exporting and importing texts.

Fonts:

If you need additional fonts or the same font in a different font size, you can click on Add to add additional font data sets as fonts <n> to the project.

This font can be changed for each project language individually. The fonts you add will then be available for formatting in the Properties 'Text' pane.



Font 0	Arial (100%)	Courier New (100%)
Font 1	Arial Black (100%)	Times New Roman (...
Font 2	Arial Black (80%)	DISPLAY (80%) ...

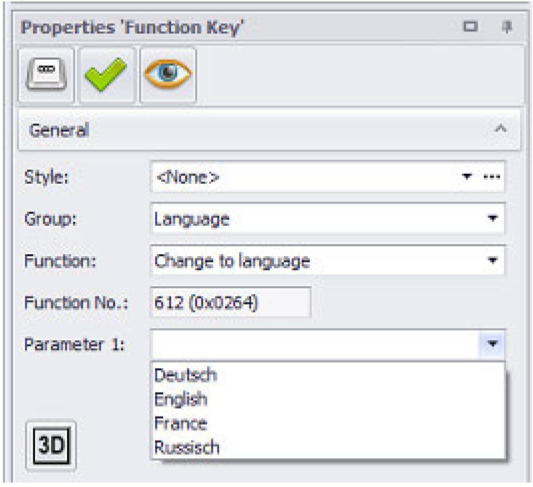
2. → "Creating texts in the Text Editor", page 643
3. **Enabling device users to switch languages with function keys**

Set Function Key

- Open screen, e.g. <Start>.
- Design the screen for your application.
- Now add a function key visualization object to it. To do this, click on Controls\Function Key.
- Position the object
  Move the cursor to the screen and draw the area for the visualization object.
- In the properties for the function key, select Language in the <Group> field and Change to language< in the Function> field.

- ▶ In the Parameter 1 field, select the project language you want, e.g., <English>.
- ▶ Double-click on the function key visualization object.
- ▶ Select the Text as foreground option for both states and enter the text you want, e.g., <English>.

Controls\ Function Key



Properties 'Function Key'

General

Style: <None>

Group: Language

Function: Change to language

Function No.: 612 (0x0264)

Parameter 1:

Deutsch
English
France
Russisch

Figure 416: Properties 'Function Key' pane with language change function selected

Paste Headline

- ▶ Click on Controls\ Text.
- ▶ Position the object
  Move the cursor to the screen and draw the area for the visualization object.
- ▶ In the properties for the Text visualization object, type the instruction you want into the Text field, e.g., <Choose your language> .
- ▶ Repeat all the steps and add additional function key visualization objects to the screen until you have a function key and Text for each language.
- ▶ Click on OK.
- ▶ Start the Galileo Simulator with function key F5 or
  by clicking on Home\ . Build.

You have now added three function keys, as well as an instruction in the form of a text visualization object, to the screen.

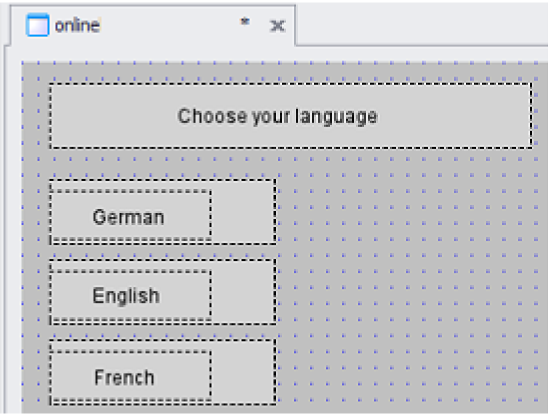


Figure 417: Section from a screen with three function key visualization objects and one text visualization object

If a device operator taps on the French button at runtime, all texts will change to the texts entered in the French column.

Example Switch Language on the Device screen

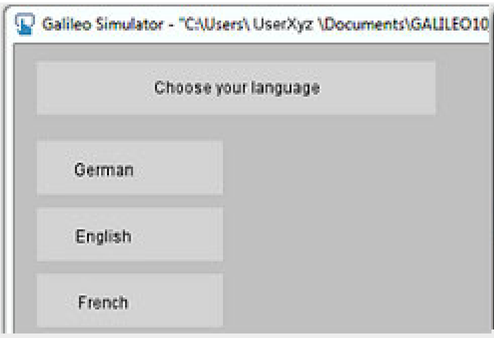


Figure 418: Device screen after selecting English

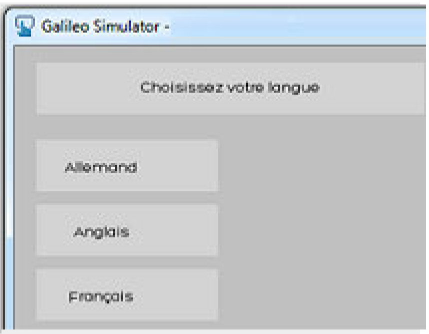


Figure 419: Device screen after selecting French

See also

- → "Languages and texts", page 630
- → "Working with the Text Editor ", page 640
- → " Texts - Text Editor", page 649
- → "Importing/Exporting texts", page 653

Functions

- → "Change to language", page 660

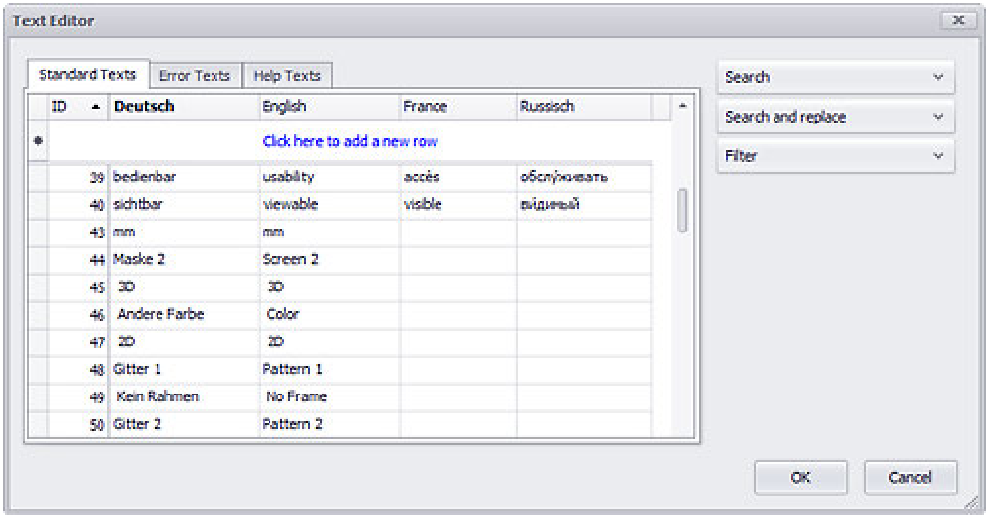
6.2 Working with the Text Editor

The Text Editor dialog box is also referred to as the "text list" in this document.

The Text Editor dialog box provides an overview of all the texts used in a project in the various project languages in order to help keep these texts manageable.

The text list can be used to find existing texts, as well as to edit them if necessary.

The tabs at the top of the dialog box can be used to switch between the text types. In addition, the various languages will be shown next to each other in columns. In order to configure project languages, click on *Home\Languages*.



ID	Deutsch	English	France	Russisch
		Click here to add a new row		
39	bedienbar	usability	accès	обслуживать
40	sichtbar	viewable	visible	відимый
43	mm	mm		
44	Maske 2	Screen 2		
45	3D	3D		
46	Andere Farbe	Color		
47	2D	2D		
48	Gitter 1	Pattern 1		
49	Kein Rahmen	No Frame		
50	Gitter 2	Pattern 2		

Context-sensitive autocomplete function for texts

As soon as you start entering text after clicking on the Click here to add a new row line, the context-sensitive autocomplete function will become active.

When entering texts, already existing texts will be proposed in a shortlist underneath the editing field. The selection list is displayed from the second character entered.

If the text you want is found in the selection list, simply select it by clicking on it.

If the text is not found in the list, type it in and finish by pressing the Enter key. This will generate a new text with a new ID.

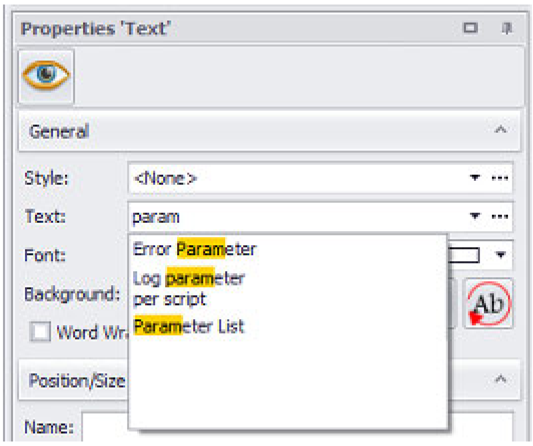


Figure 420: Excerpt from a text list with the autocomplete drop-down list expanded

Generating texts with dynamic contents

In addition to texts with fixed, static content, you can create texts with dynamic information in the form of tag values.
You can use any value tag types for this purpose, i.e.,
Bit, Error, Byte, Word, DWord, Float, and String.

In addition to the text visualization object itself, this functionality can also be used with visualization objects that work with texts:

- Button
- Function Key
- Help Button
- Screen Change
- Menu button

as well as

- Flag Display
- Switch
- Slider

There are two different dynamic string modes:

- **Simple mode**
  You can select tag values so that they will be displayed instead of a project text.
- **Advanced mode**
  You can have a project text contain placeholders located at different points for different languages
  or have physical units be converted as necessary.

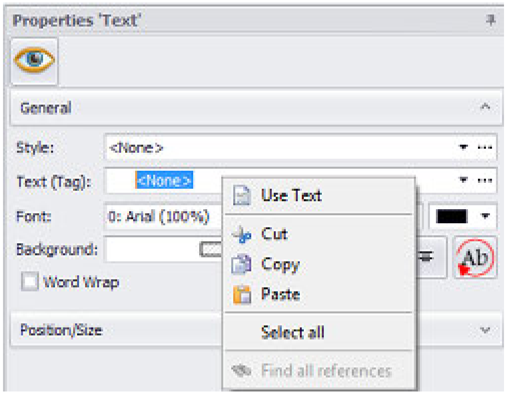
→ Make sure not to use more than five placeholders per text, as doing so will have an impact on device performance when displays are refreshed.

Simple mode

In addition to project texts, you can also use a tag directly.

You can switch from Use Text to Use Tag directly in the Properties pane and in the extended properties/states for the relevant visualization object.

To do this, right-click in order to open the context menu.

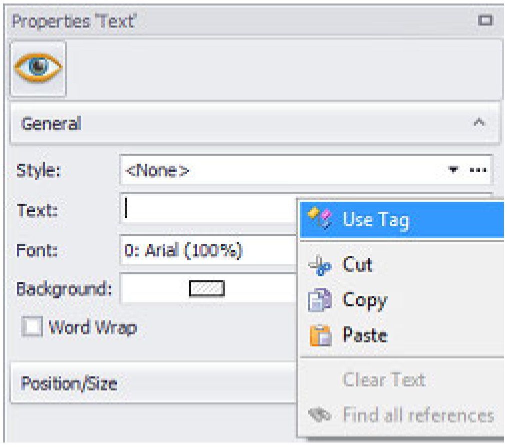


Style: <None>

Text (Tag): <None>

Font: 0: Arial (100%)

Figure 421:



Style: <None>

Text:

Font: 0: Arial (100%)

When you have selected parameter mode, there will be an additional settings called "Tag parameter."

Any value tag can be used for this purpose:

- Numeric tags
- String tags,
- Error tags
- Array elements of these tag types.
  In the case of error tags, the bit value (0/1) will be used for the display.

In simple mode, the original tag value will always be displayed, i.e., there are no conversion options.

In user controls, a "text" can be of one of four types:

- Project text: A fixed text
- Project tag: A fixed project tag
- Text parameter: User control parameter of type "text"
- Tag parameter: User control parameter of type "tag"

The normal limitations for static visualization objects apply. If a text does not fit inside the defined box, the text will either be truncated or, if the checkbox for Word Wrap is ☑, inserted into the box.

Advanced mode

When using multi-language applications, it is sometimes necessary to account for differing syntaxes.
To do this, you can use the placeholders in different order in each language as necessary.

Example
English: "There are currently <% 0> errors on machine <% 1>"
German: "Die Maschine <% 1> zeigt <% 0> Fehler an."

When using advanced mode, you can also define a conversion group and configure it.
Please note that conversion groups cannot be selected for string tags.

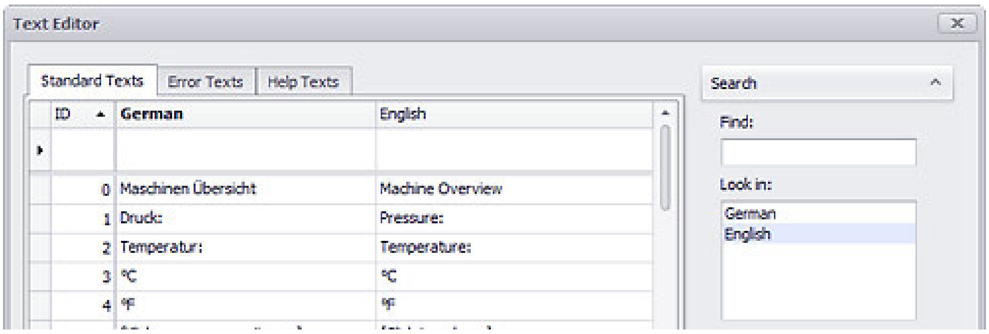
Creating texts in the Text Editor

- Open the Text Editor by clicking on *Home\Texts\Register default texts*.

There will be a column for each language under the Standard Texts, Error Texts, and Help Texts tabs.

- Go to the tab you want and click on Click here to add a new row to create a new text entry.
- Then enter the translations for the languages you are using row by row.
- Placeholders must be inserted using the following syntax: <%*Consecutive ascending No.*>
- Click on OK.

Home\Texts\Standard Texts tab



ID	German	English
0	Maschinen Übersicht	Machine Overview
1	Druck:	Pressure:
2	Temperatur:	Temperature:
3	°C	°C
4	°F	°F

Figure 422: Window Text Editor dialog box - Text list excerpt

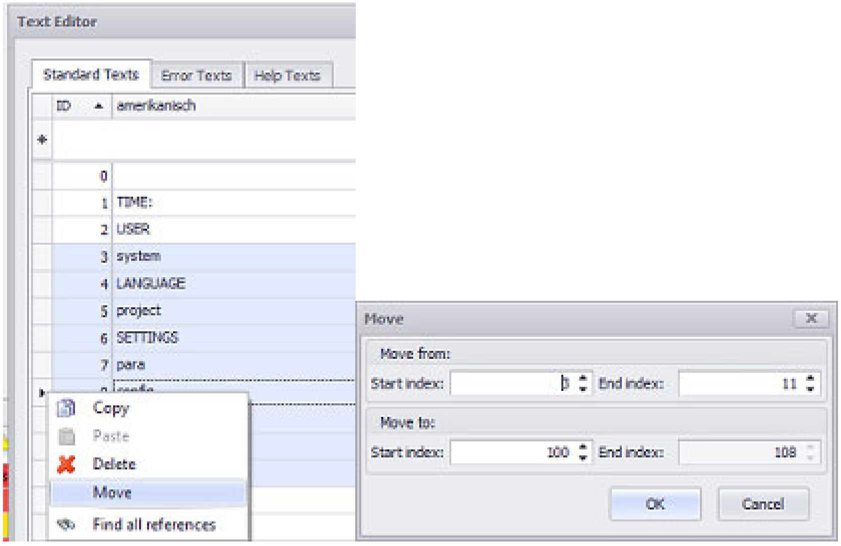
Text list context menu

- Right-click to open the context menu and select a function.

The context menu contains not only the Copy, Paste, Delete, and Find all references options, but the Move option as well.

This option can be used to move individual texts or selections of multiple texts. A dialog box will appear when you click on the option.

Home\Texts\Standard Texts tab



ID	amerikanisch
0	
1	TIME:
2	USER
3	system
4	LANGUAGE
5	project
6	SETTINGS
7	para
8	config

Move from:
Start index: 3 End index: 11

Move to:
Start index: 100 End index: 108

Figure 423: Text Editor dialog boxes - Move option in context menu

Assigning a value to placeholders

▶ In the Properties pane, go to the Text field and click on (x) to open the Dynamic string parameters dialog box.

The texts you have already created will be available in the Text box, while the table below it will list the various existing placeholders.

▶ In the Text box, select a text with placeholders.

▶ Click on ✚Add placeholder to add a placeholder.

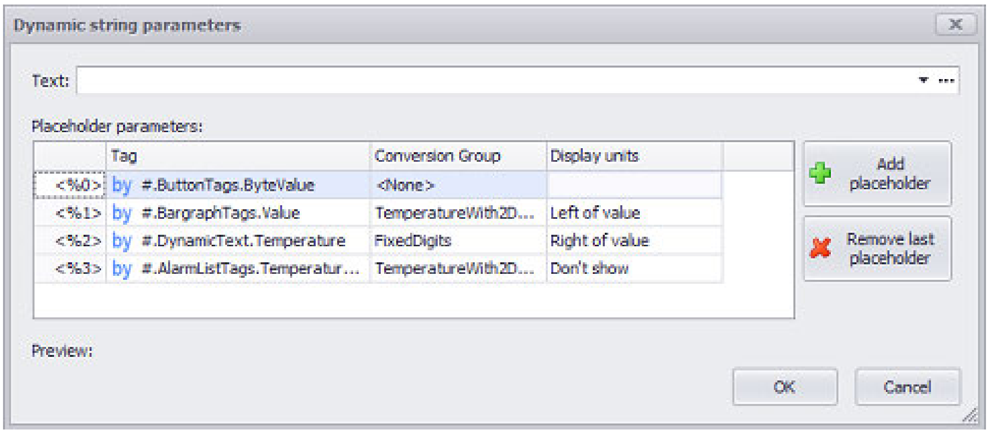
▶ Assign a tag to the placeholder.
To do so, click on the corresponding table cell.

▶ Look at the preview to check the text.

▶ Click on OK.

Clicking on ✚ Add placeholder will add a parameter set to which an existing tag and unit (conversion group) can be assigned. You can select whether the unit will be shown to the right of the value, to the left of it, or not at all.

Controls\ A Text\Properties\General\(x) tab



Dynamic string parameters

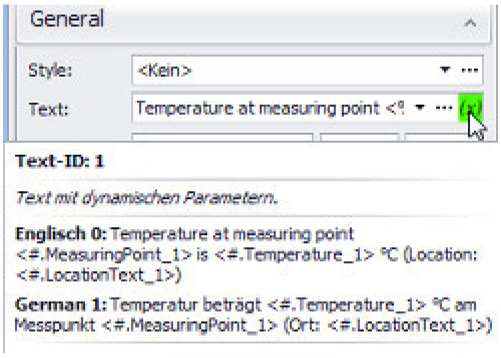
Text:

Placeholder parameters:

	Tag	Conversion Group	Display units
<%0>	by #.ButtonTags.ByteValue	<None>	
<%1>	by #.BargraphTags.Value	TemperatureWith2D...	Left of value
<%2>	by #.DynamicText.Temperature	FixedDigits	Right of value
<%3>	by #.AlarmListTags.Temperatur...	TemperatureWith2D...	Don't show

Figure 424: Dynamic string parameters dialog box after adding placeholders and making the necessary assignments

The icon color will be (x) to immediately show that the tags have been correctly assigned or (x) to indicate that an assignment is missing or invalid.



General

Style: <Kein>

Text: Temperature at measuring point <?

Text-ID: 1

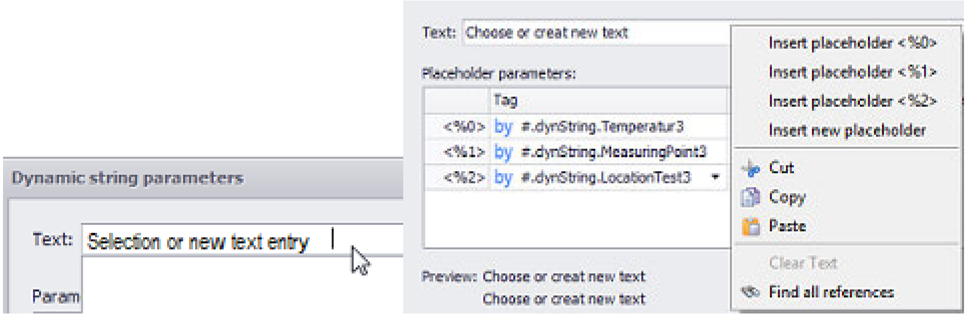
Text mit dynamischen Parametern.

Englisch 0: Temperature at measuring point <#.MeasuringPoint_1> is <#.Temperature_1> °C (Location: <#.LocationText_1>)

German 1: Temperatur beträgt <#.Temperature_1> °C am Messpunkt <#.MeasuringPoint_1> (Ort: <#.LocationText_1>)

Interface functionalities

- Click inside the Text box and create a new standard text entry directly in the project.
- Right-click inside the box to open the context menu and add placeholders.



Dynamic string parameters

Text: Selection or new text entry

Param

Text: Choose or creat new text

Placeholder parameters:

	Tag
<%0>	by #.dynString.Temperatur3
<%1>	by #.dynString.MeasuringPoint3
<%2>	by #.dynString.LocationTest3

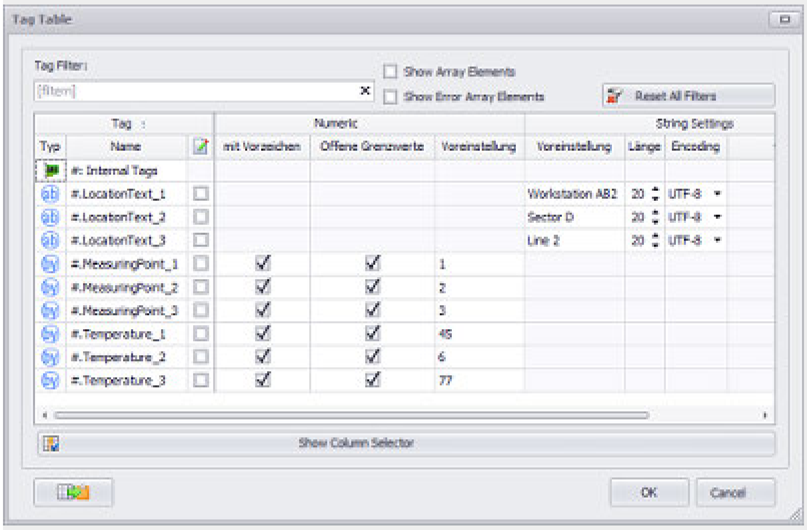
Preview: Choose or creat new text
Choose or creat new text

Figure 425: Window Text Editor dialog box - Text list excerpt

You can change the number of placeholders with the ➕ Add placeholder and ✖ Remove last placeholder buttons.

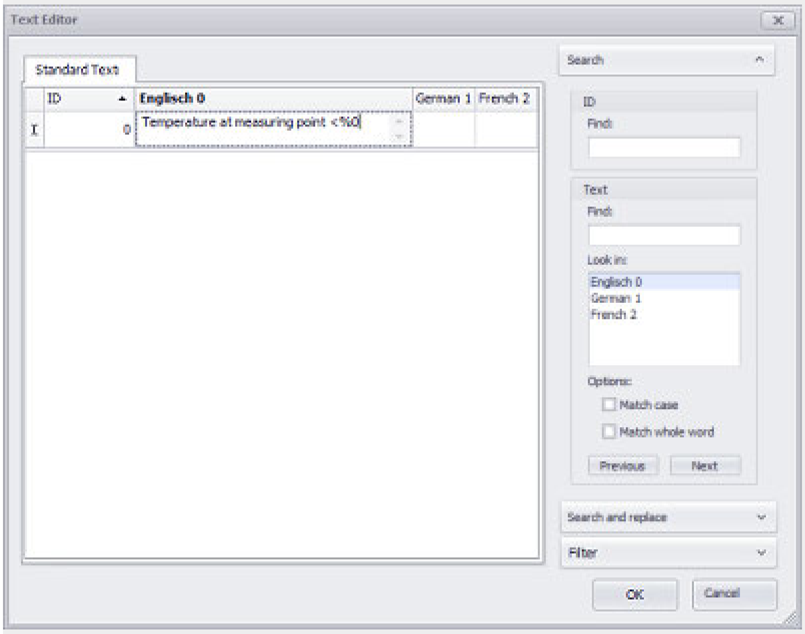
Example

- Go to Project Tree/Tags/Internal tags, create the following tags of type String and Byte, and assign the corresponding defaults.



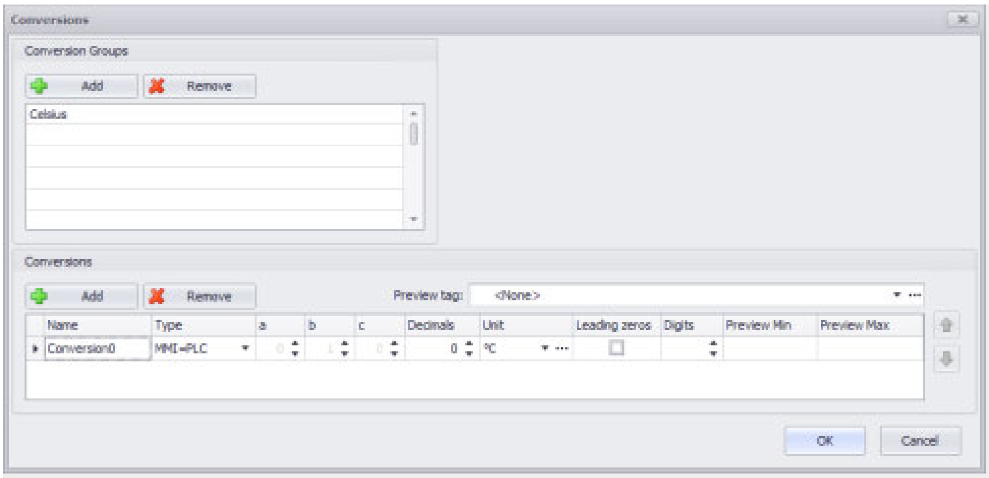
Tag			Numeric			String Settings		
Typ	Name		mit Vorzeichen	Offene Grenzwerte	Voreinstellung	Voreinstellung	Länge	Encoding
	#: Internal Tags							
ab	#.LocationText_1	□				Workstation AB2	20	UTF-8
ab	#.LocationText_2	□				Sector D	20	UTF-8
ab	#.LocationText_3	□				Line 2	20	UTF-8
by	#.MeasuringPoint_1	□	☑	☑	1			
by	#.MeasuringPoint_2	□	☑	☑	2			
by	#.MeasuringPoint_3	□	☑	☑	3			
by	#.Temperature_1	□	☑	☑	45			
by	#.Temperature_2	□	☑	☑	6			
by	#.Temperature_3	□	☑	☑	77			

- Go to *Home*\ *Texts**Standard Texts* tab and create a text saying <Temperature at measuring point <%0> is <%1> (Location: <%2>)>.



ID	Englisch 0	German 1	French 2
0	Temperature at measuring point <%0		

- Create another text for the <°C> unit.
- Create a <Celsius> conversion group for the temperature tag.
- Create a conversion and select <°C> as the unit.



Conversions

Conversion Groups

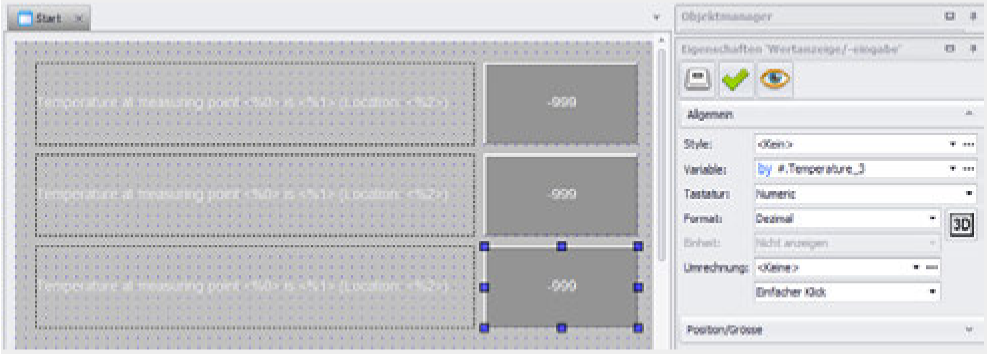
Celsius

Conversions

Preview tag: <None>

Name	Type	a	b	c	Decimals	Unit	Leading zeros	Digits	Preview Min	Preview Max
Conversion0	MMI⇒PLC	0	1	0	0	°C	□			

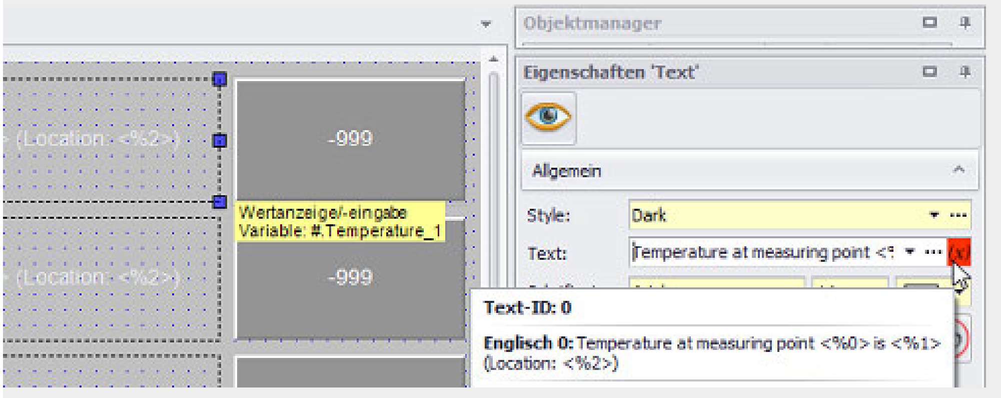
- Add the text three times on a standard screen, making sure that the texts are vertically aligned as shown in the figure below.
- Now add a value display to the right of each text.
- Assign tag <Temperatur_1> to the first value display.
- Now assign tags <Temperatur_2> and <Temperatur_3> to the other two value displays respectively.



Start
Temperature at measuring point <%0> is <%1> (Location: <%2>)
-999
Temperature at measuring point <%0> is <%1> (Location: <%2>)
-999
Temperature at measuring point <%0> is <%1> (Location: <%2>)
-999
Objektmanager
Eigenschaften 'Wertanzeige/-eingabe'
Allgemein
3D
Position/Grösse

Style: <Kein>
Variable: by #.Temperature_3
Tastatur: Numeric
Format: Dezimal
Einheit: Nicht anzeigen
Umrechnung: <Keine>
Einfacher Klick

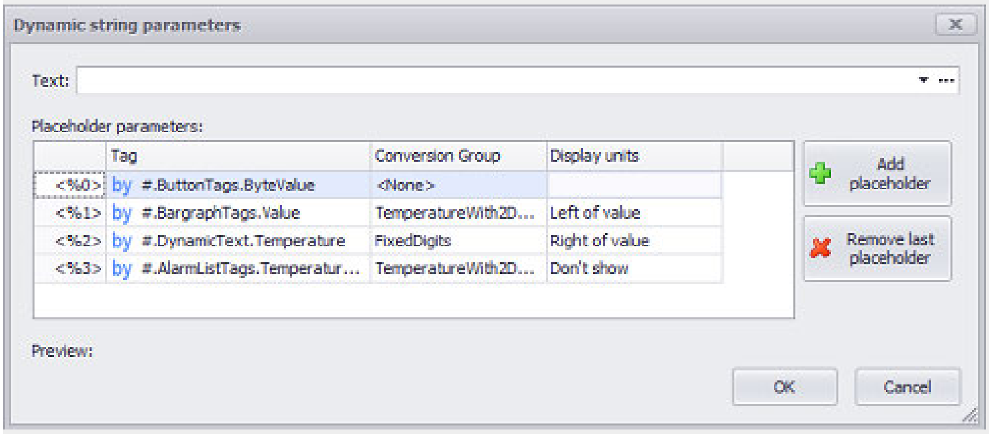
- Click on *(x)* to open the Dynamic string parameters dialog box for one of the texts on the screen.



Style: Dark
Text: Temperature at measuring point <%

Text-ID: 0
Englisch 0: Temperature at measuring point <%0> is <%1> (Location: <%2>)

- Click on ✚ Add placeholder to add the three placeholders used in the texts.
- Assign the tags to the placeholders.



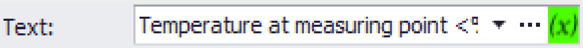
Dynamic string parameters

Text:

Placeholder parameters:

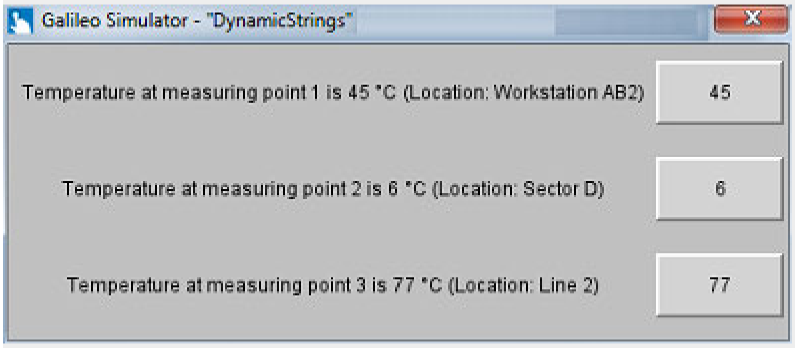
	Tag	Conversion Group	Display units
<%0>	by #.ButtonTags.ByteValue	<None>	
<%1>	by #.BargraphTags.Value	TemperatureWith2D...	Left of value
<%2>	by #.DynamicText.Temperature	FixedDigits	Right of value
<%3>	by #.AlarmListTags.Temperatur...	TemperatureWith2D...	Don't show

If the assignment is correct, the (x) turns green.



Text: Temperature at measuring point <%

When you compile the project in Galileo Simulator, the following will appear:



Galileo Simulator - "DynamicStrings"

Temperature at measuring point 1 is 45 °C (Location: Workstation AB2) | 45

Temperature at measuring point 2 is 6 °C (Location: Sector D) | 6

Temperature at measuring point 3 is 77 °C (Location: Line 2) | 77

Figure 426: Dynamic text example

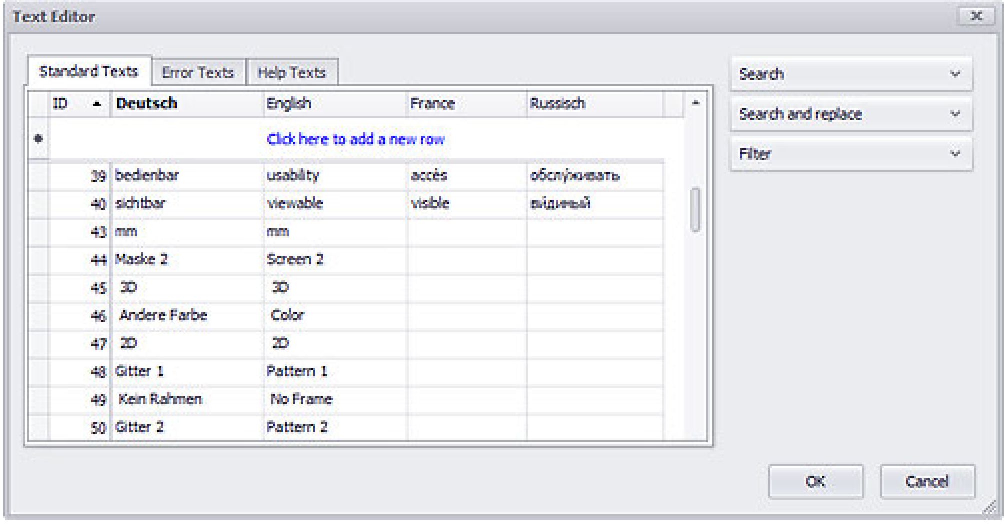
6.3 Texts - Text Editor

The Texts tag object is used to define the texts for the project.
This dialog box can be used to directly edit texts, as well as to search for and filter texts.

Since the options for all three tabs are the same, the description below applies to all text types even though it mentions the Standard Texts tab only.

All these texts can be any length and can consist of several lines. The text properties like font style, font size, colors etc. are defined in the object concerned.

Home\Texts\ Texts



Text Editor

Standard Texts | Error Texts | Help Texts

ID	Deutsch	English	France	Russisch
		Click here to add a new row		
39	bedienbar	usability	accès	обслуживать
40	sichtbar	viewable	visible	відимый
43	mm	mm		
44	Maske 2	Screen 2		
45	3D	3D		
46	Andere Farbe	Color		
47	2D	2D		
48	Gitter 1	Pattern 1		
49	Kein Rahmen	No Frame		
50	Gitter 2	Pattern 2		

Figure 427: Text Editor dialog box, Standard Texts tab

ID

Every text created in a project will be assigned an identification number (ID). This ID, in turn, will be used to link visualization objects used to enter or output a text to the corresponding text, i.e., the text itself will not be stored in the visualization object. These identification numbers are assigned automatically by GALILEO and can be manually edited in the Text Editor. In addition, any texts that are not being used in a project can be deleted using the project optimization function.
Each text type has its own set of IDs, i.e., the same ID can be used for a standard text, error text, and help text and the program will not mix them up.

→ In order to ensure that the texts you create will remain manageable, it is a good idea to subdivide them into various ID ranges.

Adding new texts

To start entering text, click on the Click here to add a new row line and then start typing. As soon as you click anywhere outside the field, the text you entered will be added to the list of texts. To add line breaks, use the Shift + Enter key combination.

You can also add the same text multiple times, as a separate ID will be assigned to each text. Please note that the context-sensitive autocomplete function will not be active when editing text in the Text Editor dialog box.

→ Entry can be aborted at any time with the ESC Button.

Editing text

Pressing the F2 key or double-clicking on a text in the Text Editor dialog box will switch to edit mode. To start editing the corresponding text, simply start typing. As soon as you click anywhere outside the field, the text you entered will be added to the list of texts. To add line breaks, use the Shift + Enter key combination. You can also add the same text multiple times, as a separate ID will be assigned to each text entry. Please note that the context-sensitive autocomplete function will not be active when editing text in the Text Editor dialog box.

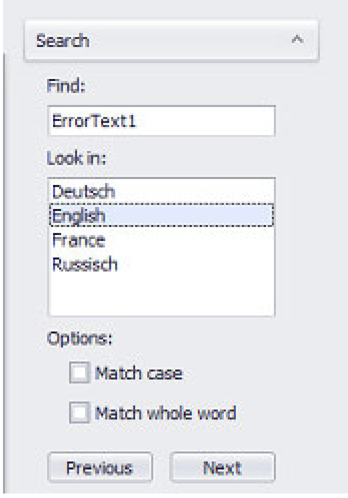
→ Entry can be aborted at any time with the ESC Button.

Search

You can also access the search function using the Ctrl+F shortcut.

If you need to search for a text in a project, you can do so under Search. Please note that searches can only be run within a specific project language and text type. As soon as you start typing in theFind: field, GALILEO will search for any matches and will immediately indicate them by highlighting the string that has been entered so far into the field. In addition, there are various options available for the search function.

Register Home\ Texts\Find



Search

Find:
ErrorText1

Look in:
Deutsch
English
France
Russisch

Options:
☐ Match case
☐ Match whole word

Figure 428: Section from *Home\Texts* dialog box with <ErrorText1> search term

Search for

This field shows the text that will be searched for in the project. If the Match whole word option is not enabled, the search for a part of the text can be started by clicking on the Previous or Next button. The search function will only search through the language selected in Look in.

Search in

The search will go through the texts for the language selected here. For a description of how to create project languages, please refer to the → section "Set Project language", page 634 section.

Match case

If this checkbox is enabled, the search will be case-sensitive.

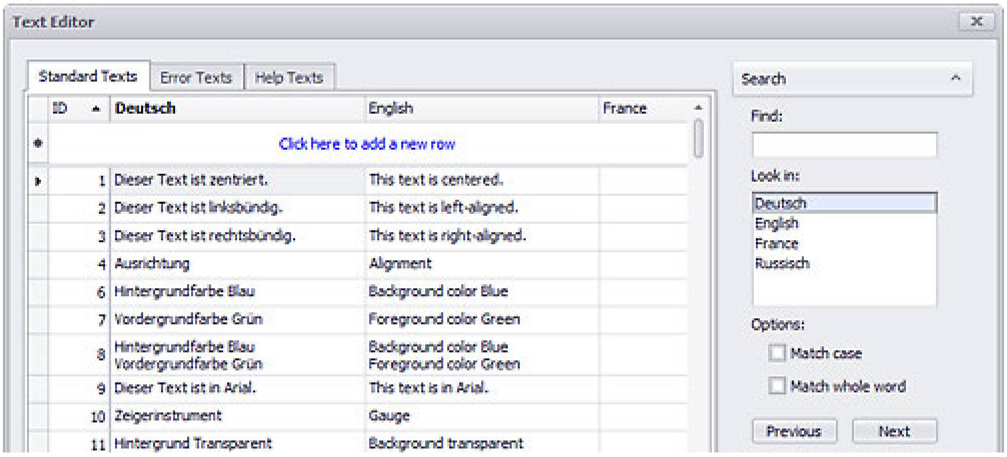
Match whole word

If this checkbox is enabled, the search will look for words that match the string entered in its entirety.

Previous / Next

The search through the list of texts will start in the search direction selected with these buttons. The search function will only search through the language selected in Look in.

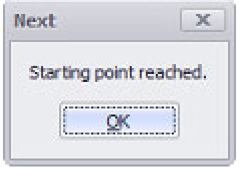
When hits are found in the list of texts, the search term entered will be highlighted in color in the corresponding hits. If no matches are found, no strings will be highlighted.



ID	Deutsch	English	France
1	Dieser Text ist zentriert.	This text is centered.	
2	Dieser Text ist linksbündig.	This text is left-aligned.	
3	Dieser Text ist rechtsbündig.	This text is right-aligned.	
4	Ausrichtung	Alignment	
6	Hintergrundfarbe Blau	Background color Blue	
7	Vordergrundfarbe Grün	Foreground color Green	
8	Hintergrundfarbe Blau Vordergrundfarbe Grün	Background color Blue Foreground color Green	
9	Dieser Text ist in Arial.	This text is in Arial.	
10	Zeigerinstrument	Gauge	
11	Hintergrund Transparent	Background transparent	

Figure 429: Text list with search for the <Conf> string in the English project language.

Once the search reaches its end, the following prompt will appear:



Next

Starting point reached.

OK

Figure 430: Message indicating the end of the search

Filter

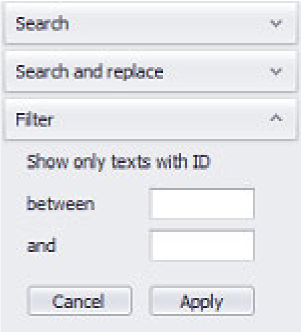


Figure 431: Section from Texts\Filter dialog box

As several hundreds or thousands of texts can be defined in the text list text lists of large projects, selecting texts that are not shown in the visible area of the list can be a tedious process. The filter can simplify this by defining a Text ID range that is to be displayed in the text list. All texts outside of this range are thus not shown in the text list.

Clicking on the Apply button will enable the filter.

Clicking on the Cancel button will disable the filter.

These filter settings are not saved with the project.

Every text genre: Default, Error and Help Texts has its own filter.

→ In order to ensure that the texts you create will remain manageable, it is a good idea to subdivide them into various ID ranges.

6.4 Importing/Exporting texts

To have your projects be available in multiple languages, check the spelling and grammar in your project, or check terminology for consistency, you can export the previously created texts and additional languages for your project so that they can be worked on or processed externally and then import them once they are ready.

The system uses the .rtf and .xlsx file formats when importing and exporting texts.

The current project folder will be used as the default folder when importing and exporting texts.
It can be changed with the options provided by Windows.

Import texts

To import text files into your project, use the text import wizard.

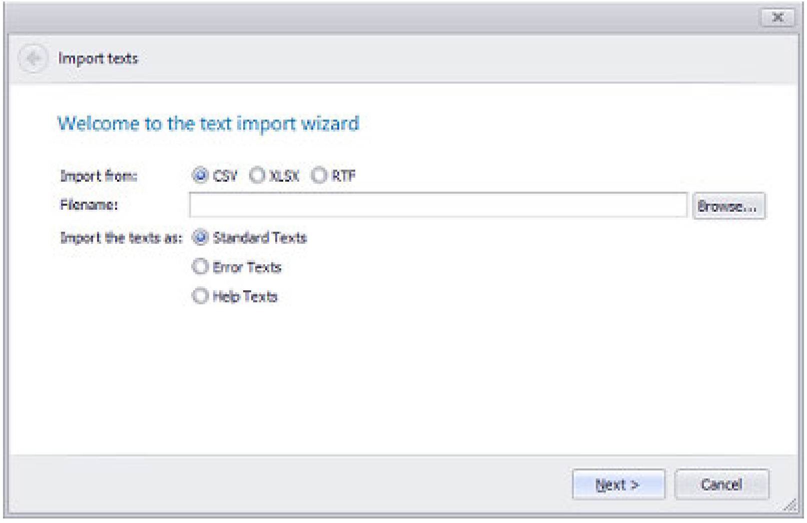
Texts in .csv ,.rtf and .xlsx file formats can be imported.

Follow the steps below to import texts:

- Click on Home\Import/Export\ Import texts.

The wizard will open in a new dialog box on your monitor.

Home\Import/Export\ Import texts



Import texts

Welcome to the text import wizard

Import from: ● CSV ○ XLSX ○ RTF

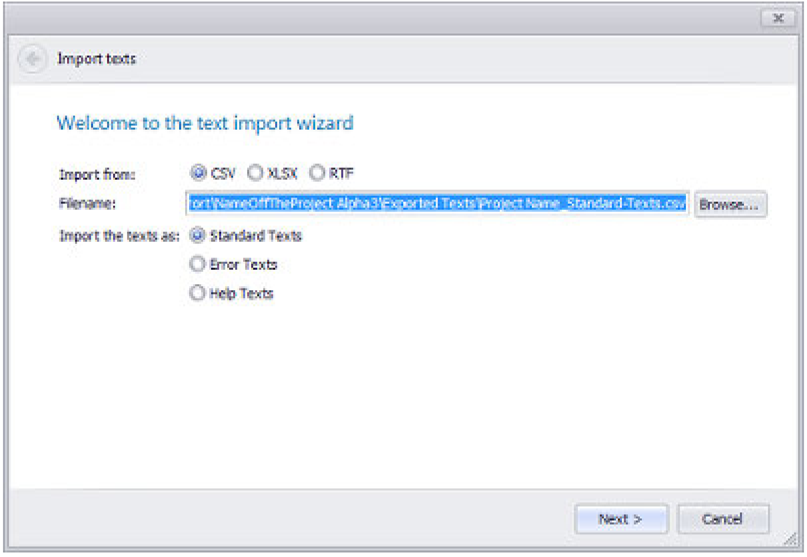
Filename:

Import the texts as: ● Standard Texts
○ Error Texts
○ Help Texts

Figure 432: Text import wizard

- Select the text format you want to import.
- Click on Browse... to select the folder where the text files are stored or directly enter the file you want together with the corresponding path.
- Select the text type that should be assigned to the texts being imported into the project: Standard Texts, Error Texts, or Help Texts.

Home\Import/Export\ Import texts



Import texts

Welcome to the text import wizard

Import from: ● CSV ○ XLSX ○ RTF

Filename: ort\NameOffTheProject Alpha3\Exported Texts\Project Name_Standard-Texts.csv

Import the texts as: ● Standard Texts ○ Error Texts ○ Help Texts

Figure 433: Import texts wizard (example using .csv)

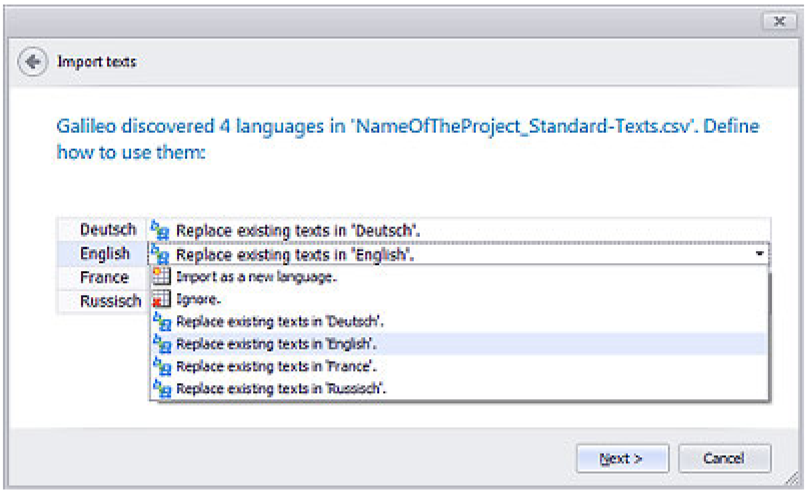
- Click on Next >.

An analysis of the selected text file will be displayed.

- Click on the relevant drop-down menu to select one of the available options. Please note that it is not possible to select multiple languages.

Import as a new language	The texts will be imported as texts for an additional language. The language name must not be in use already.
Do not use	The texts will not be imported
Replace existing texts in	Replaces the texts completely This function can be used, for example, to change the original language in which the project was created.

Home\Import/Export\ Import texts



Galileo discovered 4 languages in 'NameOfTheProject_Standard-Texts.csv'. Define how to use them:

Deutsch	Replace existing texts in 'Deutsch'.
English	Replace existing texts in 'English'.
France	
Russisch	

- Import as a new language.
- Ignore.
- Replace existing texts in 'Deutsch'.
- Replace existing texts in 'English'.
- Replace existing texts in 'France'.
- Replace existing texts in 'Russisch'.

Figure 434: Import texts wizard, step 2 – selecting an option (example using .csv)

If the option you select cannot be used, an ✖ icon will be shown next to it.

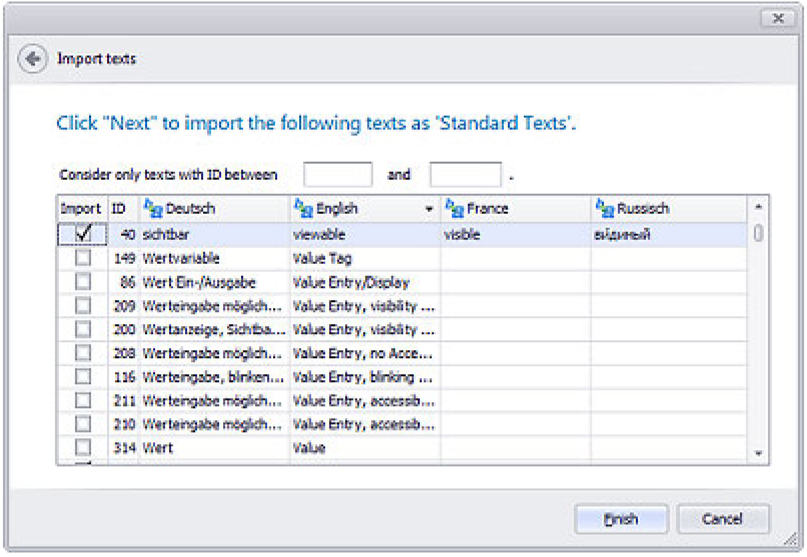
▶ Click on Next >.

A list showing the texts that will be imported will be displayed.

To prevent individual texts from being imported, disable the corresponding checkboxes (☐).

You can select the specific ranges that will be imported from the file by entering ID No. to ID No.

Home\Import/Export\ Import texts



Click "Next" to import the following texts as 'Standard Texts'.

Consider only texts with ID between ______ and ______ .

Import	ID	Deutsch	English	France	Russisch
☑	40	sichtbar	viewable	visible	відимий
☐	149	Wertvariable	Value Tag		
☐	86	Wert Ein-/Ausgabe	Value Entry/Display		
☐	209	Werteingabe möglich...	Value Entry, visibility ...		
☐	200	Wertanzeige, Sichtba...	Value Entry, visibility ...		
☐	208	Werteingabe möglich...	Value Entry, no Acce...		
☐	116	Werteingabe, blinken...	Value Entry, blinking ...		
☐	211	Werteingabe möglich...	Value Entry, accessib...		
☐	210	Werteingabe möglich...	Value Entry, accessib...		
☐	314	Wert	Value		

Figure 435: Text import wizard, step 3

▶ Click on Finish

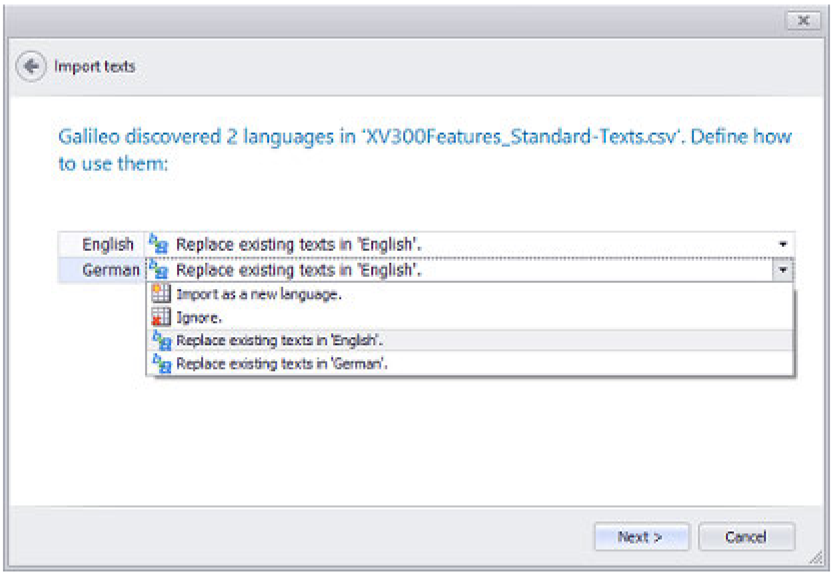
The software wizard will close

When you open the text list afterwards by clicking on *Home* *Texts,* the texts you imported will be found under the corresponding tab (e.g., in the Standard Texts tab if you chose to import the texts as standard texts).

Example showing how to swap default languages

Task: Say you have German 0 as the default language and English as a translation language and want to swap them so that you will have English 0 as the default language and German 1 as the translation language

▶ Select the corresponding options.



Import texts

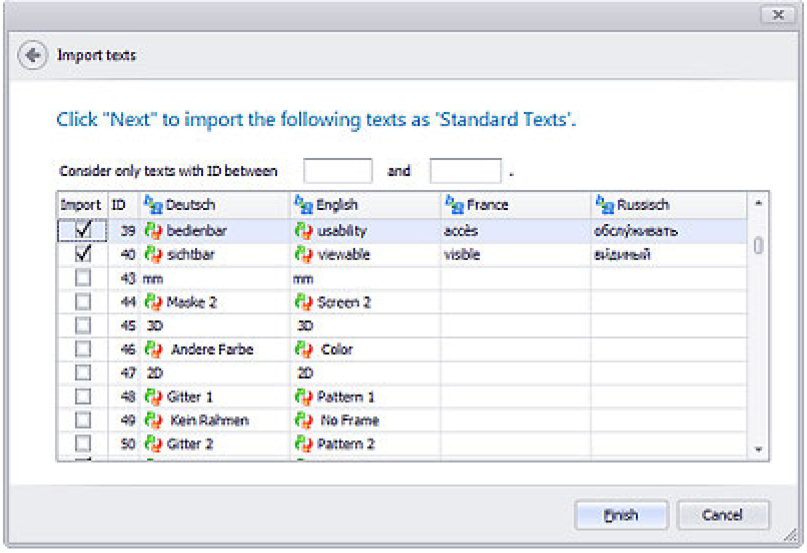
Galileo discovered 2 languages in 'XV300Features_Standard-Texts.csv'. Define how to use them:

English	Replace existing texts in 'English'.
German	Replace existing texts in 'English'.

- Import as a new language.
- Ignore.
- Replace existing texts in 'English'.
- Replace existing texts in 'German'.

▶ Click on Next >.

A list showing the texts that will be imported will be displayed.



Import texts

Click "Next" to import the following texts as 'Standard Texts'.

Consider only texts with ID between [ ] and [ ].

Import	ID	Deutsch	English	France	Russisch
☑	39	bedienbar	usability	accès	обслуживать
☑	40	sichtbar	viewable	visible	відимий
□	43	mm	mm		
□	44	Maske 2	Screen 2		
□	45	3D	3D		
□	46	Andere Farbe	Color		
□	47	2D	2D		
□	48	Gitter 1	Pattern 1		
□	49	Kein Rahmen	No Frame		
□	50	Gitter 2	Pattern 2		

Exporting texts

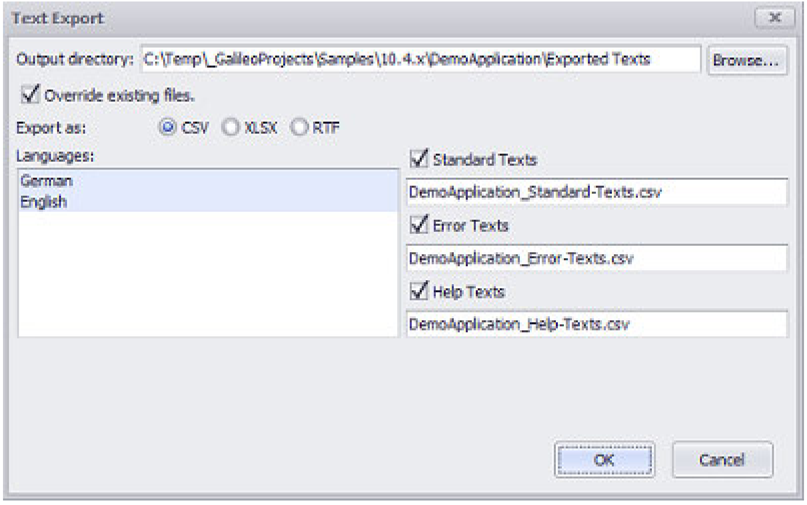
Follow the steps below to export texts:

- Click on *Home\Import/Export\* Export texts.

The wizard will open in a new dialog box on your monitor.

- Enable the checkboxes (☑) for the texts in your project that you want to export.
- Change the storage location if necessary.
- If necessary, enable the option for overwriting existing files with the same name at the storage location.
- Select the format you want.
- Click on OK.

Home\Import/Export\ Export texts



Text Export

Output directory: C:\Temp_GalileoProjects\Samples\10.4.x\DemoApplication\Exported Texts

☑ Override existing files.

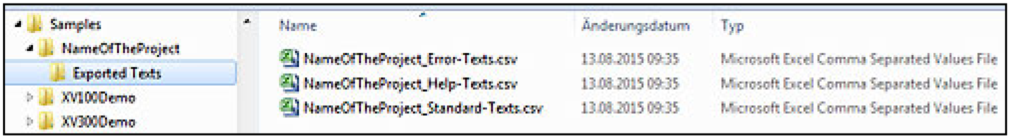
Export as: ● CSV ○ XLSX ○ RTF

Languages:
German
English

☑ Standard Texts
DemoApplication_Standard-Texts.csv
☑ Error Texts
DemoApplication_Error-Texts.csv
☑ Help Texts
DemoApplication_Help-Texts.csv

Figure 436: Exporting the texts for the <German 0> and <English 1> languages

An Exported Texts subfolder will be created in the project folder you select on your computer and the files will be stored there.



Samples
NameOfTheProject
Exported Texts
XV100Demo
XV300Demo

Name	Änderungsdatum	Typ
NameOfTheProject_Error-Texts.csv	13.08.2015 09:35	Microsoft Excel Comma Separated Values File
NameOfTheProject_Help-Texts.csv	13.08.2015 09:35	Microsoft Excel Comma Separated Values File
NameOfTheProject_Standard-Texts.csv	13.08.2015 09:35	Microsoft Excel Comma Separated Values File

Figure 437: Folder where exported texts are stored in the computer's directory structure

You can then use the individual text files as necessary, e.g., to get the texts translated.

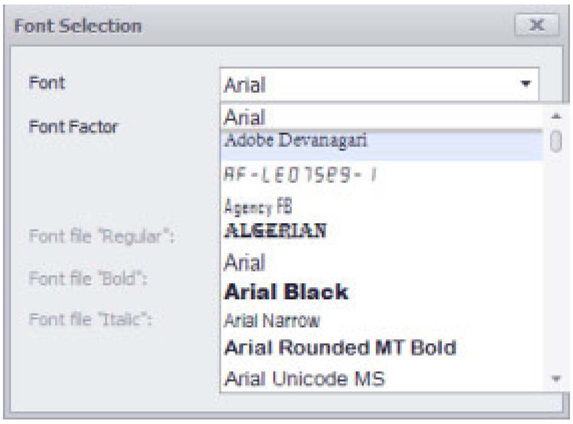
6.5 Font licenses

When creating a project in GALILEO, you can use all the fonts that are installed on the computer being used to create the project.

You can select different fonts for each language in *Home\Languages*. The fonts will be added to the GALILEO project during compilation. However, since most fonts are subject to licensing requirements, you will need to make sure that you have the necessary font licenses for each device.

→ Font licenses are normally limited to a specific number of CPUs.

Home\ Languages tab



Font Selection

Font: Arial

Figure 438: Entry for font No. ...

The following fonts are license free since they are contained in Windows CE.

Font	Windows CE 5.0	Windows embedded compact 7
Arial	x	
Comic Sans MS	x	
Courier New	x	
Gautami	x	
Georgia	x	
Gulim	x	
GulimChe	x	
Impact	x	
Kino	x	
Latha	x	
Mangal	x	
MingLiU	x	
MS Gothic	x	
MS Mincho	x	
MS Ming	x	
MS PGothic	x	
MS PMincho	x	
MS UI Gothic	x	

MSLogo	x		
NSimSun	x		
PMingLiU	x		
Raavi	x		
SC_Song	x		
Shruti	x		
SimSun	x		
Symbol	x		
Tahoma	x		
Times New Roman	x		
Trebuchet MS	x		
Tunga	x		
Verdana	x		
Webdings	x		
Wingding	x		

6.6 Text management functions

6.6.1 Change to language

Moves to the project language that is transferred as a parameter.

Number of the Function	612 (264h)
Executable from the PLC	Yes
Function with ON/OFF state	Yes
Parameter entered	Project language
Function available in standard screens	Yes
Function available with keyboards	No
Function available in script	Yes
Function available in application entry script	Yes

7. Graph Blocks

Graphsare visualization objects that can be used to display series of values as a function of time.

They show the values of one or more tags in the form of a curve or in the form of discrete measured values in various chart types. By default, these measured values are represented as a function of time.

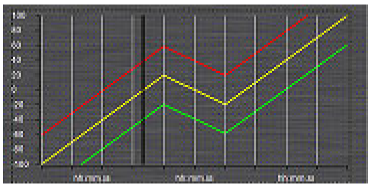


Figure 439: An example showing three values over time

Graph blocks are tag objects that take tags that can be grouped together freely and group them into a graph block list. All the tags in a graph block are sampled with a specific number of measurements, and the corresponding values are stored in RAM.

A project can have a maximum of 100 graph blocks.
In addition, each graph block list can be used to record the values for a maximum of 255 tags.
To visually represent the tags in a graph block, a graph visualization object is required. Of course, you can choose which tags in the graph block you want to display, i.e., the graph can visually represent all the tags in the block or only some of them.
However, please note that it is not possible to visually represent tags from various graph blocks in a single graph visualization object.

The Start/Stop Graph function is used to start plotting a graph. This function can be called using a function key or triggered by the PLC.

The graph visualization object will display the values of the various tags on the relevant device screen.
Plotting starts from the left edge of the object and continues towards the right, with the action that should be taken once the right edge is reached being configurable. Normally, the graph will be moved to the left by one area and plotting will continue to the right.
In addition, the recorded values in RAM can be archived.

The graph visualization object features a large number of properties – line colors, for examples – that can be used to visually configure the object in detail

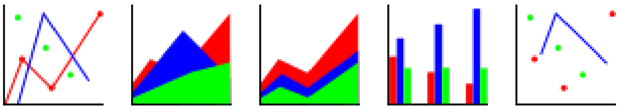


Figure 440: Available chart types

The Graph group of functions features functions that can be used to allow device users to scroll through graphs on the device screen, save graphs, and load archived values.

A marker line can be dragged to any specific point on the graph, going as far back as to when plotting started, so that the exact values for that point will be shown.

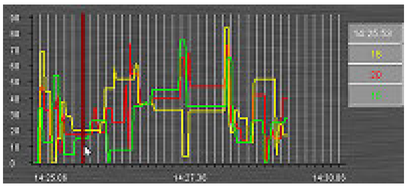


Figure 441: Example showing how to read values at a specific point in time

An overview of the functions in the Graph group is provided in a different section – please refer to → "Graph group functions", page 721.

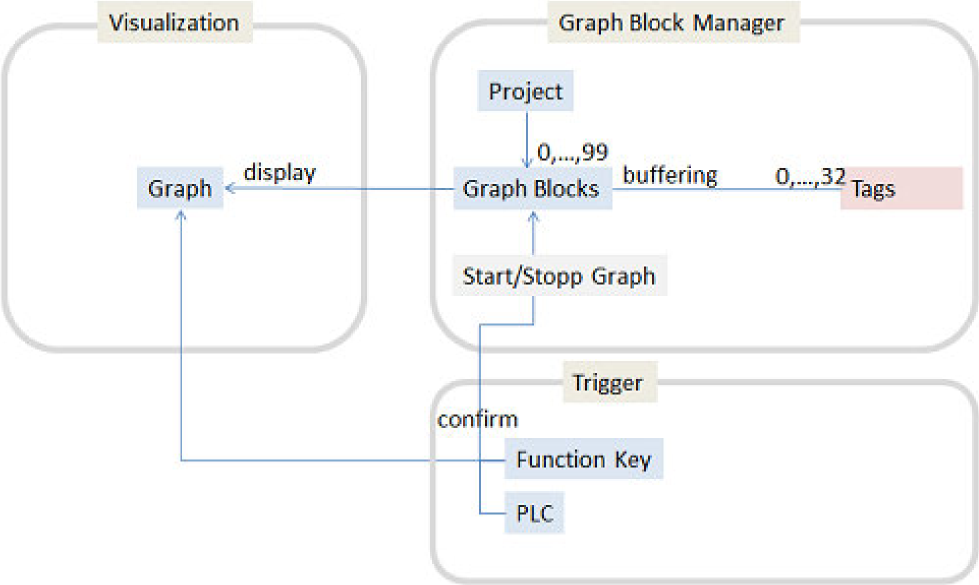


Figure 442: How graphs work: how the graph visualization object, the graph block, the Graph Block Manager, and the various tags interact with each other

7.1 Plotting and displaying tag values using Graph Blocks

Graphically display machine data

In real-life applications, a graph is used to display, record, and archive measured values.
In GALILEO, the graph visualization object is used to display these graphs, while the graph block tag object is used to plot them. For more information on how these two objects interact with each other, please refer to the → section "Graph Blocks", page 661 section.

The following steps show how to create a graph. The use case that follows describes a typical scenario, as well as the results expected on the device screen or computer.

Prerequisites

- A device must already be selected, e.g., XV-102-D6-70TWRC-10
- A communication system must already be selected, e.g., XSOFT-CODESYS-2
- A screen must have already been created, e.g., <Start screen>.
- You must have already imported the tags you want to display and edit from the PLC or created them as PLC tags.

To create a graph, follow the steps below

1. → "Creating a graph block tag object", page 672
2. → "Assign tags to the graph block", page 672
3. → "Creating a graph visualization object and linking it to a graph block", page 673
4. → "Show curves", page 667
5. → "Enabling/disabling the marker line", page 668
6. → "Start and Stop of recording", page 668Start and Stop of recording
7. → "Clearing the buffer", page 669
8. → "Loading an archive file", page 670Loading an archive file
9. → "Saving the data set in an archive file", page 671Saving the data set in an archive file
10. → "Navigating during plotting", page 673Navigating during plotting

Example showing agraph visualization object on the device screen

Prerequisites

For this exercise, you will need the following tags: wPressure, wTemperature, wFlow, wRuler1, wRuler2, and wRuler3 of data type WORD; xShowRuler of data type BIT; and sRulerTime of data type STRING. There are two ways to create the required tags and fill them with values at runtime:

- Importing them from the included PLC project.
- Creating the tags in *Project Tree\Tags* and creating a small loop script that generates three different curves. For detailed information on how to create a loop script, please refer to the → section "Using scripts to perform actions",

page 956 section.

Job

Say that you want the pressure, temperature, and flow rate to be monitored for a machine at a production plant. The current values for these tags will be obtained by the PLC, and you want to use a graph visualization object to display them. In addition, say you want for the current values in the graph visualization object to be output in a display section called Current values.

The screen will also have a Show marker line button that can be used to enable/disable a marker line. In addition, it will be possible to drag& this marker line to any point within the window. To implement this button, you will be using a button object.

When the marker line is enabled, the Values at marker line display section will show the tag values for the three curves at the point where the marker line is placed.

You will also be adding four other buttons to the graph visualization object:

Start, used to start and stop plotting.

Clear buffer, used to clear the recorded tag values and start over

Load, used to open a list of archive files for selection and show the selected archive file

Save as, used to save the recorded tag values in an archive file

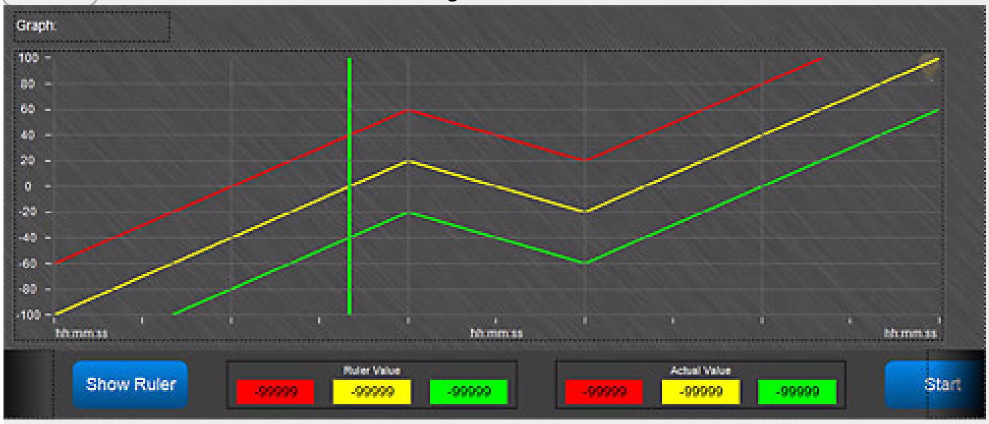


Figure 443: Device screen with graph visualization object and controls

To complete this example, follow the steps below:

Create a graph block and configure the corresponding parameters

To create a graph block, follow the steps below:

- Click on *Home\Graph Blocks*.
- Click on Add\Graph Block to create a new graph block.
- Click on the graph block and rename it <PlcData>.
- Open the Tags tab.
- Click on 🖧 to assign the tags you want to the graph block.

- Select the <wPressure>, <wTemperature>, and <wFlow> tags by enabling the corresponding checkboxes.
- Click on OK.
- Open the General tab.
- In the No. of samples field, enter a value of <1000>.
- In the Time span field, enter a value of <1>.
- Click on OK.
- Open the Archive tab.
- Select the File (DATA Directory) option from the drop-down menu in the <Storage location> field.
- Type PlcDataArchive into the <Archive name> field.
- Enter a value of 5 into the <Number of archives> field.
- Click on OK.

Create a screen with a graph visualization object

- Create a screen called GraphDisplay.MSK and set a dark gray color as its background.

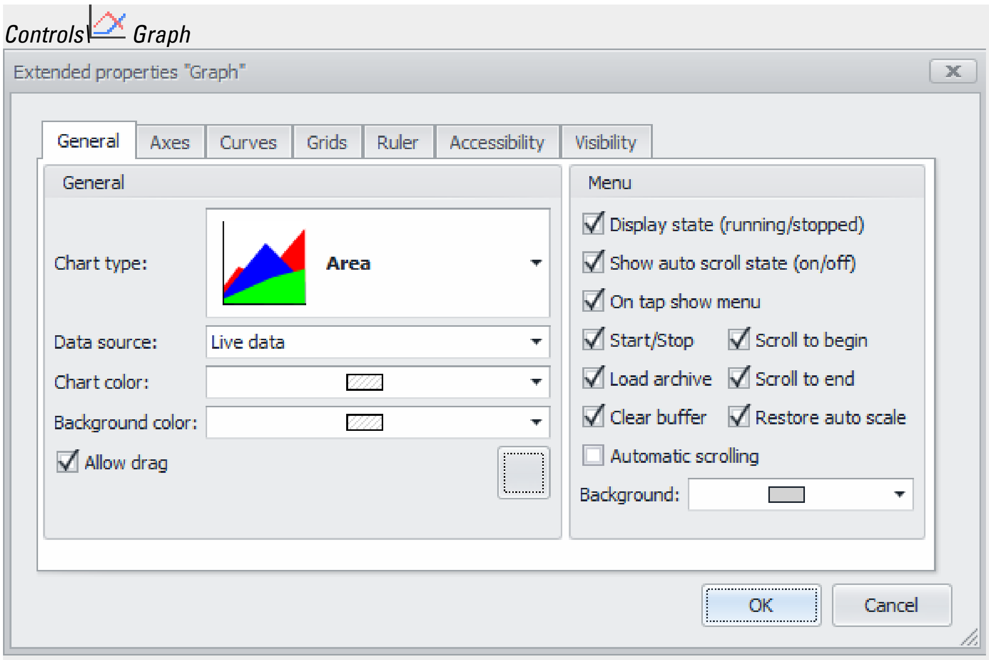
To create a graph visualization object, follow the steps below:

- Click on *Controls\Graph*.
- Position the object
  Move the cursor to the screen and draw the area for the visualization object.

A pre-formatted graph visualization object will be shown on the screen.

To edit the appearance and behavior of the graph visualization object, follow the steps below:

- Double-click on the visualization object to open the Extended properties "Graph" dialog box.
- Open the General tab.
- Select the Line chart type.
- Select <Transparent> in the Background and Chart color drop-down menus.
- Enable the Allow drag checkbox.



Controls | *Graph*

Extended properties "Graph"

General | Axes | Curves | Grids | Ruler | Accessibility | Visibility

General

Chart type: Area

Data source: Live data

Chart color: ▨

Background color: ▨

☑ Allow drag

Menu

☑ Display state (running/stopped)

☑ Show auto scroll state (on/off)

☑ On tap show menu

☑ Start/Stop ☑ Scroll to begin

☑ Load archive ☑ Scroll to end

☑ Clear buffer ☑ Restore auto scale

□ Automatic scrolling

Background:

OK | Cancel

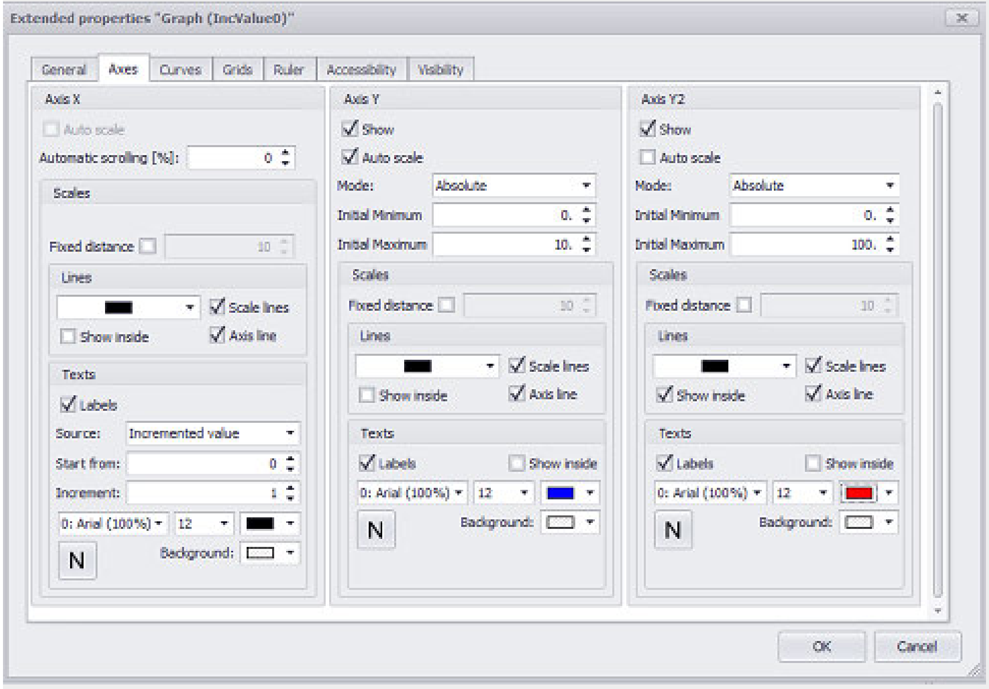
Figure 444: Extended properties "Graph"; settings in General tab

- Double-click on the visualization object to open the Extended properties "Graph" dialog box.
- Open the General tab.
- In the General section, select Transparent as the Background color and <Chart color>. Enter a value of 85 into the <Automatic scrolling> field. Select hh:mm.ss using the drop-down menu for the <Date/Time format> field. Enter a value of 300 into the <Default time interval> field.
- Enable the Allow drag checkbox.

Configure the scale and grid

- Go to the Grids tab and enable the checkbox for Grid X.
- Go to the Axes tab and, in the Texts section, enable the two Labels checkboxes.
- Select <White> as the text color for each.
- In the Axis Y section, set the Initial Minimum field to a value of <-100> and the Initial Maximum field to a value of <100>.
- In the Axis Y section, enable the checkbox for Scale lines.
- In the Grids section, enable the Grid X and Grid Y checkboxes.
- Enter a value of 85 into the <Automatic scrolling> field. Select hh:mm.ss using the drop-down menu for the <Date/Time format> field. Enter a value of 300 into the <Default time interval> field.

Controls *Graph*



Extended properties "Graph (IncValue0)"

Axis X

- □ Auto scale
- Automatic scrolling [%]: 0
- Scales
  - Fixed distance □ 10
  - Lines
    - ■ ☑ Scale lines
    - □ Show inside ☑ Axis line
  - Texts
    - ☑ Labels
    - Source: Incremented value
    - Start from: 0
    - Increment: 1
    - 0: Arial (100%) 12 ■
    - N Background: □

Axis Y

- ☑ Show
- ☑ Auto scale
- Mode: Absolute
- Initial Minimum 0.
- Initial Maximum 10.
- Scales
  - Fixed distance □ 10
  - Lines
    - ■ ☑ Scale lines
    - □ Show inside ☑ Axis line
  - Texts
    - ☑ Labels □ Show inside
    - 0: Arial (100%) 12 🟦
    - N Background: □

Axis Y2

- ☑ Show
- □ Auto scale
- Mode: Absolute
- Initial Minimum 0.
- Initial Maximum 100.
- Scales
  - Fixed distance □ 10
  - Lines
    - ■ ☑ Scale lines
    - ☑ Show inside ☑ Axis line
  - Texts
    - ☑ Labels □ Show inside
    - 0: Arial (100%) 12 🟥
    - N Background: □

Figure 445: Extended properties "Graph"; settings in Axes tab

Link the graph visualization object to the graph block tag object

Show curves

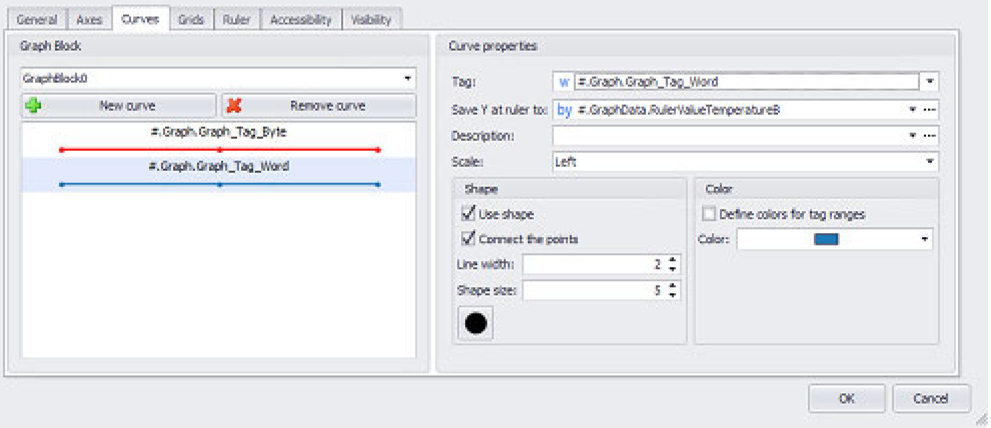
To assign the graph block tag object to the graph visualization object, follow the steps below:

- ▶ Open the Curves tab.
- ▶ Select PlcData using the <Graph Block> drop-down menu.
- ▶ Click on New curve three times.
- ▶ Click on the first curve you just created.
- ▶ In the Curve properties section, use the Tag Y drop-down menu to select the <wPressure> tag. Then use the Save Y at ruler to drop-down menu to select the <wRuler1> tag. This will make the tag value at the marker line's current location be saved to <wRuler1>. Select a red color as the Color.
- ▶ Now click on the second curve.
- ▶ In the Curve properties section, use the Tag Y drop-down menu to select the <wTemperature> tag. Then use the Save Y at ruler to drop-down menu to select the <wRuler2> tag. This will make the tag value at the marker line's current location be saved to <wRuler2>. Finally, select a yellow color as the Color.
- ▶ Now click on the third curve.
- ▶ In the Curve properties section, use the Tag Y drop-down menu to select the <wFlow> tag. Then use the Save Y at ruler to drop-down menu to select the

<wRuler3> tag. This will make the tag value at the marker line's current location be saved to <wRuler3>. Finally, select a green color as the Color.

- ▶ Click on OK.

Controls *Graph*



Graph Block

GraphBlock0

#.Graph.Graph_Tag_Byte

#.Graph.Graph_Tag_Word

Curve properties

Tag: w #.Graph.Graph_Tag_Word

Save Y at ruler to: by #.GraphData.RulerValueTemperatureB

Description:

Scale: Left

Shape

☑ Use shape

☑ Connect the points

Line width: 2

Shape size: 5

Color

□ Define colors for tag ranges

Color: 🟦

Figure 446: Extended properties "Graph"; settings in Curves tab

Enabling/disabling the marker line

To create a button that enables and disables the marker line, follow the steps below:

- ▶ In the Extended properties "Graph" dialog box, open the Ruler tab, use the Display drop-down menu to select the <Depends on a tag> option, and use the drop-down menu to the right of it to select the xShowRulertag. Now select a green color using the Color field. Enter a value of <3> into the Width field. Finally, use the Save ruler's X value to drop-down menu to select the <sRulerTime> tag.
- ▶ Create a button visualization object.
- ▶ In the Properties 'Button' pane, use the tag drop-down menu to select the <xShowRuler> tag.
- ▶ Double-click on the button to open the Extended properties "Button" dialog box.
- ▶ Open the States tab.
- ▶ Enter Show marker line as the text for the button's <Off> state. Then select a blue color for the background. Now select white as the text color and 18 as the font size. Finally, enable the Word Wrap checkbox.
- ▶ Enter Hide marker line as the text for the button's <On> state. Then select a blue color for the background. Now select white as the text color and 18 as the font size. Finally, enable the Word Wrap checkbox.
- ▶ Click on OK.

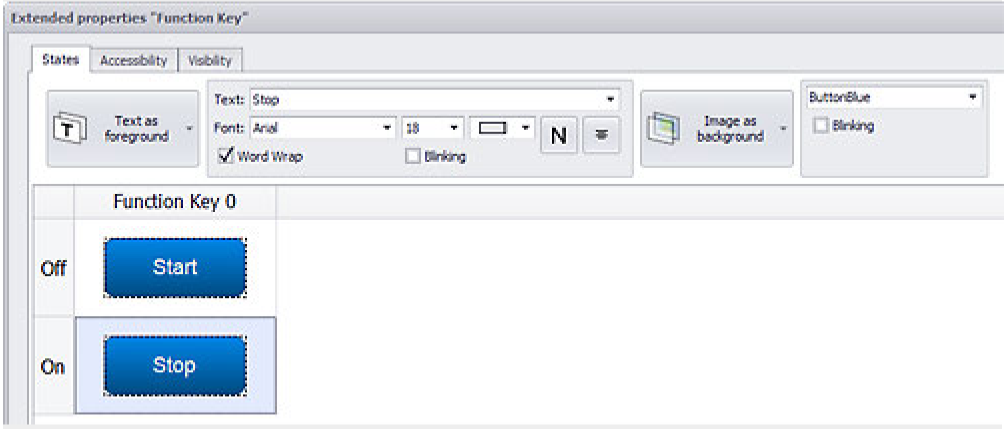
Start and Stop of recording

To create a function key visualization object that will trigger the function for starting and stopping plotting in the graph visualization object, follow the steps below:

- Click on the graph visualization object.
- Move the cursor to the visualization object's top right corner.
- Move the cursor over the 💡 icon and let it stay there until the quick access menu appears.
- Select the <Start/Stop Graph> option in the quick access menu. A function key visualization object will be automatically created and linked to the Start/Stop Graph function.
- Double-click on the function key to open the Extended properties "Function Key" dialog box.
- Open the States tab.
- Select the Text as foreground option for the function key's Off state. Now use the appropriate drop-down menus to select a font size of <18>, <Normal> (N) formatting, and white as the text color. Then type <Start> into the Text field. Finally, set a blue color as the background color or use an image for the background.
- Select the Text as foreground option for the function key's Off state. Now use the appropriate drop-down menus to select a font size of <18>, <Normal> (N) formatting, and white as the text color. Then type <Stop> into the Text field. Finally, set a blue color as the background color or use an image for the background.
- Click on OK.

Clicking or tapping on the button at runtime will start/stop plotting.

Controls\ Function Key



Extended properties "Function Key"

States | Accessibility | Visibility

Text as foreground
Text: Stop
Font: Arial | 18 | N
☑ Word Wrap □ Blinking
Image as background
ButtonBlue
□ Blinking

	Function Key 0
Off	Start
On	Stop

Figure 447: Settings in the States tab of the Extended properties "Function Key" dialog box

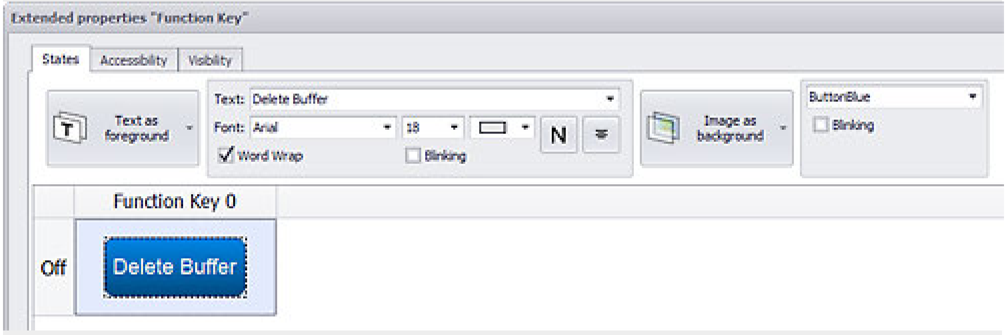
Clearing the buffer

- Click on the graph visualization object.
- Move the cursor to the visualization object's top right corner.

- Move the cursor over the 💡 icon and let it stay there until the quick access menu appears.
- Select the <Delete Graph Buffer> option in the quick access menu. A function key visualization object will be automatically created and linked to the Delete Graph Buffer function.
- Double-click on the function key to open the Extended properties "Function Key" dialog box.
- Open the States tab.
- Select the Text as foreground option for the function key's Off state. Now use the appropriate drop-down menus to select a font size of <18>, <Normal> formatting, and <White> as the text color. Then type <Clear buffer> into the Text field. Finally, set a blue color as the background color or use an image for the background.
- Click on OK.

Clicking or tapping on the button at runtime will delete the plotted graphs from RAM. Plotting will start over from the left edge of the window with the corresponding time.

Controls\ Function Key


Extended properties "Function Key"

Text: Delete Buffer

Font: Arial 18

☑ Word Wrap □ Blinking

ButtonBlue □ Blinking

	Function Key 0
Off	Delete Buffer

Figure 448: Settings in the States tab of the Extended properties "Function Key" dialog box

Loading an archive file

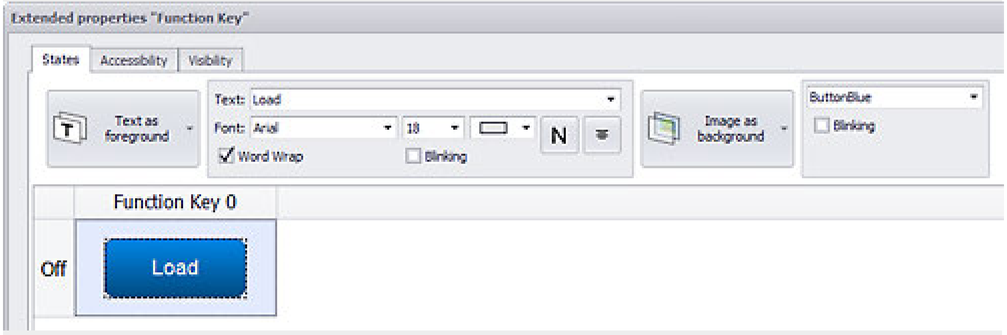
- Click on the graph visualization object.
- Move the cursor to the visualization object's top right corner.
- Move the cursor over the 💡 icon and let it stay there until the quick access menu appears.
- Select the <Load Archive File> option in the quick access menu. A function key visualization object will be automatically created and linked to the Load Archive File function.
- Double-click on the function key to open the Extended properties "Function Key" dialog box.
- Open the States tab.
- Select the Text as foreground option for the function key's Off state. Now use the appropriate drop-down menus to select a font size of <18>, <Normal>

formatting, and <White> as the text color. Then type <Load> into the Text field. Finally, set a blue color as the background color or use an image for the background.

- Click on OK.

Clicking or tapping on the button at runtime will display a list of saved archive files that can be loaded and displayed. Please note that before an archived data set can be loaded, graph plotting must be stopped first.

Controls\ Function Key



Extended properties "Function Key"

States | Accessibility | Visibility

Text as foreground

Text: Load

Font: Arial, 18

☑ Word Wrap

□ Blinking

Image as background

ButtonBlue

□ Blinking

Function Key 0

Off: Load

Figure 449: Settings in the States tab of the Extended properties "Function Key" dialog box

Saving the data set in an archive file

- Click on the graph visualization object.
- Move the cursor to the visualization object's top right corner.
- Move the cursor over the icon and let it stay there until the quick access menu appears.
- Select the <Save Archive File As> option in the quick access menu. A function key visualization object will be automatically created and linked to the Save Archive File As function.
- Double-click on the function key to open the Extended properties "Function Key" dialog box.
- Open the States tab.
- Select the Text as foreground option for the function key's Off state. Now use the appropriate drop-down menus to select a font size of <18>, <Normal> formatting, and <White> as the text color. Then type <Save as> into the Text field. Finally, set a blue color as the background color or use an image for the background.
- Click on OK.

Clicking or tapping on the button at runtime will display a keyboard that you can use to enter the name of the archive file that should be used to save the data set.

Controls\ Function Key

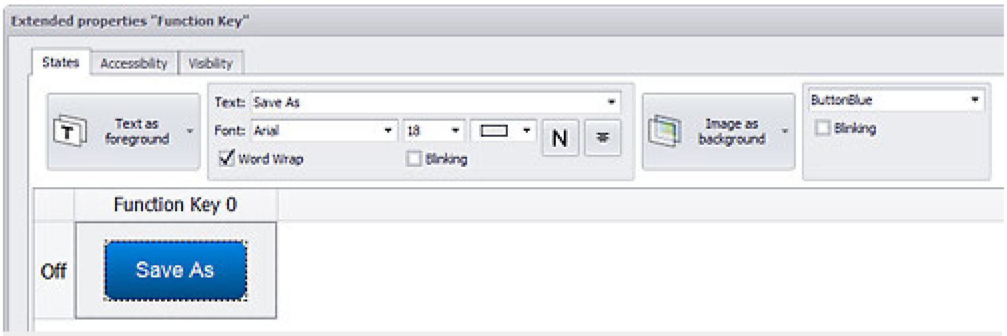


Figure 450: Settings in the States tab of the Extended properties "Function Key" dialog box

7.1.1 Creating a graph block tag object

Before creating a graph visualization object, you should first create a graph block tag object. To create a graph block tag object, follow the steps below:

- ▶ Click on *Home\Graph Blocks*.
- ▶ You can create a folder structure of your choice for the purpose of organizing your graph block tag object. To do this, click on Add Folder and rename the folder, e.g., <Temperature Curve>.
- ▶ Click on Add\Graph Block < and rename the graph block, e.g., <Temperature Curve>.
- ▶ Open the General tab.
- ▶ Enter the number of measured values you want into the No. of samples field, e.g., <10>.
- ▶ Enter the measurement time span, in seconds, into the Time span field, e.g., <60>.

You have now created a graph block object. Please note, however, that nothing will be displayed on the screen yet at this point and that no tags have been assigned yet.

7.1.2 Assign tags to the graph block

Before creating a graph visualization object, you should first create a graph block tag object.

Prerequisites

- A communication system must already be selected, e.g., XSOFT-CODESYS-2
- You must have already imported the tags you want to display and plot from the PLC or created them as PLC tags.

To assign tags to a graph block tag object, follow the steps below:

- ▶ Click on *Home\Graph Blocks*.
- ▶ Select the graph block tag object and open the Tags tab.
- ▶ Click on the drop-down arrow to open the Tag drop-down menu.

- Select the tag that you want to assign to the graph block.
  As a general rule, the tags assigned to graph blocks will be tags imported from the PLC. For test purposes, you can create an internal tag instead, e.g., <TEMPmeasure>. For more detailed information on how to create a tag, please refer to the → section "Creating primitive tags", page 541 section.
- Click the Add button.
- The tag you selected will be shown in the list underneath.
- Click on OK.

You have now assigned a tag to the graph block tag object. Please note, however, that nothing will be displayed on the screen yet at this point.

7.1.3 Creating a graph visualization object and linking it to a graph block

Before creating a graph visualization object, you should first create the corresponding graph block tag object.. Otherwise, there will not be any tag sets available for display.

To create a graph visualization object, follow the steps below:

- Click on *Controls\Graph*.
- Position the object
  Move the cursor to the screen and draw the area for the visualization object.
- Make sure that the visualization object is selected on the screen.
- Click on 📈 to open the Extended properties "Graph" dialog box.
- Open the Curves tab and click on the arrow for the Graph Block drop-down menu to select the tag object that should be displayed on the device screen by the visualization object, e.g., <Temperature Curve>.
- Click on New curve.
- In the Curve properties section, click on the drop-down menu arrow for the Tag Y field. Now select one of the graph block's tags, e.g., <TEMPmeasure[0]>.
- Use the remaining options in the Curve properties section to format the curve according to your needs.
- Click on OK.

You have now created a graph visualization object on the screen. At runtime, this object will display the measured values for the selected tags from the selected graph block.

7.1.4 Navigating during plotting

There are a series of navigation functions available for graph visualization objects.

- The area being shown can be moved within a graph block's plotted data. In other words, while the graph's curve is being plotted to the right, users can scroll to the

right or left as required.
→ "Scroll Left", page 728/ → "Scroll Right", page 729

- The start and end times for the measured values currently being displayed can be shown.
→ "Display Time Graph Start", page 723/ → "Display Time Graph End", page 723
- A marker line used to display the measured values for the various curves at any specific point in the data buffer can be used by using the → "Scroll Left (Ruler)", page 728, → "Scroll Right (Ruler)", page 729, and → "Display Time Graph Ruler", page 723 to position the marker line and display its time information.
- Zoom settings: The visible range can be expanded or compressed in the X and Y axes (see the chapters X axis and Y axis)

Following is a description showing an example of a navigation interface and providing step-by-step instructions for implementing it.

Prerequisites

- A graph block object must have already been created.
- Tags must have already been assigned to the graph block.
- A graph visualization object must have already been created on the screen.

Following is an example of a screen with a navigation interface in GALILEO:

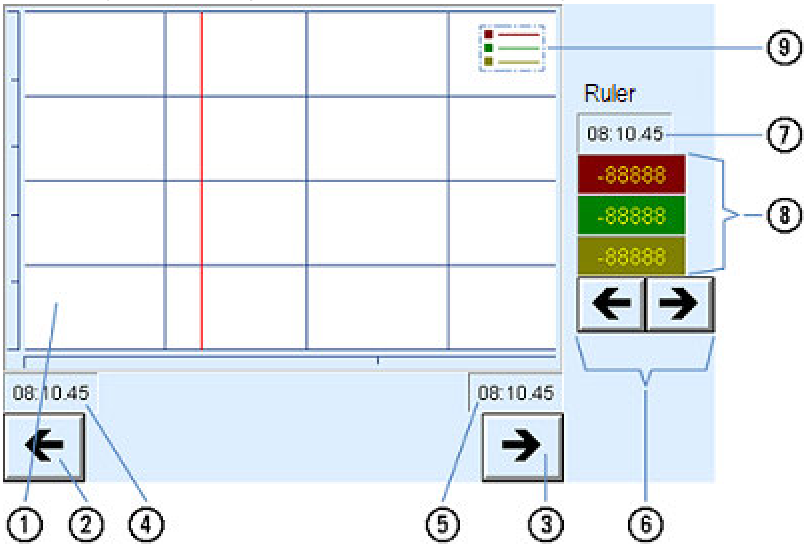


Figure 451: Device screen with graph visualization object, several function key visualization objects for navigation, and several value display/entry visualization objects

1	Visualization object Graph Block
2	Function key visualization object linked to the Scroll Left function
3	Function key visualization object linked to the Scroll Right function
4	Value display/entry visualization object linked to the Display Time Graph Start function

5 Value display/entry visualization object linked to the Display Time Graph End function

6 Function key visualization objects linked to the Scroll Left (Ruler) and Scroll Right (Ruler) functions

7 Value display/entry visualization object linked to the Display Time Graph Ruler function

8 Three value display/entry visualization objects used to display the curve values at the marker line's location. Please refer to the → "Save ruler's X value to", page 713 section as well

9 Legend for the curves being displayed; please note that this legend will only appear in GALILEO, i.e., it will not be shown on the device screen at runtime.

At runtime, the screen on the device will look as shown in the screenshot below:

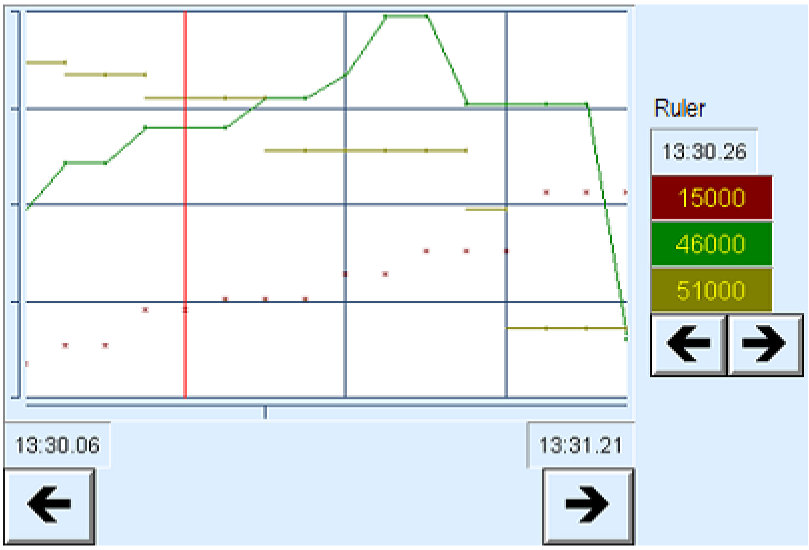
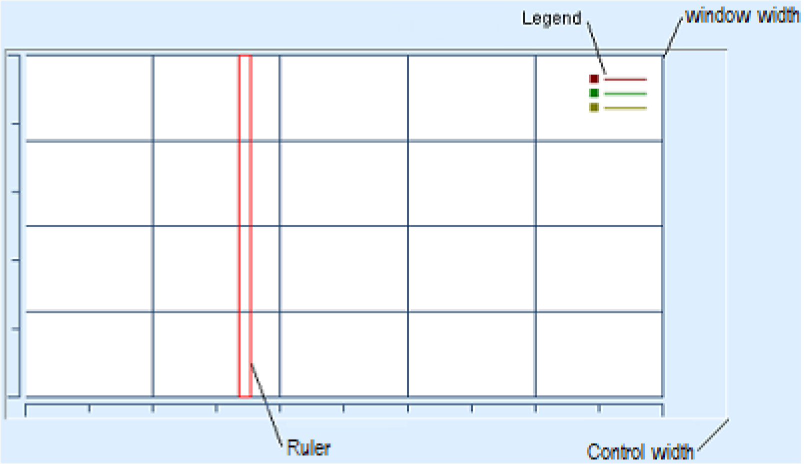
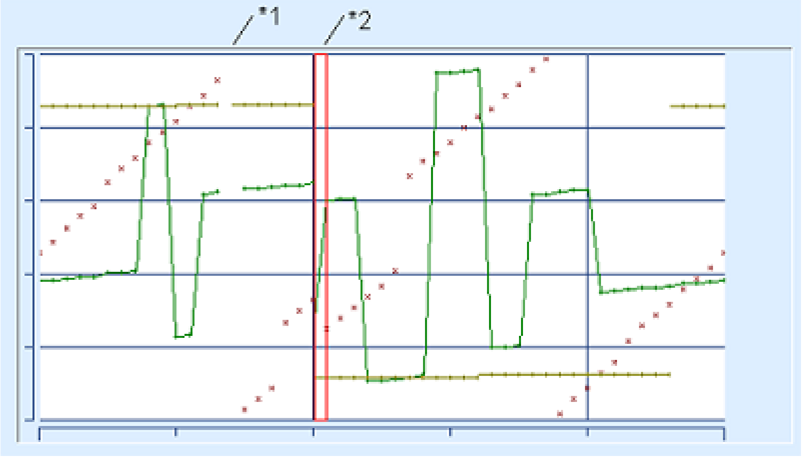


Figure 452: Device screen at runtime



The "curve legend" will show the colors for all the defined curves. Please note that this legend will only appear in GALILEO, i.e., it will not be shown at runtime.

The graph displayed on the device looks as follows:



*1	If curve plotting is stopped and then resumed, plotting will resume one measured value later. In other words, every start/stop will be indicated by having the curve be interrupted for one full interval.
*2	The measured values left of the ruler are the most current data values. The ruler indicates the current plotting position. As soon as the plot reaches the right of the window, the plot will be continued on the left. In this way the oldest tag values are overwritten with the newest values.

7.2 Graph Blocks - Graph Block Manager

The graph block tag object is used to record tag values in individual graph block lists. Within this context, "record" means that the values currently being measured by the PLC will be stored in the device's RAM.

The Start/Stop Graph function is used to start and stop recording these tag values. If recording is ongoing when the device being used is switched off, it will resume automatically when the device is switched back on. All values will be cleared when the device is switched off. In order for these values to be saved in non-volatile memory so that they will not be lost, they must be archived beforehand using the corresponding functions.

A project can have a maximum of 100 graph blocks.
In addition, each graph block list can be used to record the values for a maximum of 255 tags.

Each graph block is uniquely identified by an ID (object number). In addition, each graph block can be used to record the values for a maximum of 255 tags. To visually represent the tags in a graph block, a graph visualization object is required. Of course, you can choose which tags in the graph block you want to display, i.e., the graph can visually represent all the tags in the block or only some of them. However, please note that it is not possible to visually represent tags from various graph block tag objects in a single graph visualization object. These tags can belong to any of the following data types: BIT, BYTE, WORD, DWORD, QWORD, FLOAT.

The tag values in a graph block tag object will be deleted when the corresponding device is switched off.

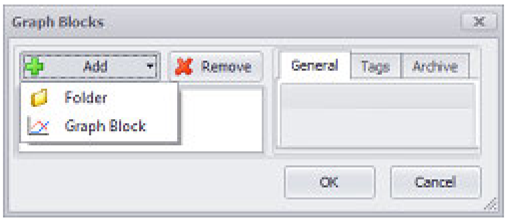
Before a tag's values can be displayed using the graph visualization object, they need to be recorded first using the corresponding graph block.

Functions both for archiving data sets and for displaying historical tag values are available for graph blocks. Moreover, there are a series of functions available for the graph visualization object that make it possible to scroll through graphs and move marker lines.

The tag values recorded with a graph block are always being polled, even if the corresponding graph is not currently showing them.

The properties for a graph block can be configured by clicking on *Home\Graph Blocks* tab.

Register Home

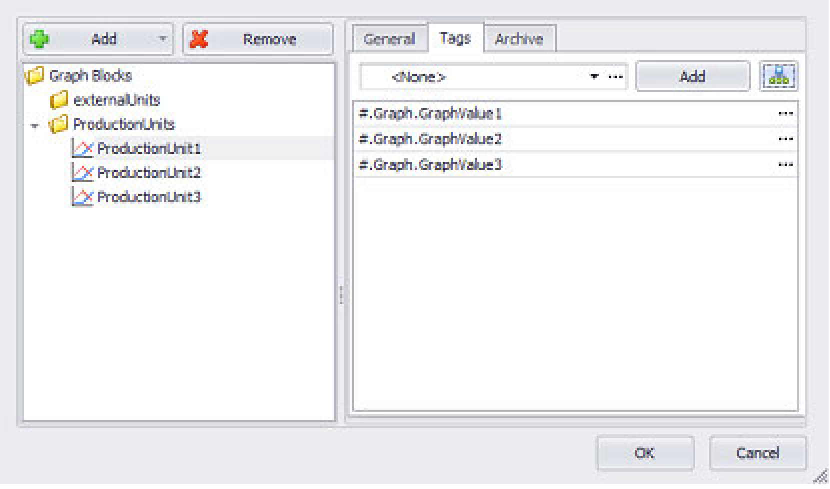


Graph block properties

→ "Add", page 679
→ "Remove", page 679
→ "Register General", page 679
→ "Tags tab", page 683
→ "Archive tab", page 684

Add

Clicking on the Add button will open a drop-down menu with the Graph Block and Folder options. These options can be used to add the corresponding element.

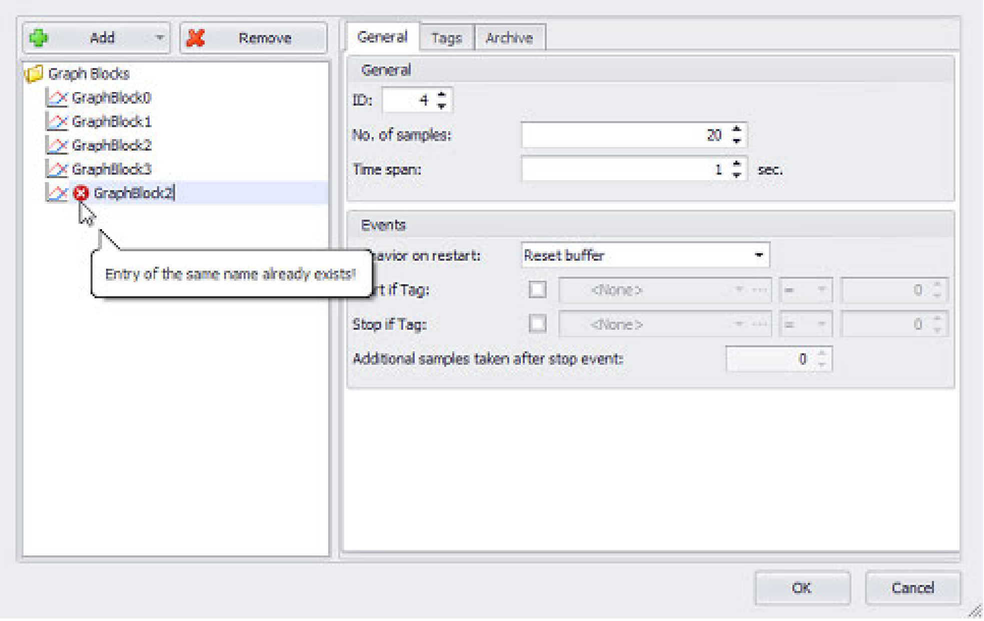


- Graph Blocks
  - externalUnits
  - ProductionUnits
    - ProductionUnit1
    - ProductionUnit2
    - ProductionUnit3

<None>

#.Graph.GraphValue1
#.Graph.GraphValue2
#.Graph.GraphValue3

When added by clicking on Add, graph block tag objects will be numbered in consecutive ascending order. Their name can be changed at any time. Please note that if you try to give a graph block a name that is already in use, an error message will appear.



- Graph Blocks
  - GraphBlock0
  - GraphBlock1
  - GraphBlock2
  - GraphBlock3
  - GraphBlock2

Entry of the same name already exists!

General

ID: 4

No. of samples: 20

Time span: 1 sec.

Events

avior on restart: Reset buffer

rt if Tag: □ <None> = 0

Stop if Tag: □ <None> = 0

Additional samples taken after stop event: 0

Figure 453: Attempting to add a second GraphBlock2 graph block results in an error message

Remove

Deletes the selected folder or graph block.
You can select multiple folders and graph blocks for deletion if necessary.

Register General

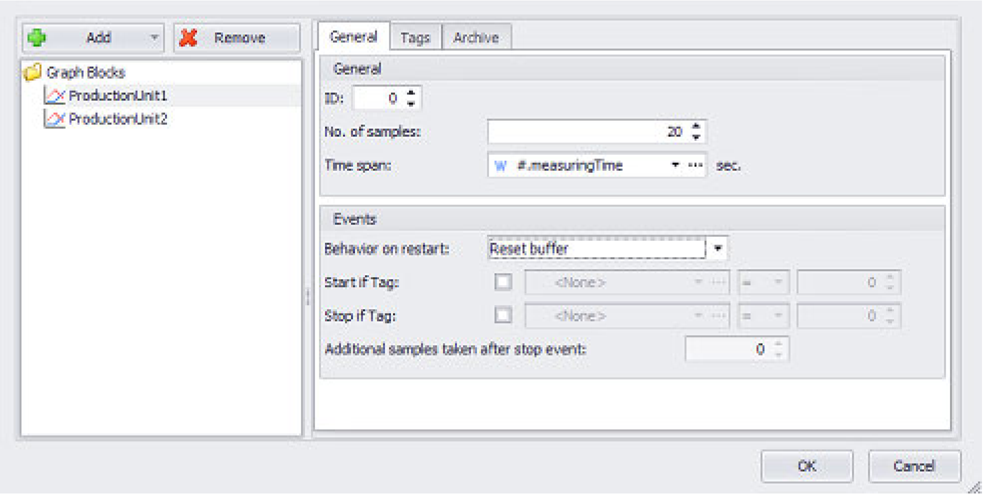
The General tab can be used to configure all the properties related to the measured values' settings and how they will be displayed.

An ID needs to be assigned to every single graph block. This ID is later used by the Start/Stop Graph 0..99 function to start and stop plotting. In addition, the settings here are used to set the number of measured values, or samples, that will be recorded and plotted for each tag in the graph block, as well as the corresponding interval. Please note that these settings will apply to all the tags in the graph block! When the number of measured values set here is reached, the oldest values will be overwritten with the latest ones. Make sure that the time span is appropriate for the polling rate being used! As a general rule, the time span set in the Time span field in this tab should not be less than the polling time span. Time span field in this tab should not be less than the polling time span. The polling settings can be configured in the properties for the corresponding tag by going to *Project Tree\Tags\<tag>.*

RAM size will depend on the device model and device configuration being used. As for the amount of RAM available for a graph block, this is determined by the size of the project, as well as by the texts, recipes, screen files, and screen sizes being used. You can use the following formula to determine how much RAM is needed for each graph block:

RAM [bytes] = (22 + number of samples * (8 + number of curves * 4))

Home\ Graph Blocks\General tab



ID: 0

No. of samples: 20

Time span: #.measuringTime sec.

Behavior on restart: Reset buffer

Start if Tag: ☐ <None> = 0

Stop if Tag: ☐ <None> = 0

Additional samples taken after stop event: 0

Figure 454: Graph Blocks dialog box showing the properties for a graph block

General section

ID (Object Number)

You can assign an ID of your choice to the graph block you are configuring. This ID (object number) is required by the Start/Stop Graph function and it does not have to be the same as the ID of the graph visualization object using the graph block. By default, IDs will be automatically assigned in ascending order.

No. of measured values

This setting, which applies to all the tags in the graph block being configured, can be used to set the maximum number of measured values. The maximum number of measured values will be limited by the size of the corresponding RAM, flash, and hard drive memories, and accordingly will vary depending on the specific device being used.
see → section "Limits per project", page 1133
Make sure that the device has enough RAM and flash memory to handle your requirements! When the number of measured values set here is reached, the oldest values will be overwritten with the latest ones, just like in a circular buffer. For the formula used to calculate the required RAM, please see above in this section.

Time span

Used to specify the interval between two measured values. This time span is defined in seconds.

Make sure that the PLC protocol you are using supports the time span and that the polling rates for the tags are set accordingly. If large amounts of data are transferred between the device and the PLC, please note that it will not be possible to guarantee the time span for recording and plotting data. Likewise, short time spans and large amounts of data may cause the device to respond slowly to operator input. For detailed information on how data communication works and what kinds of things you need to keep in mind in regard to polling, please refer to the → section "Data communication principles", page 580 section.

The time span setting can be defined either as a constant value or dynamically by using a tag.

Defining dynamic values with tags

Instead of entering a fixed value, you can link the field to a tag, so that the value can be changed dynamically. To link to a tag, follow the steps below:

- Place the cursor inside the field.
- Right-click to open the context menu and select the Use Tag option.

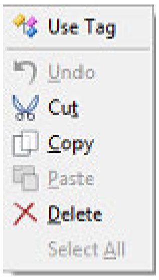


Figure 455: Context menu with Use Tag option

The field will go from looking like this: to looking like this: .

▶ Click on the drop-down menu arrow and select the tag that should represent the value.

▶ Clicking on the ... icon will open the Properties <tag name> dialog box. Make sure that the settings for the tag are correct, e.g, Default, Maximum. If necessary, you can also use tags here in order to define limits. During compilation, the system will check whether there are recursive tag references and, if so, will output an error message.

→ All value tags are always declared in decimal format regardless of the selected display format (decimal, hex, binary, octal, BCD)!

→ All values are PLC values. They will be transmitted to the PLC the way they are, i.e., without being converted.

Events section

During Reset

- Reset buffer
  If this option is selected, the contents in the Graph visualization object's window will be cleared when the Start/Stop Graph 0..99 function is used to start plotting, and plotting will accordingly start from the left edge of the window.
- Append to buffer
  If this option is selected and plotting starts, the new measured values will be appended to the existing data.

→ If the archive name is changed, the recorded measured values that were not yet saved will be deleted before the next start.

The settings in the Archive tab can be used to specify what should happen when the device starts. Please refer to the descriptions for the Start if Tag and Stop if Tag fields, as well as for the → "Start/Stop Graph Block", page 732 function.

Start / Stop if Tag

These options can be used to set conditions that define when the values of the tags in the graph block should start or stop being recorded. Each condition is defined with three fields: the first one is used to select the tag you want for the condition, the second one to choose a logical operator, and the third one to enter the required comparison value.

Starting

The values of the tags in the graph block will start being recorded when all of the following conditions are met:

- Plotting has currently not been started.
- The start condition is met.
- The stop condition is not met.

Stop

The measured values of the tags in the graph block will stop being recorded when all of the following conditions are met:

- The stop condition is met.

Please note that the start and stop settings affect the → "Start/Stop Graph Block", page 732 function!

If the stop condition is being met, it will NOT be possible to start a graph with the function.

In contrast, the start and stop conditions are irrelevant when stopping a graph.

Additional samples taken after stop event

You can also set a delay so that the graph block will continue recording a specific number of measured values after a stop. In other words, after the stop condition is met, the number of measured values you specify here will be recorded before recording is actually stopped.

Tags tab

The Tags tab can be used to select the tags for which values should be recorded using the graph block tag object. These tags can belong to any of the following data types: BIT, BYTE, WORD, DWORD, QWORD, FLOAT.

Tags in a graph block will be polled in all screens as soon as a plotting process that uses these tags is started.
Because of this, it is recommended to assign tags from as few data blocks (arrays, structures) as possible.

Home\ Graph Blocks\Tags tab

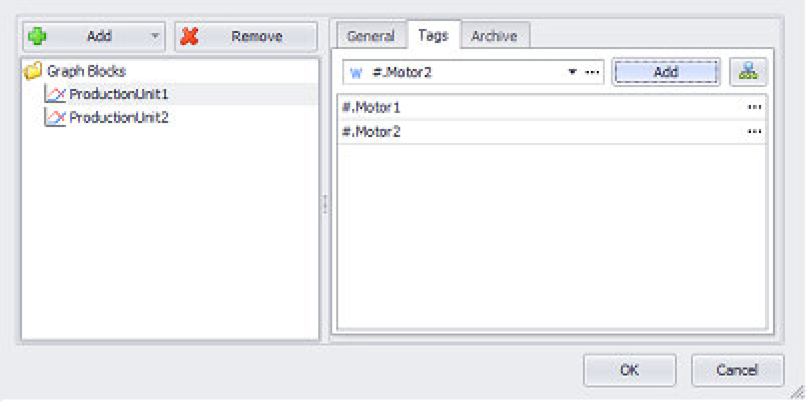


Figure 456: The Motor1[1] tag being added to the ProductionCell2 graph block. The values for this tag will be recorded with this graph block in the future

Variable

When a tag is added here, this means that its value will be recorded by the graph block at runtime. Clicking on the drop-down menu arrow will open a list of available tags (BIT, BYTE, WORD, DWORD, QWORD, FLOAT). Please note that in order for the tag you want to be shown in this list, the tag must first be imported or created in the Tags tab.

A parameter can also be specified instead of a tag. In this case, the Tag field will turn into a Type field. In order to be able to work with parameters, the visualization object must be placed on a user control. For detailed information on how to work with parameters, please refer to the → "Using user controls to optimize work", page 279 section.

Add

Clicking on the Add button will add the tag being shown on the left to the list below so that its values can be recorded.

Button ...

Limits

Clicking on the ... icon will show the limits for the value tag in a separate dialog box. You can also reach this dialog box via *Project Tree\Tags\<Value tag>\Properties*. When a tag is created, it will already be assigned limits based on the value range defined for it. This value range can be signed or unsigned and will depend on the relevant tag's data type.

The limits can be adjusted as necessary within the value range. In the case of bit tags, you will only be able to specify a default value.

For a description of the tag properties and limits, please refer to: → section "Numeric tag limits", page 563.

Tag Address

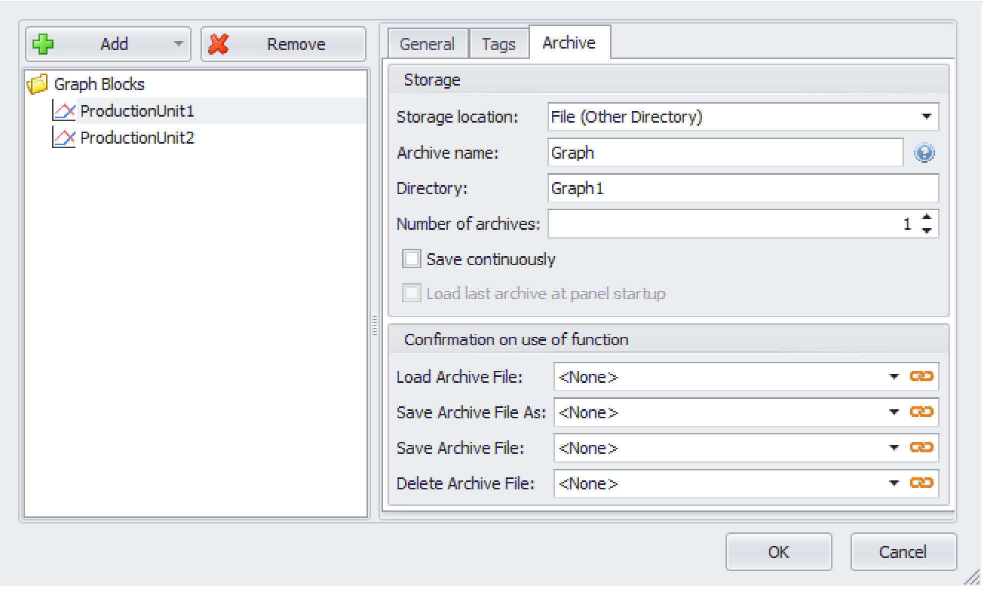
This field will only appear if you select a PLC tag, i.e., it will not appear for internal tags.
It will show the PLC address for the selected tag. The properties for PLC tags are described in greater detail in a different section – please refer to → section "Tag address", page 569.

If a single tag is transmitted to more than one communication, the corresponding settings will need to be configured in *Home\Gateway*. For detailed information on how to work with a gateway, please refer to → section "Using the gateway functionality to communicate with multiple PLCs", page 596.

Archive tab

This tab is used to configure the settings for the archive files corresponding to the graph block tag object.

Home\ Graph Blocks\Archive tab



Graph Blocks
- ProductionUnit1
- ProductionUnit2

Storage

Storage location: File (Other Directory)
Archive name: Graph
Directory: Graph1
Number of archives: 1
□ Save continuously
□ Load last archive at panel startup

Confirmation on use of function

Load Archive File: <None>
Save Archive File As: <None>
Save Archive File: <None>
Delete Archive File: <None>

Figure 457: Graph Blocks dialog box showing the properties for a graph block

Save

An archive file can be saved for each graph block tag object. The Storage section contains all the properties used to configure how the archive file will be stored. More specifically, the archive file can be stored in a DATA subdirectory or in a different directory, where DATA is an actual directory within the project directory. In addition to this, this section can be used to select the format for the date and timestamp that is added for every tag value in the archive file.

Storage location

The archive file can be stored in the following directories:

No (RAM)	The measured value data is only located in the volatile memory (RAM memory), i.e. data is lost when the device is switched off.
File (DATA directory)	When this option is selected, the archive file will be saved at the folder level to the following path: *<Project directory>\DATA\<directory>*.
File (Other Directory)	When this option is selected, the archive file will be stored in the directory specified in the Directory field. Example: Storage location on the SD- memory card: \Storagecard\<Directory>. Storage location on USB memory: \UsbStorage\<directory> Relative paths If you enter a relative path, the subdirectories and files will be created in the project directory in addition to the 3 sequential programs APPL, RUNTIME, and DATA directories. → Do not use backslash characters ("\"). Do not enter a drive name at the beginning.

Archive name

This field will only be enabled if an option other than None (RAM)< is selected in the Storage location> field.

Name of archive file for the measured values of the tags in the graph block. The file extension for these archive files is GDB. If the Number of archives field is set to <1>, so the measured values will be saved in a file called <archive name>.GDB. If you use more than one archive file, a counter suffix will be appended to the archive name.

Following place holders can be used in the archive name:

@start_date@	Date of first archive entry in the format DD-MM-YYYY
@start_time@	Time of first archive entry in the format hh-mm-ss
@start_datetime@	Date and time of first archive entry in the format DD-MM-YYYY_hh-mm-ss
@stop_date@	Date of last archive entry in the format DD-MM-YYYY
@stop_time@	Time of last archive entry in the format hh-mm-ss
@stop_datetime@	Date and time of last archive entry in the format DD-MM-YYYY_hh-mm-ss

The archive name can be defined either as a constant value or dynamically by using a tag.

If a dynamic archive name is used and the "Save continuously" option is enabled, the tag that contains the archive name should be stored in non-volatile memory without fail. If the tag is not stored in non-volatile memory, an archive file will not be generated in the event of a power failure!

The following is also important: If the value of the tag with the dynamic name changes while the device is booting up, the previously stored number will be incremented. In other words, the numbering for the newly created archive file will not necessarily start from "1".

→ If the dynamic archive name was changed, the following actions are then carried out when the Graph Block is started:

- The numbering of the archive name starts again at 1.
- The RAM will be cleared.

Defining dynamic strings with tags

To do this, you need to link the field to a tag instead of entering a fixed string, so as to ensure that the string can be modified dynamically. To link to a tag, follow the steps below:

▶ Place the cursor inside the field.

▶ Right-click to open the context menu and select the Use Tag option.

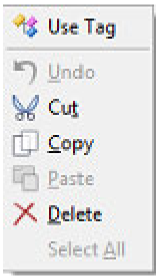


Figure 458: Context menu with Use Tag option

The field will go from looking like this: 0 ⬍ to looking like this: ▾ ....

- ▶ Use the drop-down menu to select the tag that should represent the string.
- ▶ Clicking on the ... icon will open the Properties <tag name> dialog box. which can be used to enter the number of characters and a default string.

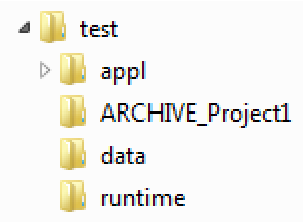
Folder

This field will only be enabled if an option other than None (RAM)< is selected in the Storage location> field.

This field can be used to select the directory where the archive file will be stored. If you select DATA as the storage location, a subdirectory with the name you enter will be created under <*Project directory*>*\data*.

If you select Other Directory as the storage location, the following rules apply:

- Absolute paths
  The directory specified in the *Directory* field will be created, e.g., C:\User\Eaton\Galileo
- Relative paths
  (Do not use backslash characters ("\") and do not enter a drive name at the beginning.)
  The directory specified in the *Directory* field will be created as a subdirectory in the project directory, e.g., ARCHIVE_Project1.



If the subdirectory does not exist, it is created by the runtime program.

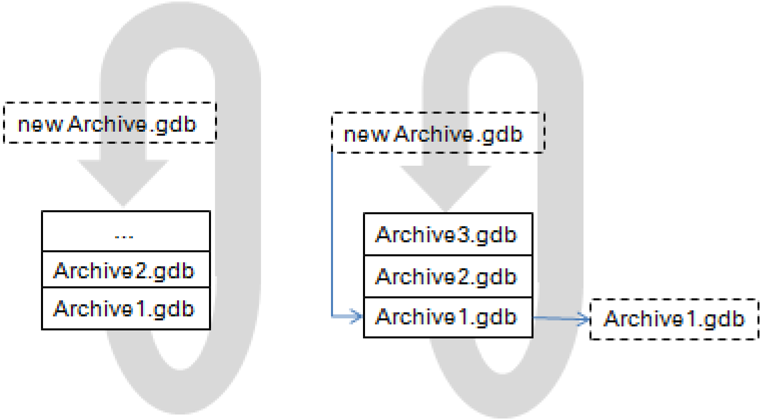
Number of Archives

This field will only be enabled if an option other than None (RAM)< is selected in the Storage location> field.

The maximum number of archive files is 32,767.

If the number of archive files is greater than 1, the graph block will switch to the next archive file every time it starts recording. Once the last archive file is filled with data, the very first archive file will be used next. To distinguish between these archive files, each file will have the specified archive name plus a number (1–n). The number of archive files can be set to any number of your choice, provided it does not exceed the maximum limit. If, for example, the number of archive files is set to 3 and the archive name entered is "Archive," the archive filenames used will be as follows: "Archive1.gdb", "Archive2.gdb", "Archive3.gdb".

If graph block recording is active when the device is switched off, it will start automatically the next time the device is switched on. In this case, the graph block will also switch to the next archive file automatically. For a description of how the contents in an archive file are organized, please refer to the → "Archive Graph Block data - Archive measured values", page 718 section



Save continuously

This field will only be enabled if an option other than None (RAM)< is selected in the Storage location> field.

☑ If the checkbox is enabled, the measured values of all the tags in the graph block will be continuously stored in a file with the <*.TDB> extension. This binary TDB file will store the data in a kind of circular buffer. When recording ends, as well as when the device is switched off and then back on, the data will be automatically saved in the GDB ASCII file. For information on how the contents in an archive file are organized, please refer to the → "Archive Graph Block data - Archive measured values", page 718 section

□ If the checkbox is disabled, the measured values will be stored in RAM (volatile memory). In this case, the → "Save Archive File", page 725 and → "Save Archive File As", page 726 functions can be used to save the data in a file.

→ This option is only available for certain device models.
This option can be used with all drives, with the exception of the OS drive.

> → If the TDB file cannot be accessed, an error message will be output and recording will be stopped. This will also occur if a Save operation fails due to the use of an invalid name or if a Save or Save As operation fails due to a missing data storage device.

Load last archive at panel startup

This field will only be enabled if an option other than None (RAM)< is selected in the Storage location> field.

If the checkbox is enabled, the archive file that was last used will be automatically loaded when the device starts and the corresponding measured values will be displayed with the graph visualization object.

> → Please note that this option can only be used if the graph block's tag values are saved continuously and the Save continuously checkbox is enabled.

Confirmation on use of function

This section can be used to add confirmation prompts to the following functions:

- → "Save Archive File As Var", page 727
- → "Save Archive File As", page 726
- → "Save Archive File", page 725
- → "Delete Archive File", page 722

Confirmation prompts are prompts that require the device operator to confirm a function before the function is executed. For detailed information on how confirmation prompts are implemented in GALILEO, please refer to the → "Using confirmation prompts to confirm whether functions should be run", page 340 section.

There is a global setting for defining whether a visualization is 🔗 linked or ⛓️‍💥 unlinked. In order for the 🔗 and ⛓️‍💥 chain link icons to be displayed, there must be more than one visualization in a group.
For more information, please refer to: → section " Web visualization", page 1070

7.3 Graph

Graph visualization objects are used to graphically display recorded tag values for tags of type BIT, BYTE, WORD, DWORD, QWORD, FLOAT.
In turn, these values are recorded using graph block tag objects.
To specify the tags that should be displayed, the corresponding properties must be configured for the graph object under *Controls\ Graph\Extended properties "Graph"*\Curves tab.

A project can have a total of 0 to 99 graph visualization objects, with each graph being uniquely identified by an ID (object number).
Moreover, each graph visualization object can be used to display the values of a maximum of 32 tags, i.e., the number of curves that a graph object can display is limited to a maximum of 32.

Every graph visualization object can be used to graphically display the tags in a graph block. You can choose which tags in the graph block you want to display, i.e., the graph can visually represent all the tags in the block or only some of them. However, please note that it is not possible to visually represent tags from various graph blocks in a single graph visualization object.

The tag values will be displayed in a window. The maximum window width, and the maximum number of measured values shown at any one time as a result, is limited by the object width, i.e., for an X-axis resolution of 640, a maximum of 640 measured values can be shown at any one time. This also means that even if the graph block has more measured values than this limit, it will still only be possible to show a maximum of 640 measured values within the corresponding graph visualization object.

When the device being used is switched off, the tag values in the graph block will be deleted.
The graph block can also be configured to delete these values when plotting starts by using the appropriate option in the graph block's extended properties so that the archive will be reset. For more detailed information on how the archive can be reset, please refer to the descriptions in the → "During Reset", page 682 and → "Start/Stop Graph Block", page 732 sections.

During plotting, the marker line can be moved to past tag values as well so that they will be shown explicitly.

Moreover, there are a series of functions available for the graph visualization object that make it possible to scroll through graphs and move marker lines. Functions both for archiving data sets and for displaying historical tag values are available for the graph block list in a graph block.

There are several ways to enlarge the visualization object.
One is to use the Zoom In and Zoom Out functions in order to enable device users to zoom as needed. Another is to define a smaller scale range. This scale range is

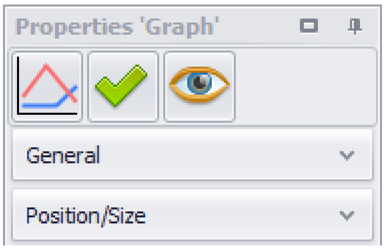
defined by a minimum value and maximum value, both of which can be defined using tags so that device operators will be able to change the values at runtime with the appropriate controls.

The menu provides a group of the most important control options.

Prerequisites

To be able to follow the descriptions below, make sure you click on a graph on the current screen.

Controls\ tab



Properties 'Graph'

General

Position/Size

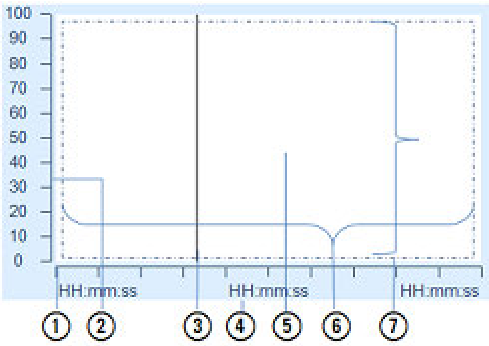
'Graph' properties

→ " Extended properties "Graph"", page 693

✓

→ section "General", page 691

Definition of Terms Graph Block



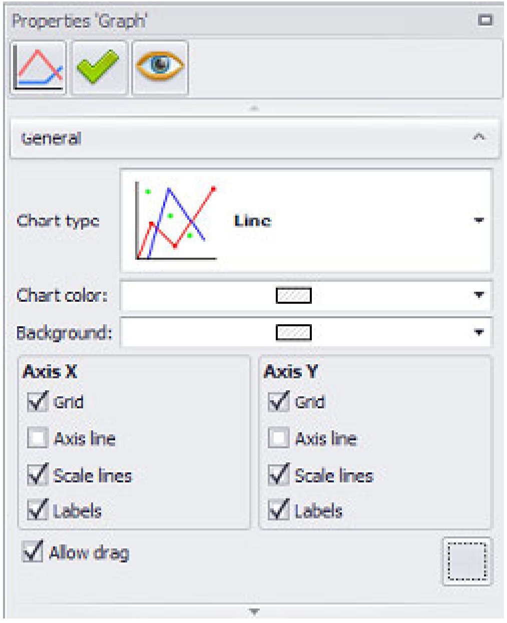
1	X Axis
2	Y Axis
3	Ruler
4	Background
5	Graph display area
6	Windows width
7	Window height

General

The General section can be used to configure all the basic properties that affect how the visualization object will be displayed, with the exception of its size and position.

The Properties 'Graph' pane provides a subset of the properties available in the Extended properties "Graph" dialog box.
Accordingly, its settings are complemented by the → " Extended properties "Graph"", page 693 found in the → "Register General", page 694 and → "Axes tab", page 695 tabs.

Controls *Graph*



Properties 'Graph'

General

Chart type: Line

Chart color:

Background:

Axis X
☑ Grid
□ Axis line
☑ Scale lines
☑ Labels

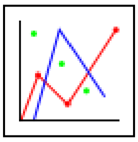
Axis Y
☑ Grid
□ Axis line
☑ Scale lines
☑ Labels

☑ Allow drag

Figure 459: Properties Graph pane, General section

Chart type

Used to select the chart type:

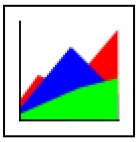


Line

Displays the evolution of values over a time interval

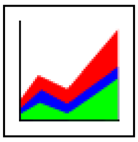
The X-axis will represent a time value.

This chart type can be used to show how values evolve over a period of time – especially if there is a large number of values and the order in which they are displayed is important.



Area

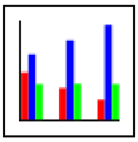
Displays the evolution of values over a period of time



Stacked Area

Displays the evolution of the proportion represented by individual values over a period of time

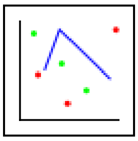
The value that follows uses the start of the previous value and is used as a basis.



Column

Used to illustrate how individual elements compare against each other

Displays values as bars.



Scatter XY
The X-axis will represent a tag value.
Displays arrays; can be used to compare value pairs.
Can be used to illustrate relationships between numeric values in different data series or to plot two number groups as a series of XY coordinates.

Chart color:

Background:

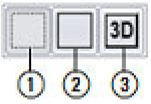
You can change the color if you want. The following options are available for this purpose:Transparent
Standard Colors
More Colors...The most recently used colors will be shown (user-specific).

X Axis
Y Axis

Used to configure the appearance of the X-axis and Y-axis.

Allow drag

These touch gestures can only be used with PTC devices such as the XV300 and XP500.
This function is also supported by GALILEO Open applications.
If this checkbox is enabled ☑,
it will be possible to use a flick gesture to move the graph display and the marker line through the window at runtime.

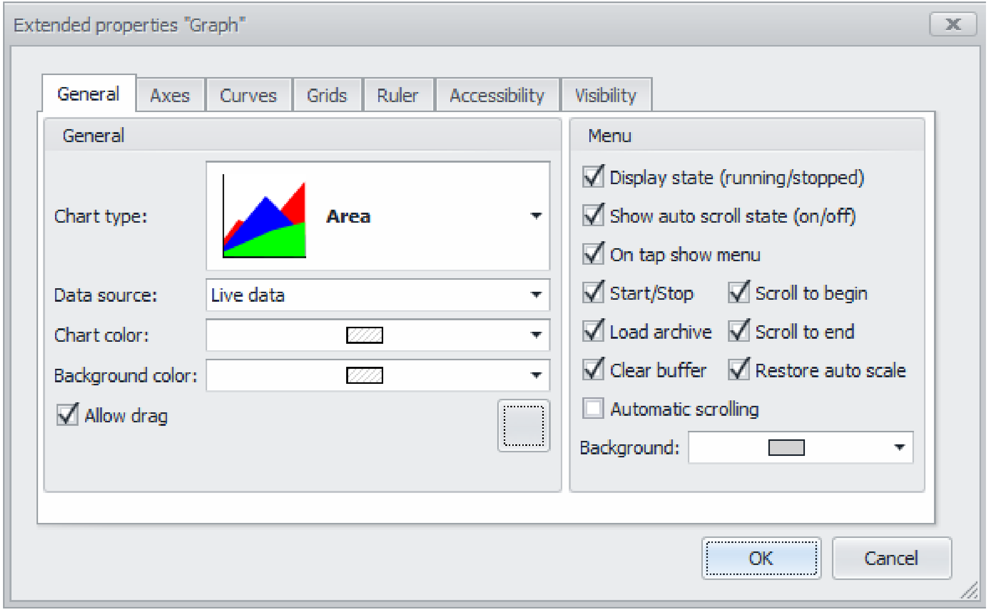


Display style
The visualization object can be displayed with any one of three different designs.

1. No frame
2. with frame
3. 3D frame

Extended properties "Graph"

Controls *Graph*



Extended properties "Graph"

General | Axes | Curves | Grids | Ruler | Accessibility | Visibility

General

Chart type: Area

Data source: Live data

Chart color: ▨

Background color: ▨

☑ Allow drag

Menu

☑ Display state (running/stopped)

☑ Show auto scroll state (on/off)

☑ On tap show menu

☑ Start/Stop ☑ Scroll to begin

☑ Load archive ☑ Scroll to end

☑ Clear buffer ☑ Restore auto scale

☐ Automatic scrolling

Background: ▭

OK Cancel

Figure 460: Window Extended properties Graph, General section

Register General

The General section can be used to configure the basic properties for the visualization object.

Additional settings can be configured in addition to the settings available in the → "General", page 691 properties.

Data source: Used to define where the data being displayed should come from
The options available in the individual menus will depend on the specific selection.

Live data Graphic mode

*.GDB file Saved graph block file record

Tag array Mode for displaying values of tag arrays in the project

Menu

The tap-graph menu on a graph provides quick access to a number of functions.

A tap-graph menu can be optionally displayed on a graph visualization object. The following settings are used to configure this menu and define
which buttons will be shown. Please note that certain buttons may not be available depending on the selected data source.

The tap-graph menu will be opened after a delay and will be automatically closed after three seconds if no action is detected.

Display state (running/stopped) An icon at the top right of the graph will be used to show whether recording is running

Show auto scroll state (on/off):	An icon at the top right of the graph will be used to show whether automatic scrolling is on or off
On tap show menu	Used to define whether the tap-graph menu should appear on the graph if the graph is tapped

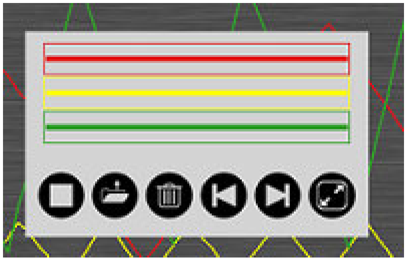


Figure 461: Example of a tap-graph menu for three values

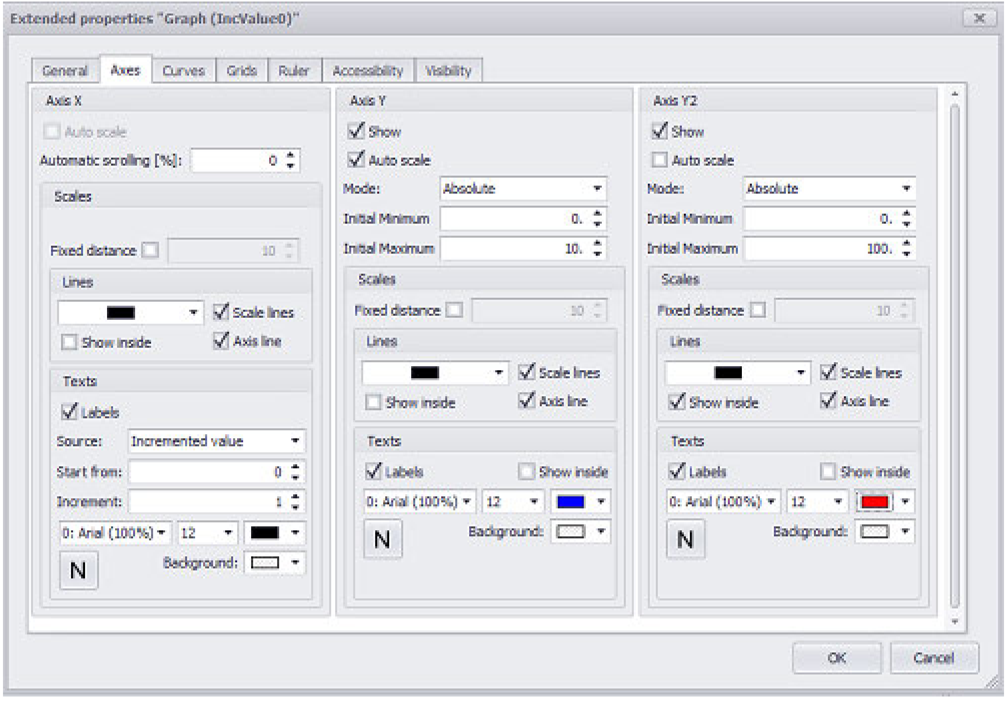
The following functions will be available on the tap-graph menu if they are ☑ enabled:

- Start/Stop
- Load archive
- Clear buffer
- Scroll to begin
- Scroll to end
- Restore auto scale
- Automatic scrolling

Background	You can change the color if you want. The following options are available for this purpose:Transparent Standard Colors More Colors...The most recently used colors will be shown (user-specific).

Axes tab

Controls *Graph*



Extended properties "Graph (IncValue0)"

General | Axes | Curves | Grids | Ruler | Accessibility | Visibility

Axis X
- □ Auto scale
- Automatic scrolling [%]: 0
- Scales
  - Fixed distance □ 10
  - Lines: ☑ Scale lines, □ Show inside, ☑ Axis line
  - Texts: ☑ Labels
  - Source: Incremented value
  - Start from: 0
  - Increment: 1
  - 0: Arial (100%) 12
  - N Background:

Axis Y
- ☑ Show
- ☑ Auto scale
- Mode: Absolute
- Initial Minimum 0.
- Initial Maximum 10.
- Scales
  - Fixed distance □ 10
  - Lines: ☑ Scale lines, □ Show inside, ☑ Axis line
  - Texts: ☑ Labels, □ Show inside
  - 0: Arial (100%) 12
  - N Background:

Axis Y2
- ☑ Show
- □ Auto scale
- Mode: Absolute
- Initial Minimum 0.
- Initial Maximum 100.
- Scales
  - Fixed distance □ 10
  - Lines: ☑ Scale lines, ☑ Show inside, ☑ Axis line
  - Texts: ☑ Labels, □ Show inside
  - 0: Arial (100%) 12
  - N Background:

OK Cancel

Figure 462: Extended properties Graph dialog box, Axes tab when tag array is selected as the data source in the General tab

X Axis

This property is used to define whether the values in the Graph will be shown as a function of time so that the X axis will be used as a time axis. Otherwise, the values in the Graph visualization object will be shown as a function of a PLC tag.

Certain fields may be disabled depending on which option is selected in this field.

STOP

CAUTION

Changing the options configured for the X-axis can result in the settings under the Curves tab being lost!

GALILEO will show the ⚠ icon to point out that changing the selected option will result in other settings being lost. If you happen to change it by accident, simply close the Extended properties dialog box by clicking on Cancel.

Automatic scrolling

This field will appear if a time format has been selected for the axis.

The automatic scrolling setting defines the percentage of the graph after which the entire window will be refreshed or the percentage of the graph that curves will be moved. It is used to ensure that the graph will not be refreshed too frequently, which can result in objectionable effects when a low sample rate is being used.

The value entered for automatic scrolling must fall between 0 and 100.

→ This setting will only take effect of graph block recording has started and the current measured value is within the graph.

Automatic scrolling example [%]

n = 0 %

The graph will be moved to the left by one measured value after every measured value. To do this, the oldest measured value on the left will be deleted from the graph, the graph will be moved to the left, and the new measured value will be plotted. This procedure will be repeated with every new measured value.

n = 10 %

As soon as the current measured value is at the outer right edge of the window, the graph will be moved 10% to the left and the new measured values will start to be plotted. Accordingly, the next measured values will be plotted without the graph having to be moved until the values reach the outer right edge of the window again, at which point the procedure will be repeated.

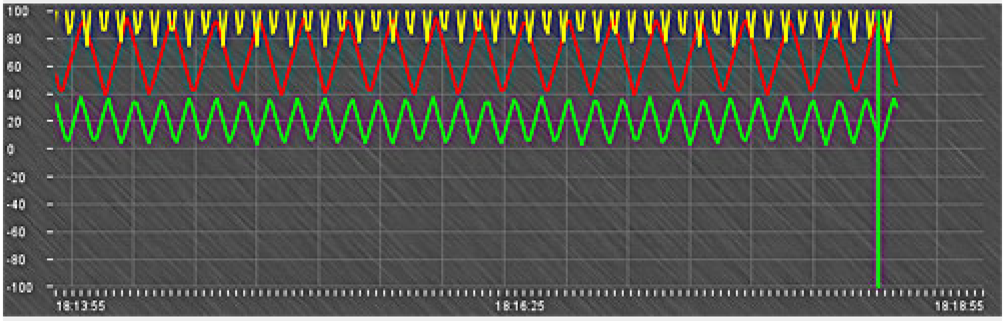
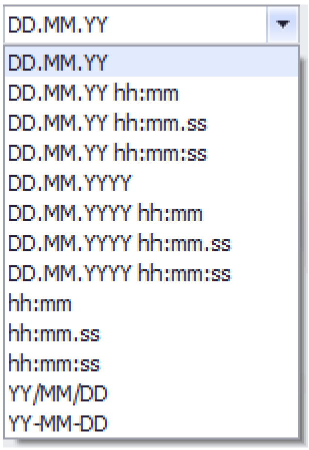


Figure 463: Graph when automatic scrolling = 10 %

Time/Date format
Time and Date Display Format



DD.MM.YY

DD.MM.YY
DD.MM.YY hh:mm
DD.MM.YY hh:mm.ss
DD.MM.YY hh:mm:ss
DD.MM.YYYY
DD.MM.YYYY hh:mm
DD.MM.YYYY hh:mm.ss
DD.MM.YYYY hh:mm:ss
hh:mm
hh:mm.ss
hh:mm:ss
YY/MM/DD
YY-MM-DD

Figure 464: Drop-down menu for Date/Time format field in Extended properties "Graph" dialog box

DD	Day (1...31)
MM	Month (1...12)
YY	Year two digit (e.g. 15)
YYYY	Year four digit (e.g. 2015)
hh	Hour (0...23)
mm	Minute (00...59)
ss	Second (0...59)

Default time interval

This field will appear if a time format has been selected for the axis.

The default time interval setting is used to define the intervals at which the measured values should be plotted. In other words, this means that you can choose to record a lot of measured values while only displaying a few of them, effectively rendering recording and plotting independent from each other.

Automatic scrolling example [%]

n = 0 %

The graph will be moved to the left by one measured value after every measured value. To do this, the oldest measured value on the left will be deleted from the graph, the graph will be moved to the left, and the new measured value will be plotted. This procedure will be repeated with every new measured value.

n = 10 %

As soon as the current measured value is at the outer right edge of the window, the graph will be moved 10% to the left and the new measured values will start to be plotted. Accordingly, the next measured values will be plotted without the graph having to be moved until the values reach the outer right edge of the window again, at which point the procedure will be repeated.

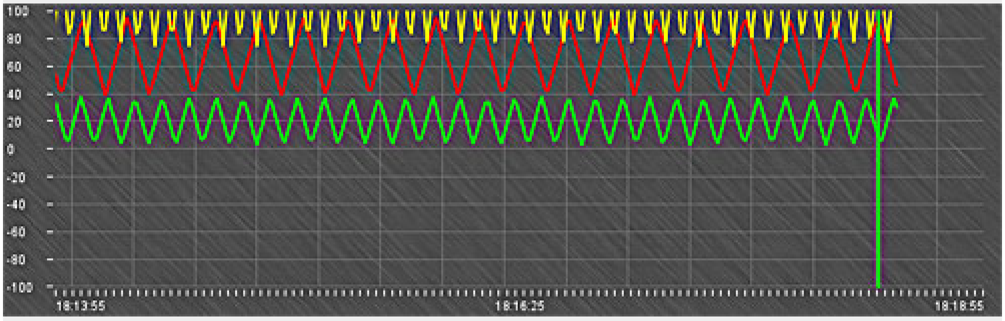


Figure 465: Graph when automatic scrolling = 10 %

Y Axis

The Y-axis is used for vertical axis subdivisions.

Show — The Y-axis will be shown if this checkbox is enabled

Auto scale — The two Y-axis dimension scales can have the "Auto scale" option, which detects the minimum and maximum of all the value series assigned to the scale. When this option is enabled, the graph will be drawn with the optimal Y-axis zoom factor so that the height of all assigned values is maximized.

Type — Dimensioning, "Absolute" and "Percentage" are the two options available

Absolute

If you select this option, the tags' values will be displayed as absolute values. In other words, the window height and the maximum Y-axis value that can be displayed will be the same as the maximum measured value.

Percentage

If you select this option, the tags' values will be displayed as a percentage. In this case, the window height and the maximum Y-axis value that can be displayed will be equal to 100%.

The Initial Minimum and Initial Maximum have no effect on the scale. The window height will always represent 100% of the maximum tag value. If constant values are not entered into the Initial Minimum or Initial Maximum field, i.e., if tags are used instead in order to dynamically specify the minimum or maximum value, the limits for the assigned tags, i.e., minimum or maximum, will be used to determine the 100% level.

Initial Minimum

Initial Maximum

If you select this option, the tags' values will be displayed as absolute values. In other words, the window height and the maximum Y-axis value that can be displayed will be the same as the maximum measured value.

Initial Minimum/Initial Maximum

These fields will be enabled only if a tag format has been selected.

The Initial Minimum and Initial Maximum fields can be used to define the value range for the scale in the axis directions. The scale will begin with the value in the Initial Minimum field and end with the value in the Initial Maximum field.
Any value between 0 and 100% can be entered. Moreover, the Initial Minimum and Initial Maximum entered can be completely independent from the limits for the tags being displayed.

Initial Minimum and Initial Maximum can be defined as constant values or or as dynamic values using tags. To set them as constant values, simply enter the numbers you want into the corresponding fields. This also means that it will not be possible to change the settings' value on the device. To set them as dynamic limits instead, enter the tags you want.

Zooming using tags

If a tag is used for the Minimum and Maximum fields, it will be possible for device users to change the area being shown at runtime on the device screen, provided a slider or value display/entry object is set up so that they can change the corresponding values, for example. This will enable operators to zoom in and out of the graph. The entire range will be shown if Minimum is set to 0% and Maximum to 100%.

The range shown will be set according to the percentages given in the Minimum and Maximum tags. This means that the selected tags are only allowed to have values between 0 and 100. If Minimum is less than 0, it will be set to 0%. Likewise, if Maximum is greater than 100, it will be set to 100%. This also means that zoom settings can be modified as necessary by simply changing the corresponding tag values.

Y2 Axis

This section can be used to define a second Y-axis scale for graphs over time.
The "Curves" tab can then be used to define which tag belongs to which of the two Y-axis scales.
Both scales can be used together.

Scale

The Scales sub-section contains various options for the corresponding axis, the lines used to subdivide the axis, and the axis text labels. The scale will be automatically subdivided based on the size of the visualization object and the scale range defined by the minimum and maximum values.

Fixed Distance

If this checkbox is ☑ enabled, the corresponding axis will be subdivided into fixed segments.
For example: If a value of <25> is entered, a scale dash will be placed on the X-axis every 25 measured values.
This means that the Minimum, Maximum, and Fixed distance settings determine how the scale will be subdivided..

If the checkbox is disabled, the scale will be subdivided automatically based on the size of the visualization object and the scale range.

The maximum number of measured values per scale dash is limited to half of all the measured values in the corresponding graph block (please refer to → "No. of measured values", page 681 as well).

→ If the distance between two scale dashes is shorter than 20 pixels, the executable program will only show every other scale dash on the device screen.

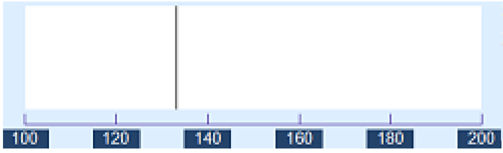


Figure 466: Graph without Fixed Distance

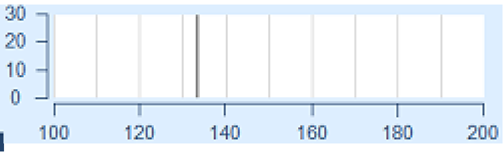


Figure 467: Graph Block with Fixed Distance = 25

Lines

Line Color

The color selected here will be used for the corresponding axis line and the scale dashes.

You can change the color if you want. The following options are available for this purpose:

- Transparent
- Standard Colors
- More Colors...

The most recently used colors will be shown (user-specific).

For detailed information on the available colors, please refer to the following section:
→ section "Color choice - Object's Color Palette", page 234

Scale lines

If this checkbox is ☑ enabled, scale dashes will be shown along the corresponding axis. These dashes can be shown even if the actual axis is not being displayed.

→ As soon as the distance between two lines is smaller than 20 pixels, only every second line will be shown on the device screen at runtime.

Axis line

If this checkbox is ☑ enabled, an X-axis will be shown.
Even if an X-axis is not being shown, the scale dashes can still be shown.

Inside

If this checkbox is ☑ enabled, the scale dashes will be shown on top of the X-axis; otherwise, they will be below it.

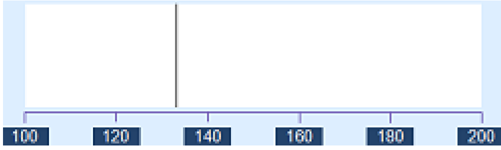


Figure 468: Graph with scale dashes below the axis

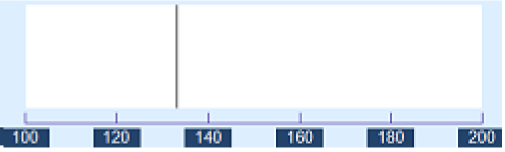


Figure 469: Graph with scale dashes on top of the axis

Texts

→ The options that are available will depend on the selected → "Data source:", page 694 in Extended properties Graph, General section.

Labels

If this checkbox is ☑ enabled, the corresponding axis lines and dash lines will be shown with an axis label.

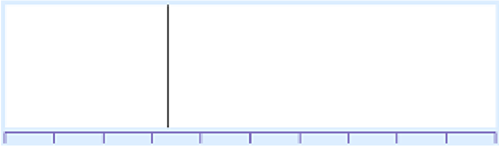


Figure 470: Graph without labels

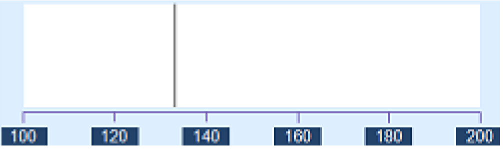
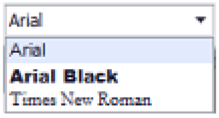


Figure 471: Graph with labels: white font color, dark blue background color

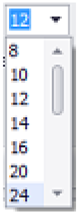
Font

With the exception of the <Arial> default font, the fonts available will be those that have been added in *Home\Languages\Project Languages*.



Font Size

The font size can be left with its default value or can be manually changed to a value within a range of 1 to 999 pt. The line height will be automatically adjusted. In addition, the text in the inserted text box will automatically be broken as required for the object's size if the automatic word wrap option has been enabled.



Text color

You can change the color if you want. The following options are available for this purpose:

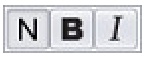
- Transparent
- Standard Colors
- More Colors...

The most recently used colors will be shown (user-specific).

For detailed information on the available colors, please refer to the following section:
→ section "Color choice - Object's Color Palette", page 234

Font style N

The following formatting options are available:



- Default (**N**ormal)
- **Bold** (**B**old)
- *Italic (**I**talic)*

Background:

You can change the color if you want. The following options are available for this purpose:

- Transparent
- Standard Colors
- More Colors...

The most recently used colors will be shown (user-specific).

For detailed information on the available colors, please refer to the following section:
→ section "Color choice - Object's Color Palette", page 234

Tag array as a data source

If you select tag array as the data source, the X-axis label can be displayed in various ways, since there is no time information:

- Incremented value
- Incremented project text ID

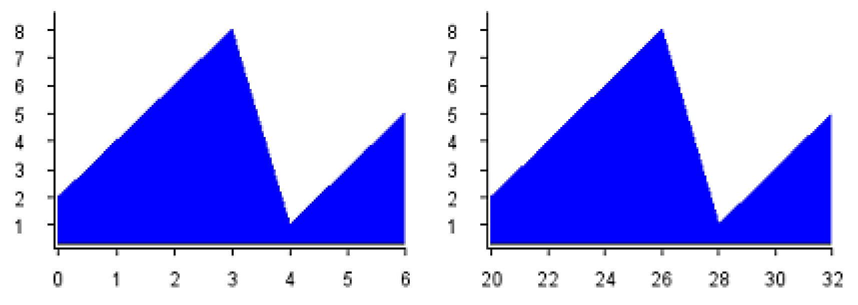
- Tag

This makes it possible to get custom labels for the individual array elements/values.

Incremented value

In this case, the label will be a numeric value and you will need to define the corresponding start value and increment value.

The figure below shows two different configurations – one with a start value of 0 and an increment of 1 and one with a start value of 20 and an increment of 2:

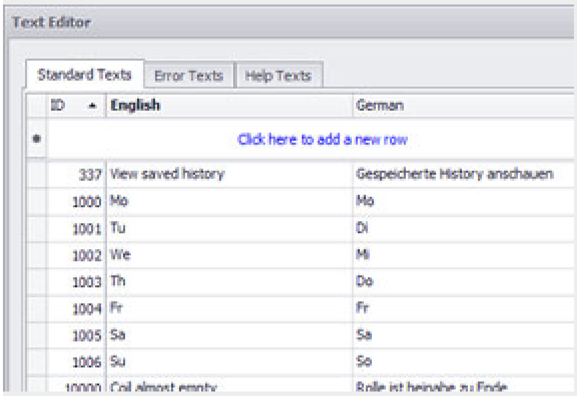


The two graphs are actually showing the exact same array data. Keep in mind that the start value always corresponds to the first element of the assigned tag array, i.e., this is why the label for the first array element is "0" in the first graph and "20" in the second one.

Incremented project text ID

You can use this option to make the labels be based on project texts. In this case, you will need to define a start value corresponding to the first project text ID to be used and an increment value for the project text IDs that should follow.

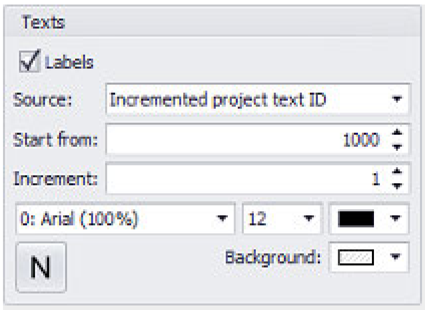
Following is the text configuration used for the example further below:



ID	English	German
337	View saved history	Gespeicherte History anschauen
1000	Mo	Mo
1001	Tu	Di
1002	We	Mi
1003	Th	Do
1004	Fr	Fr
1005	Sa	Sa
1006	Su	So
10000	Coil almost empty	Rolle ist beinahe zu Ende

Text IDs 1000 to 1006 are assigned to weekday abbreviations.

If we use the following settings to configure the graph object:



Texts

☑ Labels

Source: Incremented project text ID

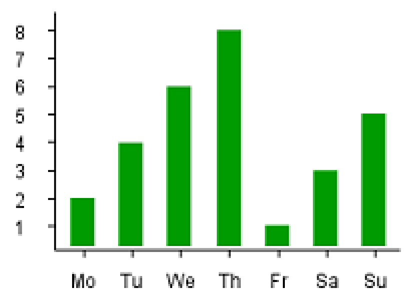
Start from: 1000

Increment: 1

0: Arial (100%) 12

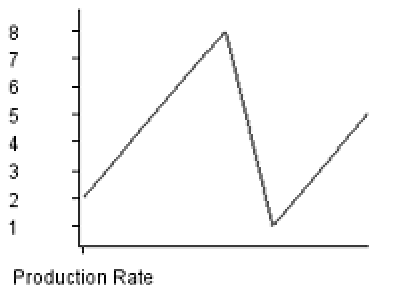
Background:

We get the following axis labels:



Tag

You can use this option to select an individual string tag as the label for the X-axis. In this case, there will not be a separate label for each array element value.



Curves tab

The settings in the Curves tab are used to select tag values being recorded in a graph block so that they can be displayed on the graph visualization object.

The first step is to select the graph block containing the tags that should be displayed. You can then select which specific tags should be displayed. This means that a graph visualization object can only display tag values for a single graph block. Of course, this also means that it is necessary to decide in advance which graph blocks (if several are used) should contain which tags in order to ensure that the project can be set up as needed.

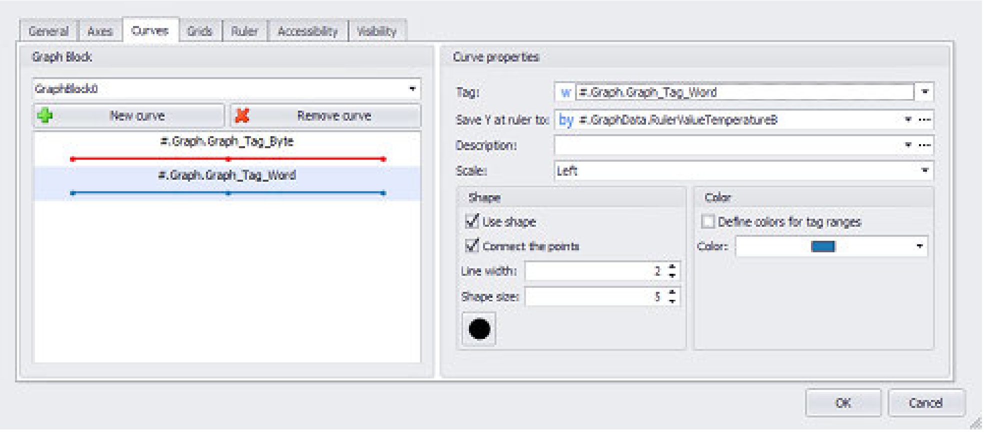
The selection pane will show the curves that will be displayed with the graph visualization object. The curves shown will actually reflect the selected curve properties, meaning that the selection pane serves as a preview pane simultaneously. In addition, the Curve properties section can be used to configure various properties for the curves.

If you selected "Tag array" as a data source, you will be able to select arrays instead of tags. The array does not need to have an identical size, and if there are differences, the largest array will be used to define the number of measuring points (when an array is exceeded, a value of 0 will be shown).

The elements displayed for an array can be reduced. To do this, you can set a static value or a tag that represents the number of elements that should be displayed. If this value is greater than the size of the array, the value will be ignored and the array size will be used instead.

The "Shape" section can be used to configure various options depending on the selected shape.

Controls\ Graph



GraphBlock0

#.Graph.Graph_Tag_Byte

#.Graph.Graph_Tag_Word

Tag: w #.Graph.Graph_Tag_Word

Save Y at ruler to: by #.GraphData.RulerValueTemperatureB

Description:

Scale: Left

☑ Use shape

☑ Connect the points

Line width: 2

Shape size: 5

□ Define colors for tag ranges

Color:

Figure 472: Curves tab in the Extended properties Graph dialog box with three curves in the selection pane

Graph Block

To select the graph block containing the tag values you want to display, click on the drop-down menu arrow. As soon as you select a graph block, all the tags in the selected graph block will be available for selection in the Tag Y field. If the field is empty, you will need to first define and add a graph block. For information on how to add a graph block, please refer to the → section "Add", page 679 section.

New curve

Clicking on the New curve button will add a curve so that the graph visualization object can display the values for an additional tag from the graph block. After the new curve is added, you will still need to select the tag you want for it. Please note that you cannot have more than 255 curves in a single object.

Delete curve

Clicking on the Remove curve button will immediately delete the curve that is selected in the selection pane.

Curve properties

Tag

Clicking on the drop-down menu arrow for the Tag Y field will open a drop-down menu containing all the tags assigned to the selected graph block. When you select a tag, it will be assigned to the curve that is selected in the selection pane and the curve will be given the same name as the tag, e.g., <#.Motor [0]>. At runtime, the tag values will be recorded in the graph block and will also be plotted on the device screen as soon as the device operator switches to the screen containing the graph visualization object.

Save Y at ruler to

The tag specified in this field will be used to store the value of the plotted tag at the marker line's location. The tag specified in this field can be assigned to a value display/entry object in order to implement a discrete display of the measured value at the marker line's position.

Description

Used to assign a text in order to identify the series of graphs shown.
This text will appear on the tap-graph menu in order to allow users to select the series they want to see.

Connect the points

If this checkbox is enabled, the discrete tag values will be connected by a curve.

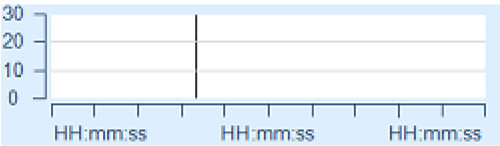


Figure 473: Graph with curve properties: Points not connected

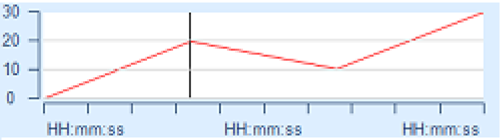


Figure 474: Graph with curve properties: Connected points

Define colors for tag ranges

If this checkbox is enabled, three additional color drop-down menus will appear:

- Upper section
- In Range
- Bottom range

Different colors can be set for each range; otherwise, the curve will be displayed with a single color. The limits for the ranges can be set in the properties for the tags themselves.

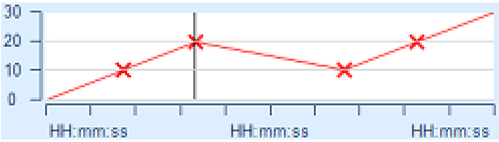


Figure 475: Graph in which points are connected, an X shape is being used, and the selected color is red

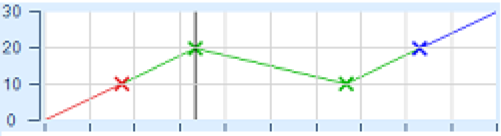


Figure 476: Graph in which points are connected, an X shape is being used, and different colors were set for the different value ranges

Line Width

The line width is defined in pixels, with the permitted range being 1 to 255 pixels.

Color

The Color drop-down menu will only be available if the Define colors for tag ranges checkbox is disabled.

You can change the color if you want. The following options are available for this purpose:

- Transparent
- Standard Colors
- More Colors...

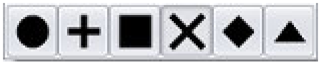
The most recently used colors will be shown (user-specific).

For detailed information on the available colors, please refer to the following section:
→ section "Color choice - Object's Color Palette", page 234

Used Shape

A variety of options can be configured here, with the specific options depending on the selected chart type.

If this checkbox is enabled, you will be able to use the selection list to the right in order to select the shape that you want to use to indicate the discrete tag values:



If the checkbox is disabled, the tag values will be connected by the curve without any special symbols used to indicate the individual measuring points.

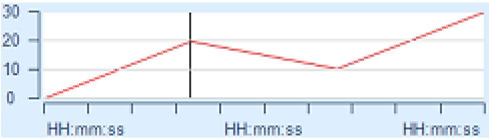


Figure 477: Graph in which the Use shape option is disabled

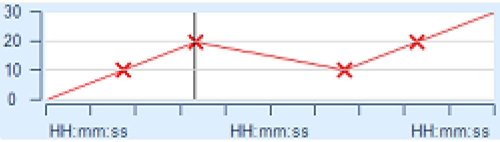


Figure 478: Graph in which the shape selected is X and the shape size is 10

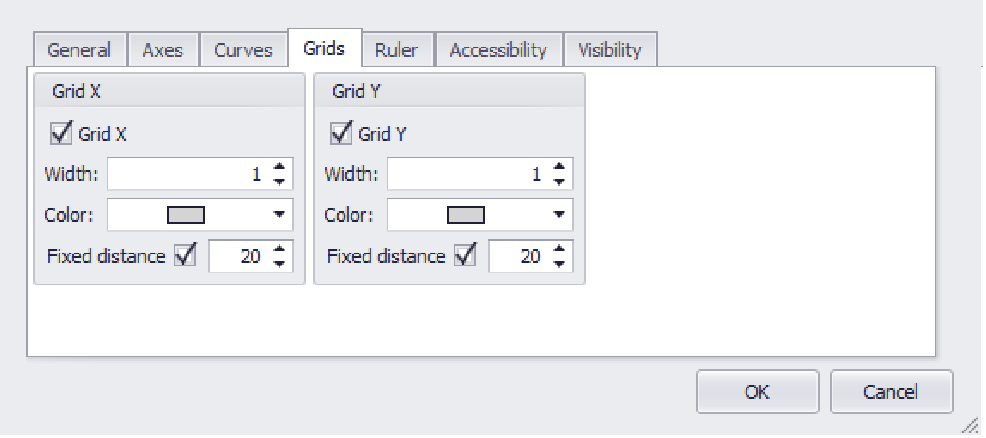
Shape size

The specified shape size will be the number of pixels for the theoretical square containing the symbol used to indicate individual tag values as measuring points. A value between 1 and 16 can be entered here.

Grids tab

You can configure the graph object in such a way as to display a grid. The settings below can be used to configure this grid.

Controls\ Graph\Extended properties tab



General | Axes | Curves | Grids | Ruler | Accessibility | Visibility

Grid X

☑ Grid X

Width: 1

Color:

Fixed distance ☑ 20

Grid Y

☑ Grid Y

Width: 1

Color:

Fixed distance ☑ 20

Figure 479: Extended properties "Graph" dialog box, Grids tab

Device screen with specific Grids settings

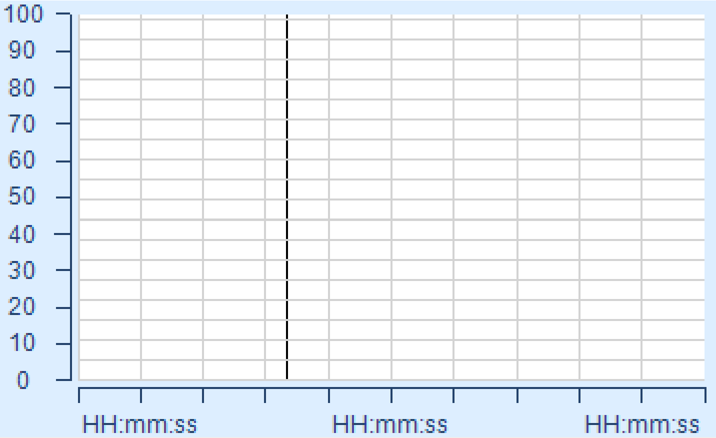


Figure 480: Device screen with graph; the X-axis grid matches the scale dashes for the X-axis, the Y-axis grid is set at a fixed distance of 10 measured values

Grid X

If this checkbox is enabled, the X-axis grid will be drawn with vertical lines that will automatically be drawn at the scale dashes for the X-axis.

Grid

□ Grid	☑ Grid
An X-axis grid will not be shown, i.e., no vertical lines will be drawn.	The X-axis grid will be shown with vertical lines that will automatically be drawn at the scale dashes for the X-axis.

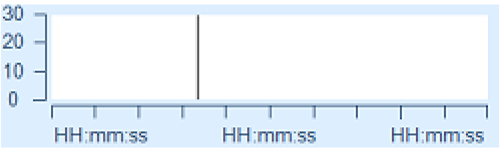


Figure 481: Graph Block without X-Grid

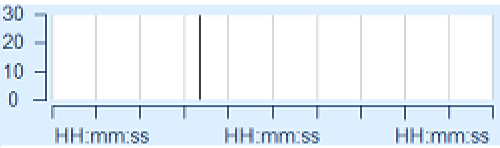


Figure 482: Graph Block with X-Grid

Width

The grid line width is defined in pixels, with the permitted range being 1 to 255 pixels.

Grid line color

You can change the color if you want. The following options are available for this purpose:

- Transparent
- Standard Colors
- More Colors...

The most recently used colors will be shown (user-specific).

For detailed information on the available colors, please refer to the following section: → section "Color choice - Object's Color Palette", page 234

Fixed Distance

If this checkbox is enabled, grid lines will be drawn at a fixed distance using the specified color.

The number of measured values specified in the field to the right is used to define the fixed distance at which grid lines will be drawn. For example: if a value of <25> is entered and the Fixed distance checkbox is enabled, a grid line will be drawn along the X-axis every 25 measured values

The maximum number of measured values per segment is limited to half of all the measured values in the corresponding graph block (please refer to → "No. of measured values", page 681 as well).

→ As soon as the distance between two lines is smaller than 20 pixels, only every second line will be shown on the device screen at runtime.

□ Fixed Distance	☑ Fixed Distance
X-axis grid lines will automatically be drawn at the scale dashes for the X-axis.	Grid lines will be drawn at the specified fixed distance.

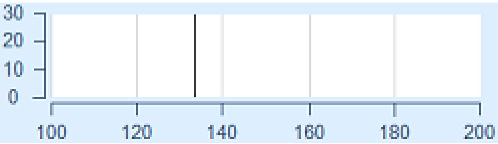


Figure 483: Graph with X-axis grid with the Fixed distance setting disabled

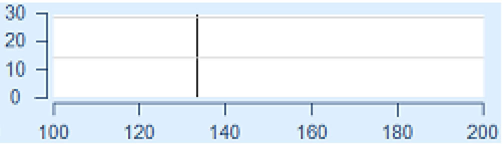


Figure 484: Graph with X-axis grid with a fixed distance = 10

Grid Y

If this checkbox is enabled, the Y-axis grid will be drawn with horizontal lines that will automatically be drawn at the scale dashes for the Y-axis.

□ Grid

A Y-axis grid will not be shown, i.e., no horizontal lines will be drawn.

☑ Grid

If the checkbox is enabled, the Y-axis grid will be drawn with horizontal lines that will automatically be drawn at the scale dashes for the Y-axis.

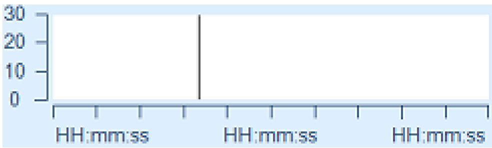


Figure 485: Graph Block without Y-Grid

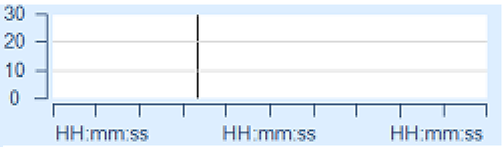


Figure 486: Graph Block with Y-Grid

Width

The grid line width is defined in pixels, with the permitted range being 1 to 255 pixels.

Grid line color

You can change the color if you want. The following options are available for this purpose:

- Transparent
- Standard Colors
- More Colors...

The most recently used colors will be shown (user-specific).

For detailed information on the available colors, please refer to the following section: → section "Color choice - Object's Color Palette", page 234

Fixed Distance

If this checkbox is enabled, grid lines will be drawn at a fixed distance using the specified color.

The number of measured values specified in the field to the right is used to define the fixed distance at which grid lines will be drawn. For example: if a value of <15> is entered and the Fixed distance checkbox is enabled, a grid line will be drawn along the Y-axis every 15 measured values

The maximum number of measured values per segment is limited to half of all the measured values in the corresponding graph block (please refer to → "No. of measured values", page 681 as well).

→ As soon as the distance between two lines is smaller than 20 pixels, only every second line will be shown on the device screen at runtime.

☐ Fixed Distance

Y-axis grid lines will automatically be drawn at the scale dashes for the Y-axis.

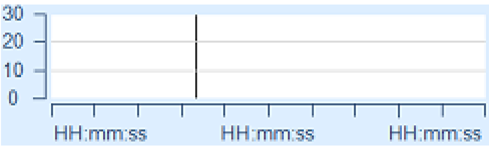


Figure 487: Graph with Y-axis grid with the Fixed distance setting disabled

☑ Fixed Distance

Grid lines will be drawn at the specified fixed distance.

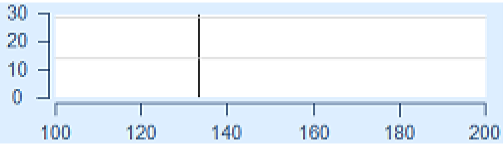


Figure 488: Graph with X-axis grid with a fixed distance = 15

Ruler tag

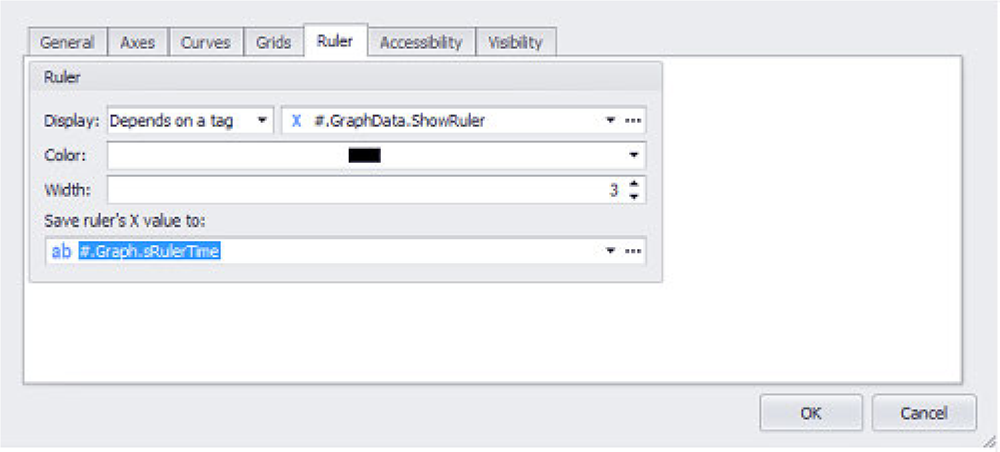
There is exactly one marker line for each graph visualization object.

This marker line can be moved to any plotted tag values in the graph visualization object during plotting. When this is done, the marker line will show the Y-axis and X-axis values for the selected measuring point, provided a tag has been assigned to the Save ruler's time to field.

In addition, the marker line can be used to interpret the measured value data in an archive. The functions in the Graph group can be used to change the marker line's position within archived data. The → "Display Time Graph Ruler", page 723 function will show the time and date for the marker line's current position. Moreover, the values of the individual curves at that position can be saved in bit or numeric tags

These tag values can then be displayed within the screen using → section " Value Display/Entry", page 506 objects, for example.

Controls *Graph*



Ruler

Display: Depends on a tag X #.GraphData.ShowRuler

Color: ■

Width: 3

Save ruler's X value to:

ab #.Graph.sRulerTime

Figure 489: Extended Properties window \ Ruler tab

Show

- Always
  The marker line will always be shown.

- Never
  The marker line will not be used at all. If you select this option, the remaining fields in the section will be disabled.
- Depends on a tag
  When this option is selected, a bit tag can be used to show and hide the marker line. If the bit tag has a value of 0, the marker line will be hidden, while a value of 1 will result in the marker line being shown. This bit tag can be selected by clicking on the drop-down menu arrow on the field to the right of the Display field. The ... pane will open the dialog box for the tag's limits directly so that you can set the bit tag using the "Default" setting if necessary. Please note, however, that this button will only be enabled if you select the Depends on a tag option for the Display setting. In addition, you can set up the application in such a way that device users will be able to change the value of the bit tag with a button visualization object, for example.
  For a description of the button visualization object, please refer to the → section " Button", page 365 section.

Color

Color of the ruler

You can change the color if you want. The following options are available for this purpose:

- Transparent
- Standard Colors
- More Colors...

The most recently used colors will be shown (user-specific).

For detailed information on the available colors, please refer to the following section:
→ section "Color choice - Object's Color Palette", page 234

Width

The width of the marker line is defined in pixels, with the permitted range being 1 to 255 pixels.

Save ruler's X value to

The X-axis value at the marker line position can be stored in the string tag selected here.

7.4 Graph Legend

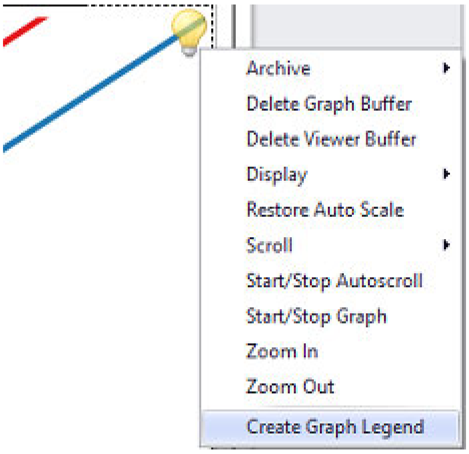
For the visualization conrol Graph an additonal, dynamic curve legend can be created.

Create a Graph Legend

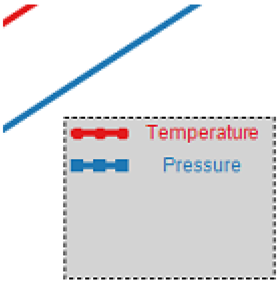
▶ For doing this, the mouse has to be hovered over an existing Graph control.

The 💡 Quick access menu appears in the right upper corner, which offers Graph functions.

▶ From the menu, select Create Graph Legend



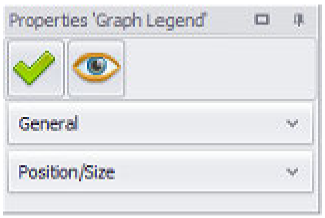
After clicking on it, a "Graph Legend" control gets created and automatically linked to the Graph control.



→ If a curve has no description configured, no text will be displayed, just the curve line preview drawing happens.

The Graph Legend control offers following configuration settings:

Controls\ Graph Legend



'Graph Legend' properties

✓

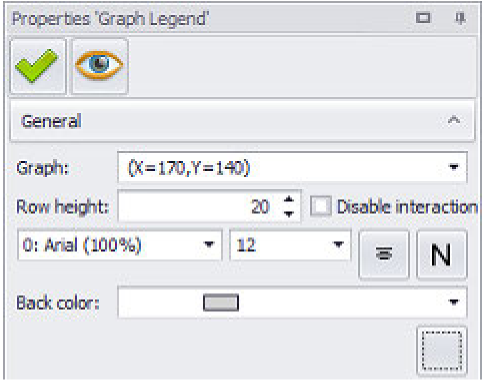
👁

→ section "General", page 715

General

The General section can be used to configure all the basic properties that affect how the visualization object will be displayed, with the exception of its size and position.

Controls\ Graph Legend



Graph: (X=170,Y=140)

Row height: 20 □ Disable interaction

0: Arial (100%) 12

Back color:

Figure 490: Properties Graph Legend pane, General

Graph: The Graph control, for which this Graph Legend is for.

Row Height: The row height for each of the visualized curves

Disable Interaction: ☑ - If this checkbox is ticked, the Graph Legend is always a "read-only" control which only displays information.

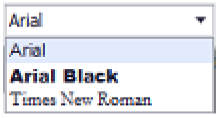
□ - If it is unticked, single curves can be shown/hidden by clicking on the curve.

→ Use the ✓ Accessibility settings in addition, if you would like to prevent the interaction only under certain circumstances.

Font settings for the curve description text (Font, Font Size, Alignment, Style)

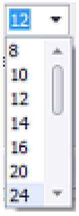
Font

With the exception of the <Arial> default font, the fonts available will be those that have been added in *Home\Languages\Project Languages*.



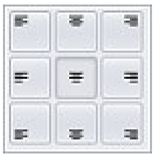
Font Size

The font size can be left with its default value or can be manually changed to a value within a range of 1 to 999 pt. The line height will be automatically adjusted. In addition, the text in the inserted text box will automatically be broken as required for the object's size if the automatic word wrap option has been enabled.



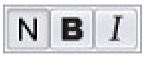
Alignment

The buttons shown below can be used to configure the text to be left-aligned, centered, or right-aligned vertically and horizontally.



Font style N

The following formatting options are available:



- Default (**N**ormal)
- **Bold** (**B**old)
- *Italic (**I**talic)*

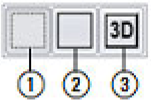
Back color: of the Graph Legend
You can change the color if you want. The following options are available for this purpose:

- Transparent
- Standard Colors
- More Colors...

The most recently used colors will be shown (user-specific). For detailed information on the available colors, please refer to the following section: → section "Color choice - Object's Color Palette", page 234

Frame 3D

The visualization object can be displayed with any one of three different designs.



1	no frame
2	with frame
3	3D frame

7.5 Archive Graph Block data - Archive measured values

This function saves the collected values in a file. The settings for archiving graph data are configured in the graph block tag object; please refer to → section " Graph Blocks - Graph Block Manager", page 678. The values in these files can be loaded if necessary and displayed on the device using a graph visualization object. For a description of the properties for the graph visualization object, please refer to the → section " Graph", page 690 section.

GRSW2 , GRSW3	Archive data can be saved on the flash card, as well as deleted or loaded from the card. This data can also be saved on an external server via Ethernet, for which a loop script can be used as well. This makes it possible to save data at regular intervals. Please refer to the → "Save Archive File", page 725, → "Save Archive File As", page 726, → "Load Archive File", page 724, and → "Delete Archive File", page 722 functions
GPI, WinGRS	The data in the graph archive can be saved both manually and automatically.

GDB- and TDB Files

The recorded tag values are stored in RAM and can be saved continuously from there to a TDB file or as a tag block in a GDB file.

The GDB file will contain the tag values in ASCII format.
This archive file is generated when one of the Save Archive File … functions is used for archiving or when recording is stopped. The Start/Stop Graph 0...99 function is the function used to stop recording. The GDB file can be loaded and displayed by a graph visualization object at any time. Please note that the tag values in RAM will be lost in the event of a power failure, as well as when power is switched off. In this case, only the tag values that were saved before the power went out will be available.

The TDB file will contain the tag values in binary format and serves as a backup copy. This TDB copy will be saved continuously, provided the Save continuously option has been enabled in the properties for the corresponding graph block. This means that in the event of a power failure, as well as when power is switched off, the TDB file will still contain the latest tag values. When the device is switched back on, the TDB file will automatically be converted to a GDB file that can be displayed by the corresponding graph at any time. For a description of the archiving properties, please refer to the→ section "Archive tab", page 684 section.

Graph Block-Archive

GRAPHxy.GDB	Graph Block archive as ASCII file. (See → "GDB File", page 718)
GRAPH.TDB	Graph archive in binary format. This file is only used internally. This file only exists in the compilation.

7.5.1 GDB File

Tag values in graph block tag objects can be saved in GDB files. In these cases, the files will be saved in ASCII format and can be worked on as necessary.

*.GDB files are subdivided into two sections:

- **File Header**

  The file header entries are for Info purposes and are only partly included when an archive file is loaded. The individual columns are separated by Tabs (ASCII character 09). The file header contains the following data:

File	→ "Storage location", page 685
ID	Graph Block tag object properties→ "ID (Object Number)", page 680
Tag Count	No. of measured values → "No. of measured values", page 681
Time span	Time span → "Time span", page 681
Start Time	The time recording started; see → "Events section", page 682 data section
Start Time File	The time for the first measurement; see sample GDB archive file, data section → section "Start Time File:", page 720
Stop Time	The time recording stopped
No. of measured values	The number of measured values in the file. See also:→ "No. of measured values", page 681

- **Data storage area**

  The data section contains the tag values from the graph block list in table format. This data section starts with ID #GDB#.
  This ID will then be followed by the table header with the time/date format in the first column and the relevant tag names in the columns that follow. Finally, the rows that follow will contain all the measurements, with each row containing one measurement for each tag. The first column records the time of the measurement and the remaining columns contain the corresponding tag values at the time specified. The individual columns are separated by Tabs (ASCII character 09).
  #GDB#
  DD.MM.YYYY hh:mm:ss Tag 1 Tag 2 ... Tag n

Sample GDB archive file

```
File:                        DATA\archiv.gdb
Graph Block No.:             2
Tag Count:                   5
Time Span in Sec.:           4
Start Time:                  2015-02-07 15:03:11
Start Time File:             2015-02-07 15:03:11
Stop Time:                   2015-02-07 15:03:31
Sample Count:                6

#GDB#

DD.MM.YYYY hh:mm:ss          Tag 1   Tag 2   Tag 3   Tag 4   Tag 5
2015-02-07 15:03:11          15      247     -762    34572   9753
2015-02-07 15:03:15          31      63      -762    29631   6452
2015-02-07 15:03:19          37      24      -747    32974   7502
2015-02-07 15:03:23          61      13      -687    32974   8123
2015-02-07 15:03:27          81      6       -635    30561   8339
2015-02-07 15:03:31          93      -5      -598    27463   8576
```

7.6 Graph group functions

The Graph group contains all the functions having to do with the graph visualization object and the graph block object.

In order to be able to run functions that affect a graph visualization object, you will always need a function key visualization object. You can design the button for this function key according to your specific needs, e.g., by adding a background color or image to it.
The function key visualization object is described in the → section "Using function keys to trigger actions", page 389 section.

Basic function grouping

Object	Function	Functions
Graph Block:AddAdd	Archive	→ section "Delete Archive File", page 722 → section "Load Archive File", page 724 → section "Load Archive File", page 724 → section "Load Archive File Full Path As Var", page 724 → section "Save Archive File", page 725 → section "Save Archive File As", page 726 → section "Save Archive File As Var", page 727 → section "Save Archive File Full Path As Var", page 727
	Starting/stopping recording and plotting	→ section "Start/Stop Graph Block", page 732
Graph Block	Show Recording	→ section "Display Time Graph End", page 723 → section "Display Time Graph Ruler", page 723 → section "Display Time Graph Start", page 723 → section "Scroll Left", page 728 → section "Scroll Left (Ruler)", page 728 → section "Scroll Left 1 (Ruler)", page 728 → section "Scroll Right", page 729 → section "Scroll Right (Ruler)", page 729 → section "Scroll Right 1 (Ruler)", page 730 → section "Scroll to End", page 730 → section "Scroll to End (Ruler)", page 731 → section "Scroll to Start", page 731 → section "Scroll to Start (Ruler)", page 731 → section "Start/Stop Autoscroll", page 732 → section "Zoom In", page 734 → section "Zoom Out", page 735
Graph viewer	Displaying historical data	→ section "Delete Graph Buffer", page 722 → section "Delete Viewer Buffer", page 722 → section "Delete Archive File", page 722 → section "View Archive File", page 733 → section "View Archive File Var", page 733 → section "View Archive File Full Path As Var", page

		734

7.6.1 Delete Graph Buffer

Clears all recorded values from the graph block object's buffer.

Number of the Function	3092 (C14h)
Executable from the PLC	Yes
Function with ON/OFF state	No
Parameter 1	Type name of Graph Block (See also:Defining graph blocks)
Function available in standard screens	Yes
Function available with keyboards	No
Function available in script	Yes
Function available in application entry script	Yes

7.6.2 Delete Viewer Buffer

Clears the display in the graph visualization object that is currently displaying the graph block object.

Number of the Function	3098 (C1Ah)
Executable from the PLC	Yes
Function with ON/OFF state	No
Parameter 1	Type name of Graph Block
Function available in standard screens	Yes
Function available with keyboards	No
Function available in script	Yes
Function available in application entry script	Yes

7.6.3 Delete Archive File

This function can be used to delete archive files (GDB files). Meanwhile, the → "Save Archive File", page 725, → "Save Archive File As", page 726, and → "Load Archive File", page 724 functions can be used to save and load graph block objects. When this function is run, a list with all the archive files for the relevant graph block will appear.

→ Not all device types support this function

→ Please note that this function can only be used when plotting is stopped (please refer to → "Start/Stop Graph Block", page 732 as well).

Number of the Function	3091 (C13h)
Executable from the PLC	No
Function with ON/OFF state	No
Parameter 1	Type name of Graph Block
Function available in standard screens	Yes
Function available with keyboards	No

Function available in script	Yes
Function available in application entry script	Yes

7.6.4 Display Time Graph End

This function shows the time for the end position of the area currently being shown in the graph visualization object using the selected time/date format.

→ The function key and graph visualization objects must be found on the same screen.

Number of the Function	3073 (C01h)
Executable from the PLC	No
Function with ON/OFF state	No
Parameter 1	Number of graph visualization object (0 X/Y coordinates)
Parameter 2	Date/time format
Function available in standard screens	Yes
Function available with keyboards	No
Function available in script	No
Function available in application entry script	No

→ section "Date Properties/Time Display", page 372

7.6.5 Display Time Graph Ruler

This function shows the time and date of the current ruler position in the selected time/date format.

→ The function key and graph visualization objects must be found on the same screen.

Number of the Function	3074 (C02h)
Executable from the PLC	No
Function with ON/OFF state	No
Parameter 1	Number of graph visualization object (0 X/Y coordinates)
Parameter 2	Date/time format
Function available in standard screens	Yes
Function available with keyboards	No
Function available in script	No
Function available in application entry script	No

→ section "Date Properties/Time Display", page 372

7.6.6 Display Time Graph Start

This function shows the time for the starting position of the area currently being shown in the graph visualization object using the selected time/date format.

➔ The function key and graph visualization objects must be found on the same screen.

Number of the Function	3072 (C00h)
Executable from the PLC	No
Function with ON/OFF state	No
Parameter 1	Number of graph visualization object (0 X/Y coordinates)
Parameter 2	Date/time format
Function available in standard screens	Yes
Function available with keyboards	No
Function available in script	No
Function available in application entry script	No

→ section "Date Properties/Time Display", page 372

7.6.7 Load Archive File

This function can be used to load archive files (GDB files). Meanwhile, the Save Archive File and Save Archive File As functions can be used to save graph block objects. When this function is run, a list with all the existing archive files for the relevant graph block object will appear.

➔ Not all device types support this function

➔ Please note that current tag values will be overwritten when an archive file is loaded.
Please note that this function can only be used when plotting is stopped (please refer to → "Start/Stop Graph Block", page 732 as well).

Number of the Function	3084 (C0Ch)
Executable from the PLC	No
Function with ON/OFF state	No
Parameter 1	Type name of Graph Block
Function available in standard screens	Yes
Function available with keyboards	No
Function available in script	No
Function available in application entry script	No

7.6.8 Load Archive File Full Path As Var

This function can be used to load the tag values in an archive file (GDB file) to a selected graph block object.

➔ Important:
The parameter is entered as a string tag with a filename. This filename must be provided without a path and without an extension.

→ Please note that current tag values will be overwritten when an archive file is loaded.
Please note that this function can only be used when plotting is stopped (please refer to → "Start/Stop Graph Block", page 732 as well).

The data can be reloaded using the FunctionLoad Archive File.

Number of the Function	3102 (C1Eh)
Executable from the PLC	No
Function with ON/OFF state	No
Parameter 1	Type name of Graph Block
Parameter 2	String tag containing the file name
Function available in standard screens	Yes
Function available with keyboards	No
Function available in script	Yes
Function available in application entry script	Yes

7.6.9 Load Archive File Var

This function can be used to load the tag values in an archive file (GDB file) to a selected graph block object.

→ Important:
The parameter is entered as a string tag with a filename. This filename must be provided without a path and without an extension.

→ Please note that current tag values will be overwritten when an archive file is loaded.
Please note that this function can only be used when plotting is stopped (please refer to → "Start/Stop Graph Block", page 732 as well).

Number of the Function	3094 (C16h)
Executable from the PLC	No
Function with ON/OFF state	No
Parameter 1	Type name of Graph Block
Parameter 2	String tag containing the file name
Function available in standard screens	Yes
Function available with keyboards	No
Function available in script	Yes
Function available in application entry script	Yes

7.6.10 Save Archive File

This function can be used to save the current tag values for the selected graph block object to an archive file (GDB file). The corresponding directory and archive name can be specified using the *Directory< and >Archive name* fields in Home\Graph Blocks\graph block\Archive tab.

When the function is run, an archive file will be generated with the specified archive name in the specified directory. If more than one archive file is being used (), a number will be appended to the archive name. This number will be incremented by 1 every time the archive is saved. Once the specified number of archives is reached, the first archive file will be overwritten and the count will start all over. The number of archives can be set in the Archive tab too.

Saved data can be loaded using the Load Archive File function and displayed with a graph visualization object.

→ Not all device types support this function

→ If this function is triggered by a script, the target directory can only be defined on one network drive.

Number of the Function	3093 (C15h)
Executable from the PLC	Yes
Function with ON/OFF state	No
Parameter 1	Type name of Graph Block
Function available in standard screens	Yes
Function available with keyboards	No
Function available in script	Yes
Function available in application entry script	Yes

7.6.11 Save Archive File As

This function can be used to save the current tag values for the selected graph block object to an archive file (GDB file). The corresponding directory can be specified using the *Directory< field in >Home\Graph Blocks*\graph block\Archive tab.

Starting this function will call up the alphanumeric Keyboard for entering a Filename.

The data can be reloaded using the function Load Archive File.

→ Not all device types support this function

Number of the Function	3085 (C0Dh)
Executable from the PLC	No
Function with ON/OFF state	No
Parameter 1	Type name of Graph Block
Function available in standard screens	Yes
Function available with keyboards	No
Function available in script	No
Function available in application entry script	No

7.6.12 Save Archive File As Var

This function can be used to save the current tag values for the selected graph block object to an archive file (GDB file). The corresponding directory and archive name can be specified using the *Directory< and >Archive name* fields in Home\Graph Blocks\graph block\Archive tab.

Important:
The parameter is entered as a string tag with a filename. This filename must be provided without a path and without an extension.

The data can be reloaded using the function Load Archive File.

Number of the Function	3095 (C17h)
Executable from the PLC	No
Function with ON/OFF state	No
Parameter 1	Type name of Graph Block
Parameter 2	String tag containing the file name
Function available in standard screens	Yes
Function available with keyboards	No
Function available in script	Yes
Function available in application entry script	Yes

7.6.13 Save Archive File Full Path As Var

This function can be used to save the current tag values for the selected graph block object to an archive file (GDB file). Both the directory and the filename are passed as a second parameter, and accordingly are completely independent from the graph block.

Important:
The parameter is entered as a string tag with a filename. This filename must be provided without a path and without an extension.

The data can be reloaded using the function Load Archive File.

Number of the Function	3101 (C1Dh)
Executable from the PLC	No
Function with ON/OFF state	No
Parameter 1	Type name of Graph Block
Parameter 2	String tag containing the file name
Function available in standard screens	Yes
Function available with keyboards	No
Function available in script	Yes
Function available in application entry script	Yes

7.6.14 Scroll Left

This function scrolls to the left so that the area of the curve currently being shown is moved accordingly.

This area will move by the percentage specified in parameter 2,
with a value of 100% corresponding to the entire window width of the graph visualization object.

→ The function key and graph visualization objects must be found on the same screen.

Number of the Function	3078 (C06h)
Executable from the PLC	No
Function with ON/OFF state	No
Parameter 1	Number of graph visualization object 0 X/Y coordinates)
Parameter 2	Percentage relative to the area being displayed
Function available in standard screens	Yes
Function available with keyboards	No
Function available in script	No
Function available in application entry script	No

7.6.15 Scroll Left (Ruler)

This function moves the marker line to the left by the percentage specified in parameter 2,
with a value of 100% corresponding to the entire window width of the graph visualization object.

This function scrolls the ruler by one measured value to the left. The visible range is moved as required so that the ruler is constantly visible.

→ The function key and graph visualization objects must be found on the same screen.

Number of the Function	3075 (C03h)
Executable from the PLC	No
Function with ON/OFF state	No
Parameter 1	Number of graph visualization object 0 X/Y coordinates)
Parameter 2	Percentage relative to the area being displayed
Function available in standard screens	Yes
Function available with keyboards	No
Function available in script	No
Function available in application entry script	No

7.6.16 Scroll Left 1 (Ruler)

This function scrolls the ruler by one measured value to the left. The visible range is moved as required so that the ruler is constantly visible.

The visible range is moved as required so that the ruler is constantly visible.

→ The function key and graph visualization objects must be found on the same screen.

Number of the Function	3080 (C08h)
Executable from the PLC	No
Function with ON/OFF state	No
Parameter 1	Number of graph visualization object 0 X/Y coordinates)
Function available in standard screens	Yes
Function available with keyboards	No
Function available in script	No
Function available in application entry script	No

7.6.17 Scroll Right

This function scrolls to the right so that the area of the curve currently being shown is moved accordingly.

This area will move by the percentage specified in parameter 2,
with a value of 100% corresponding to the entire window width of the graph visualization object.

→ The function key and graph visualization objects must be found on the same screen.

Number of the Function	3079 (C07h)
Executable from the PLC	No
Function with ON/OFF state	No
Parameter 1	Number of graph visualization object 0 X/Y coordinates)
Parameter 2	Percentage relative to the area being displayed
Function available in standard screens	Yes
Function available with keyboards	No
Function available in script	No
Function available in application entry script	No

7.6.18 Scroll Right (Ruler)

This function moves the marker line to the right by the percentage specified in parameter 2, with a value of 100%
corresponding to the entire window width of the graph visualization object.

The visible range is moved as required so that the ruler is constantly visible.

→ The function key and graph visualization objects must be found on the same screen.

Number of the Function	3076 (C04h)
Executable from the PLC	No

Function with ON/OFF state	No
Parameter1 Parameter2	1: Graph visualization object (0 X/Y coordinates) 2: Percentage relative to the area being displayed
Function available in Standard screens	Yes
Function available with keyboards	No
Function available in script	No
Function available in application entry script	No

7.6.19 Scroll Right 1 (Ruler)

This function scrolls the ruler by one measured value to the right. The visible range is moved as required so that the ruler is constantly visible.

This function scrolls the ruler by one measured value to the right. The visible range is moved as required so that the ruler is constantly visible.

→ The function key and graph visualization objects must be found on the same screen.

Number of the Function	3081 (C09h)
Executable from the PLC	No
Function with ON/OFF state	No
Parameter 1	Number of graph visualization object 0 X/Y coordinates)
Function available in standard screens	Yes
Function available with keyboards	No
Function available in script	No
Function available in application entry script	No

7.6.20 Scroll to End

This function scrolls the currently visible range of the graph to the end of the measured values, i.e. to the most recent measured value.

→ The function key and graph visualization objects must be found on the same screen.

Number of the Function	3082 (C0Ah)
Executable from the PLC	No
Function with ON/OFF state	No
Parameter 1	Number of graph visualization object 0 X/Y coordinates)
Function available in standard screens	Yes
Function available with keyboards	No
Function available in script	No
Function available in application entry script	No

7.6.21 Scroll to End (Ruler)

This function scrolls the ruler to the end of the measured values, i.e. to the most recent measured value. The visible range is moved as required so that the ruler is constantly visible.

This function scrolls the ruler to the end of the measured values, i.e. to the most recent measured value. The visible range is moved as required so that the ruler is constantly visible.

→ The function key and graph visualization objects must be found on the same screen.

Number of the Function	3088 (C10h)
Executable from the PLC	No
Function with ON/OFF state	No
Parameter 1	Number of graph visualization object 0 X/Y coordinates)
Function available in standard screens	Yes
Function available with keyboards	No
Function available in script	No
Function available in application entry script	No

7.6.22 Scroll to Start

This function scrolls the currently visible range of the graph to the start of the measured values, i.e. to the oldest measured value.

→ The function key and graph visualization objects must be found on the same screen.

Number of the Function	3083 (C0Bh)
Executable from the PLC	No
Function with ON/OFF state	No
Parameter 1	Number of graph visualization object 0 X/Y coordinates)
Function available in standard screens	Yes
Function available with keyboards	No
Function available in script	No
Function available in application entry script	No

7.6.23 Scroll to Start (Ruler)

This function scrolls the ruler to the start of the measured values, i.e. to the oldest measured value. The visible range is moved as required so that the ruler is constantly visible.

This function scrolls the ruler to the start of the measured values, i.e. to the oldest measured value. The visible range is moved as required so that the ruler is constantly visible.

→ The function key and graph visualization objects must be found on the same screen.

Number of the Function	3089 (C11h)
Executable from the PLC	No
Function with ON/OFF state	No
Parameter 1	Number of graph visualization object 0 X/Y coordinates)
Function available in standard screens	Yes
Function available with keyboards	No
Function available in script	No
Function available in application entry script	No

7.6.24 Start/Stop Autoscroll

By default, the graph area being shown will scroll automatically, i.e., the graph will be refreshed as specified in the Automatic scrolling field in *Controls\Graph\Extended properties\General tab*.

If this function is deactivated, the graph will not autoscroll.

The function will remain disabled until either it function is turned back on or the values being displayed are so old that they are no longer found in the data buffer.

The data buffer is designed as a circular buffer. This data buffer is designed as a circular buffer, and its size will depend on the number of samples as defined in *Home\Graph Blocks\<graph block>\General tab*.

Current measured values are automatically refreshed after a Screen Change.

→ The function key and graph visualization objects must be found on the same screen.

Number of the Function	3090 (C12h)
Executable from the PLC	No
Function with ON/OFF state	Yes
Parameter 1	Number of graph visualization object 0 X/Y coordinates)
Function available in standard screens	Yes
Function available with keyboards	No
Function available in script	No
Function available in application entry script	No

7.6.25 Start/Stop Graph Block

These functions can be used to start and stop recording and plotting for all the tags in a graph block object by selecting the matching name. If recording is triggered by the PLC, it will be necessary to enter the ID instead of the name. If recording is ongoing when the device being used is switched off, it will start again automatically when the device is switched back on.

Recording for a graph block object can also be started and stopped using the start and stop settings. These settings can be configured in *Home\Graph Blocks\<graph block>\General tab*.

The Graph group also includes functions that can be used to save, load, and scroll through the values in a graph visualization object.

Number of the Function	3099 (0C1B)
Executable from the PLC	Yes
Function with ON/OFF state	Yes
Parameter 1	Type name of Graph Block Or ID
Function available in standard screens	Yes
Function available with keyboards	No
Function available in script	Yes
Function available in application entry script	No

7.6.26 View Archive File

This function can be used to load old archive files (GDB files) without overwriting the current measured values.

To display the corresponding data, the graph block object must be linked to a graph visualization object.

When the function is run, a list with all existing archive files for the graph block object will appear.

→ Not all device types support this function

Number of the Function	3096 (C18h)
Executable from the PLC	No
Function with ON/OFF state	No
Parameter 1	Type name of Graph Block
Function available in standard screens	Yes
Function available with keyboards	No
Function available in script	No
Function available in application entry script	No

7.6.27 View Archive File Var

This function can be used to load old archive files (GDB files) without overwriting the current measured values. To display the corresponding data, the graph block object must be linked to a graph visualization object. When the function is run, a list with all existing archive files for the graph message will appear.

→ Not all device types support this function

Important:
The parameter is entered as a string tag with a filename. This filename must be provided without a path and without an extension.

Number of the Function	3097 (C19h)
Executable from the PLC	No
Function with ON/OFF state	No
Parameter 1	Type name of Graph Block
Parameter 2	String tag containing the file name
Function available in standard screens	Yes
Function available with keyboards	No
Function available in script	No
Function available in application entry script	Yes

7.6.28 View Archive File Full Path As Var

This function can be used to load old archive files (GDB files) without overwriting the current measured values. To display the corresponding data, the graph block object must be linked to a graph visualization object.

Important:
The parameter is entered as a string tag with a filename. This filename must be provided without a path and without an extension.

Number of the Function	3103 (C1Fh)
Executable from the PLC	No
Function with ON/OFF state	No
Parameter 1	Type name of Graph Block
Parameter 2	String tag containing the complete file path and the filenames.
Function available in standard screens	Yes
Function available with keyboards	No
Function available in script	Yes
Function available in application entry script	Yes

7.6.29 Zoom In

This function can be used to zoom in to the area of the graph visualization object that is being shown by the X-axis zooming factor specified in the Parameter 2 field, i.e., to stretch the area being shown. The maximum permitted value for this factor is 255.

The function key and graph visualization objects must be found on the same screen.

The Zoom Out function can be used to zoom out.

Number of the Function	3086 (C0Eh)
Executable from the PLC	No

Function with ON/OFF state	No
Parameter 1	Type name of Graph Block
Parameter 2	Factor
Function available in standard screens	Yes
Function available with keyboards	No
Function available in script	No
Function available in application entry script	No

→ "Ruler tag", page 712

7.6.30 Zoom Out

This function can be used to zoom out of the area of the graph visualization object that is being shown by the X-axis zooming factor specified in the Parameter 2 field, i.e., to shrink the area being shown. The maximum permitted value for this factor is 255.

The Graph Block can be enlarged using the Zoom In function.

The function key and graph visualization objects must be found on the same screen.

Number of the Function	3087 (C0Fh)
Executable from the PLC	No
Function with ON/OFF state	No
Parameter 1	Type name of Graph Block
Parameter 2	Factor
Function available in standard screens	Yes
Function available with keyboards	No
Function available in script	No
Function available in application entry script	No

→ "Ruler tag", page 712

8. Recipes

Normally, the word recipe" is used to refer to a combination of ingredients with quantities, temperatures, and times that is used to make a product – especially a food dish. In manufacturing, a recipe usually refers to a specific product model or a specific method. In GALILEO, a recipe is a way to group various tags into a tag set.

The tags in a tag set are filled with values, resulting in one or more value sets. In turn, each value set counts as one recipe entry. The following analogy should illustrate the concept better: A "recipe" for making bread would be specifying that a combination of flour, water, yeast, and oven temperature is required. Meanwhile, a "recipe entry" would contain the actual values for a specific bread, e.g., one recipe entry would specify 100 grams of flour, 0.5 liters of water, 20 grams of yeast, and an oven temperature of 180 °C for a specific type of bread, another one would specify the same ingredients with an oven temperature of 220 °C for a different type of bread, and so on.

Multiple recipe entries are grouped into a single recipe tag object. This recipe tag object is not a visualization object, but is instead an object used to create and manage the various value sets.

Each Recipe tag object, together with all the values for the various corresponding recipe entries, can be saved on the relevant device in a <RecipeObjectName.TAB> file. (this file will hereafter be referred to as a Recipe File). Recipe tag objects can be loaded from recipe files and displayed, as well as edited. The name of a recipe file will be the same as the name for the corresponding recipe tag object.

 Recipe:Visualization objects visualization objects are used to visually represent recipe tag objects on the device screen at runtime in the form of a selection list of recipe entries. The properties for these visualization objects can be used to change the way in which they are displayed, e.g., in terms of position and size In addition, these objects can be manipulated with touch gestures on certain devices.

Recipes "live" at runtime, meaning that many of the properties in a recipe tag object are not edited until at runtime, at which point it is the corresponding device operator who edits them. For example, the Save Recipe Entry function can be used to enable device operators to edit the name of a recipe entry. In fact, and in contrast to other objects, the recipe tag object only has a few settings that are configured in advance.

Data communications with the PLC are not triggered automatically when using recipes. Instead, data must intentionally be transferred to the PLC when desired.

The tag values in a recipe tag object can be edited at runtime. Edited values will be transferred directly to the PLC at runtime, provided the corresponding tags are assigned to a PLC address.

The Recipes function group features additional functions that can be used to change the way in which recipe entries are sorted when displayed. Triggering these functions always requires the use of a **function key** visualization object.

Recipe limits

The following limits apply to recipes:

Objects Recipe	Max. amount per project	32
Recipe entries	Max. amount per recipe object	9'999
Variables	Max. amount per recipe entry	2'000

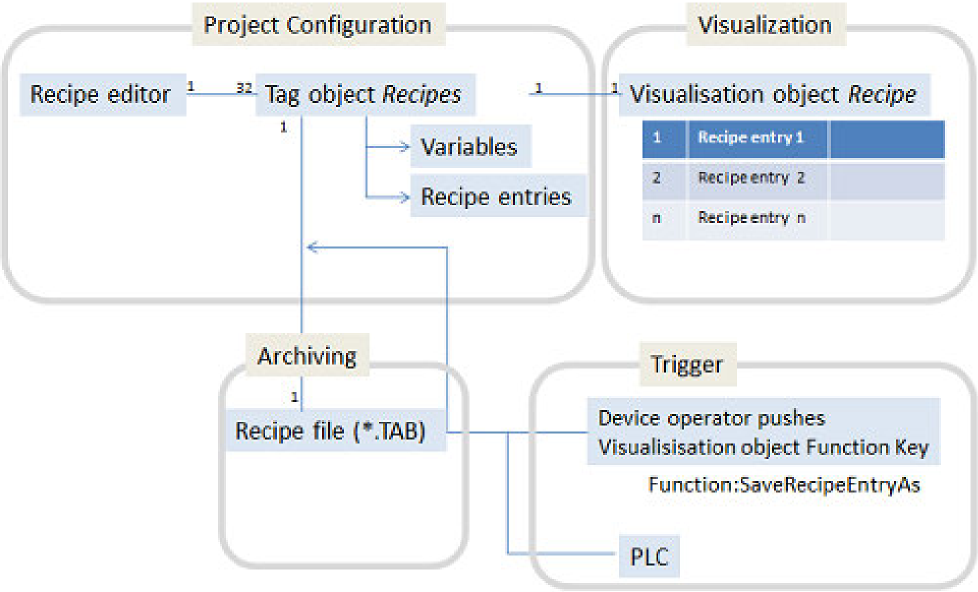


Figure 491: How recipes work: how the recipe tag object and the recipe visualization object interact with each other

8.1 Using recipes to control processes

In real-life applications, recipes are used to make it possible to quickly switch between production processes in production systems. The selection is made by the corresponding device operator on the device screen, and, if necessary, it is also possible to enable the operator to edit parameters for a production process.

The following steps show how to create a recipe tag object and display it on a screen using a recipe visualization object. The use case that follows describes a typical scenario, as well as the results expected on the device screen or computer.

Prerequisites

- A device must already be selected, e.g., XV-102-D6-70TWRC-10
- A communication system must already be selected, e.g., XSOFT-CODESYS-2
- A screen must have already been created, e.g., <Start screen>.
- You must have already imported the tags you want to display and edit from the PLC or created them as PLC tags.

To create a recipe, follow the steps below:

1. **Creating a recipe tag object**

   You will need to create a recipe tag object before you can create a recipe visualization object. To create a recipe tag object, follow the steps below:

   ▶ Open the Recipe Editor by clicking on *Home\Recipes*.

   ▶ You can create a folder structure of your choice for the purpose of organizing your recipe tag objects. To do this, click on Add\Add Folder and rename the folder, e.g., <Cake>.

   ▶ Click the Add\Recipe button.

   ▶ Enter the name you want for the recipe tag object, e.g., <Color Mixture>.

   ▶ Open the General tab.

   ▶ In the No. of entries field, enter the maximum number of recipe entries, e.g., <20>.

   ▶ In the Entry name length field, enter the maximum number of characters for each recipe entry name e.g., <32>. This way, you can limit the number of characters that device operators will be able to enter later on.

   ▶ Enable the No identical names are allowed checkbox in order to force device operators to assign a unique name to each entry.

   ▶ Click on OK.

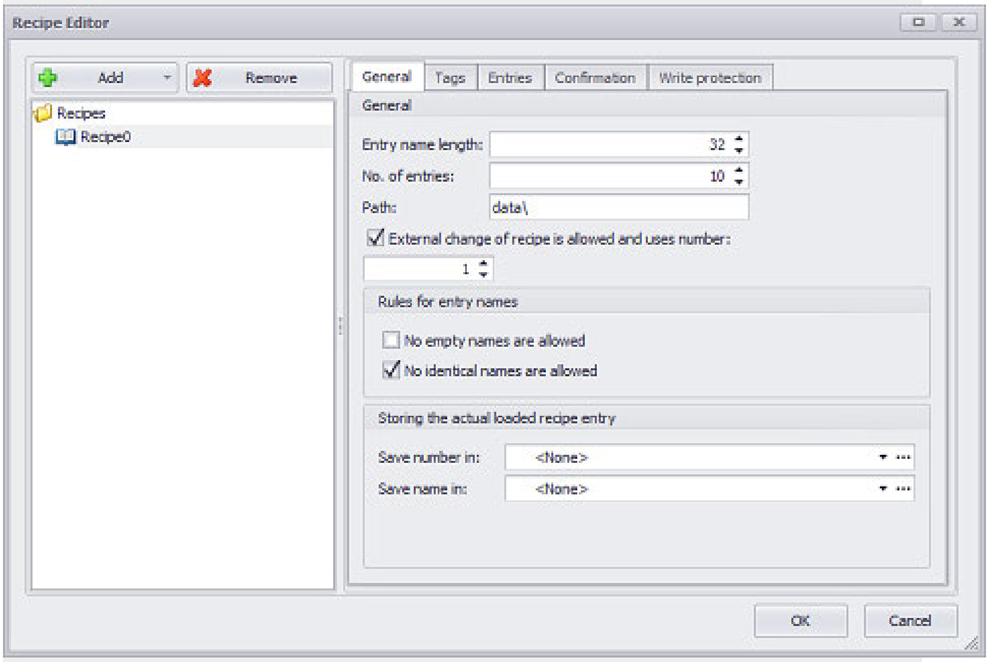
   You have now created a recipe tag object. Please note, however, that nothing will be displayed on the screen yet at this point.

   Example showing how to create a recipe tag object

   ▶ Open screen the Recipe Editor.

   ▶ Add a recipe called <Color Mixture>.

- Switch to Register General.
- In the No. of entries field, enter <20> as the maximum number of recipe entries.



General

Entry name length: 32

No. of entries: 10

Path: data\

☑ External change of recipe is allowed and uses number: 1

Rules for entry names

☐ No empty names are allowed

☑ No identical names are allowed

Storing the actual loaded recipe entry

Save number in: <None>

Save name in: <None>

Figure 492: Recipe Editor dialog box

- Click on OK.

2. **Assigning tags to a recipe**

In order to assign tags to a recipe tag object, you will need to use the Recipe Editor. You can assign imported tags from the PLC or define internal tags in GALILEO and assign them.

Prerequisites

If you want to add tags, make sure to first create them in*Project Tree/Tags* or import them.

To assign tags to a recipe tag object, follow the steps below:

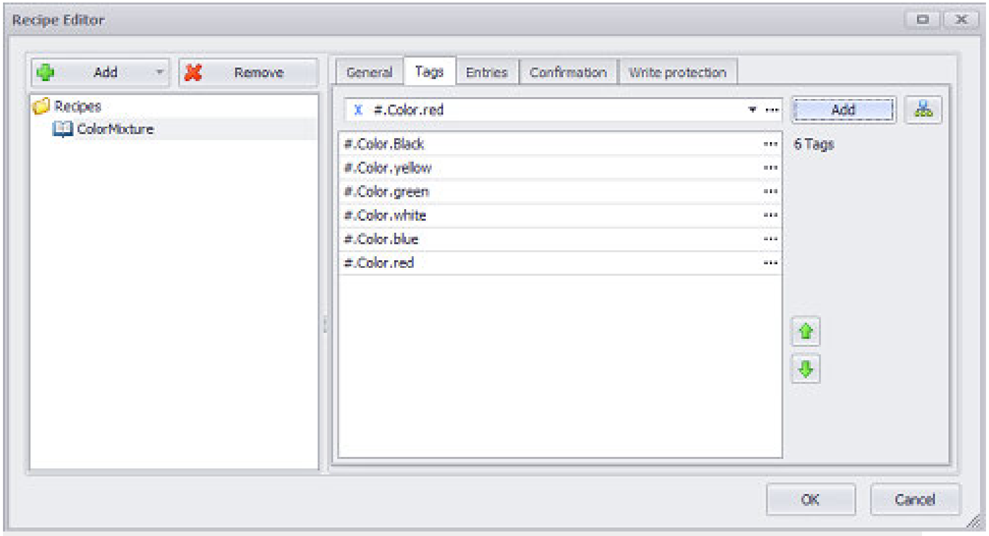
- Open the Recipe Editor by clicking on *Home\Recipes*.
- Select the recipe tag object to which you want to assign tags, e.g., <Color Mixture>
- Open the Tags tab.
- Click on the drop-down menu arrow for the Tag field to select the tag you want, then click on Add.

The tag will be added to the recipe tag object and will be shown in the list of tags.

- ▶ Repeat the last step as many times as necessary until you have added all the tags you want to the recipe tag object.

You have now assigned tags to a recipe tag object. Please note, however, that nothing will be displayed on the screen yet at this point.

Example showing how to assign tags to a recipe tag object



X #.Color.red

#.Color.Black
#.Color.yellow
#.Color.green
#.Color.white
#.Color.blue
#.Color.red

6 Tags

3. **Creating a recipe visualization object**

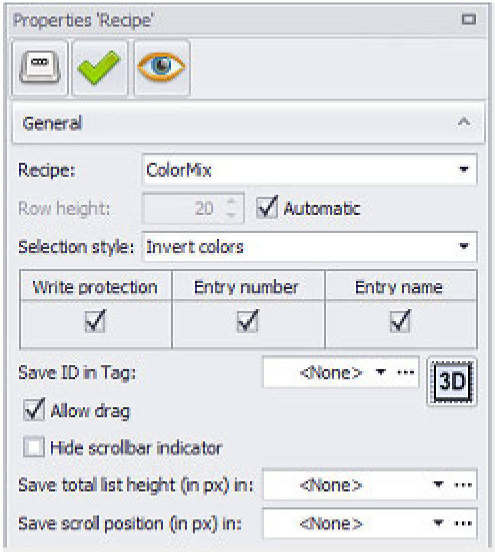
Before creating a recipe visualization object, you should first create the corresponding recipe tag object. Otherwise, there will be no tag sets to visualize.

To create a recipe visualization object, follow the steps below:

- ▶ Click on *Controls\Recipe*.
- ▶ Position the object
  Move the cursor to the screen and draw the area for the visualization object.
- ▶ Make sure that the visualization object is selected on the screen.
- ▶ In the Properties 'Recipe' pane, click on the drop-down menu arrow for the Recipe field to select the recipe you want, e.g., <Color Mixture>.
- ▶ Enable the Write protection, Entry number, and Entry name checkboxes. If none of these checkboxes are enabled, an empty list will be displayed on the screen at runtime.
- ▶ Enable the Allow drag checkbox. This will make it possible to scroll through the recipe entries in the list on the device screen at runtime.
  These touch gestures can only be used with PTC devices such as the XV300 and XP500.
  This function is also supported by GALILEO Open applications.

The recipe visualization object on the screen will automatically be filled out with the recipe entries previously created in the recipe tag object.

Example showing how to create a recipe visualization object



Properties 'Recipe'

General

Recipe: ColorMix

Row height: 20 ☑ Automatic

Selection style: Invert colors

Write protection	Entry number	Entry name
☑	☑	☑

Save ID in Tag: <None>

☑ Allow drag

☐ Hide scrollbar indicator

Save total list height (in px) in: <None>

Save scroll position (in px) in: <None>

Figure 493: **Recipe** properties

4. **Preset Values**

The values for the tags can be preset so that the preset values will be used whenever there is no communication with the PLC. This makes it possible to fill tags with values in sample applications that do not use a PLC.

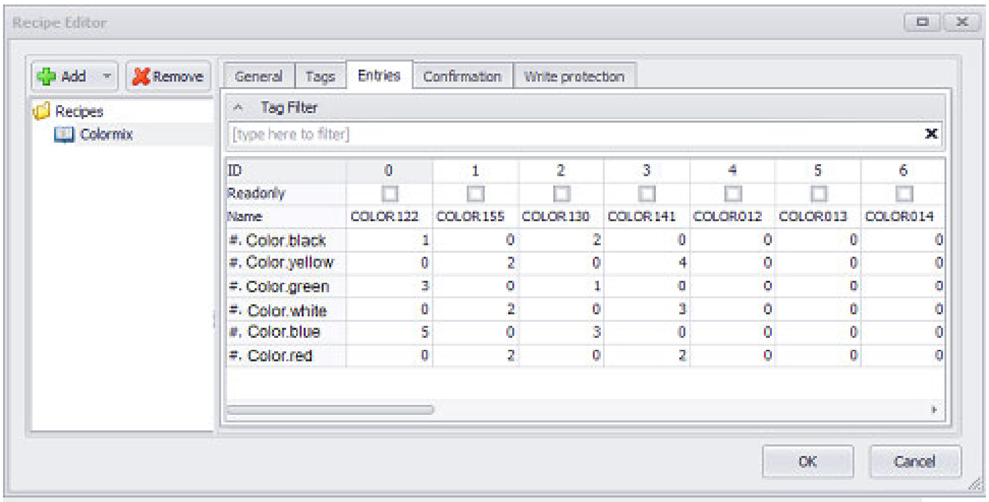
To assign values to the tags in a recipe tag object, follow the steps below:

- Open the Recipe Editor by clicking on *Home\Recipes*.
- Select the recipe tag object to which you want to assign tags, e.g., <Color Mixture>
- Open the Entries tab.
- Each column represents a recipe entry. For each recipe entry, click on the cells for each tag and enter the values you want. In the Name cell, enter the name of the recipe entry, e.g., <COLOR155>.
- Start the Galileo Simulator with function key F5 or by clicking on Home\ . Build.

The tag will be added to the recipe tag object and will be shown in the list of tags.

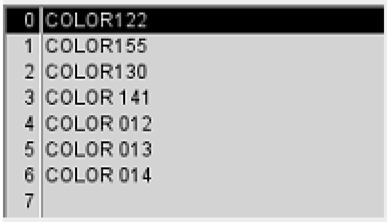
You have now assigned values to the various recipe entries. As soon as the visualization object is displayed on the device screen, the recipe entries' names will be shown, provided the Entry name checkbox has been enabled in the properties for the recipe visualization object.

Example showing how to preset values for tags



ID	0	1	2	3	4	5	6
Readonly	□	□	□	□	□	□	□
Name	COLOR 122	COLOR 155	COLOR 130	COLOR 141	COLOR012	COLOR013	COLOR014
#. Color.black	1	0	2	0	0	0	0
#. Color.yellow	0	2	0	4	0	0	0
#. Color.green	3	0	1	0	0	0	0
#. Color.white	0	2	0	3	0	0	0
#. Color.blue	5	0	3	0	0	0	0
#. Color.red	0	2	0	2	0	0	0

Figure 494: Recipe tag object with recipe entry values from the example



0	COLOR122
1	COLOR155
2	COLOR130
3	COLOR 141
4	COLOR 012
5	COLOR 013
6	COLOR 014
7	

Figure 495: Device screen with recipe visualization object at runtime

5. Displaying and editing the values in a recipe at runtime

Keep in mind that, at runtime, the visualization object will list the recipe entries in a recipe, but not the values. In other words, it is first necessary to select a recipe entry at runtime and then load the corresponding values from the recipe file and show them using another visualization object, such as a value display/entry object. If you want to make it possible for the values to be edited, this will also require a separate visualization object for input purposes.

To make it possible to display the values for a recipe entry at runtime, follow the steps below:

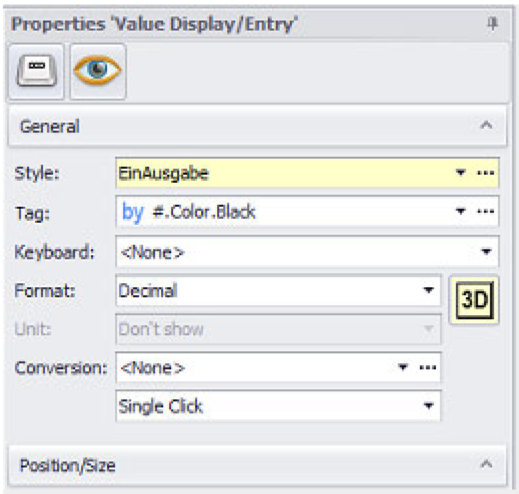
- ▶ Click on *Controls\Value Display/Entry*.
- ▶ Position the object
  Move the cursor to the screen and draw the area for the visualization object.
- ▶ In the Properties 'Value Display/Entry' pane, click on the drop-down menu arrow for the Tag field and select the tag for which you want values to be displayed, e.g., <#.Colors.black>.
- ▶ Click on the drop-down menu arrow for the Keyboard field to select the keyboard that should be used to enter a new value, e.g., <Numeric >. If you do not enter a keyboard, values will be displayed, but it will not be possible to edit them.

- Repeat these steps for all the tags in the recipe entry that should be displayed and edited.
- Start the Inspector by pressing the F5 key.

The values shown in the value display/entry visualization objects will not be the values for the recipe entry selected in the list. The recipe entry will need to be loaded first. Clicking on one of the value display/entry visualization objects will open the defined system keyboard, allowing operators to enter a value. Important: The recipe entry must be saved. The recipe entry selected on the device screen is saved in the *.TAB file.

→ To quickly insert the object, use a function key with a special function on the screen 💡

Example showing how to make it possible to display and edit values at runtime

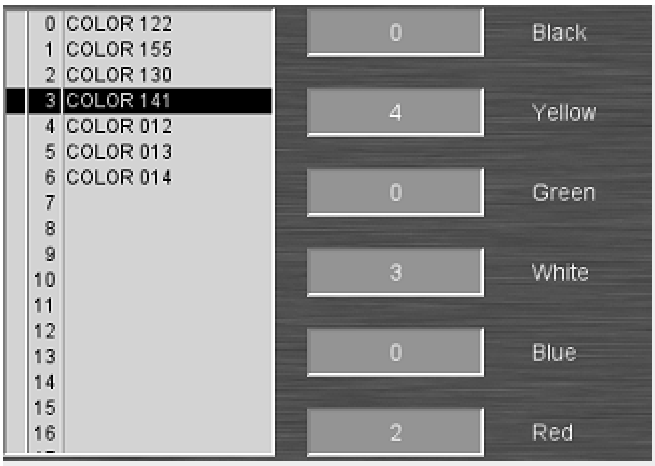

Properties 'Value Display/Entry'

General

Style: EinAusgabe
Tag: by #.Color.Black
Keyboard: <None>
Format: Decimal
Unit: Don't show
Conversion: <None>
Single Click

Position/Size

Figure 496: Properties 'Value Display/Entry' pane



0 COLOR 122
1 COLOR 155
2 COLOR 130
3 COLOR 141
4 COLOR 012
5 COLOR 013
6 COLOR 014
7
8
9
10
11
12
13
14
15
16

0	Black
4	Yellow
0	Green
3	White
0	Blue
2	Red

Device screen section with Color Mixture <recipe> visualization object and six value display/entry visualization objects for displaying and editing the values for six different tags.

6. **Save Recipe (optional)**

Say you want to allow device operators to save the recipe entry that is currently selected on the device screen at runtime by tapping a button.

Prerequisites
You must have already created a recipe tag object and a recipe visualization object. In addition, both objects must already be linked to each other.

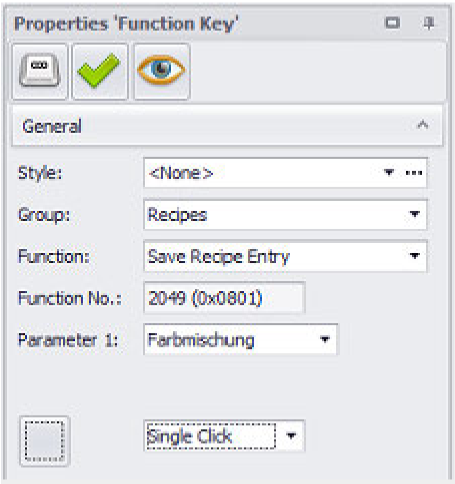
To make it possible for device operators to save a recipe entry selected on the device screen, follow the steps below:

- ▶ Click on *Controls\Function Key*.
- ▶ Position the object
  Move the cursor to the screen and draw the area for the visualization object.
- ▶ In the Properties 'Function Key' pane, click on the drop-down menu arrow for the Group field and select the <Recipes> option.
- ▶ Click on the drop-down menu arrow for the Function field and select the <Save Recipe Entry> function.
- ▶ Click on the drop-down menu arrow for the Parameter 1 field and select the recipe tag object that will be saving the entry, e.g., <Color Mixture> .
- ▶ Start the Inspector by pressing the F5 key.

When the button for the function key is tapped on, the recipe entry selected on the device screen will be saved in the TAB file.

→ To quickly insert the object, use a function key with a special function on the screen

Example showing how to add a function for saving a recipe entry in a recipe tag object



Properties 'Function Key'

General

Style: <None>

Group: Recipes

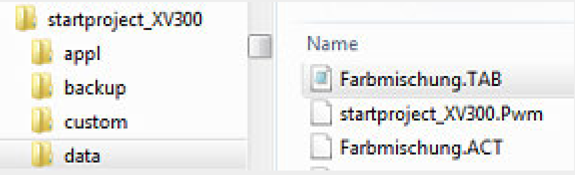
Function: Save Recipe Entry

Function No.: 2049 (0x0801)

Parameter 1: Farbmischung

Single Click

Figure 497: Properties 'Function Key' pane



startproject_XV300
appl
backup
custom
data

Name
Farbmischung.TAB
startproject_XV300.Pwm
Farbmischung.ACT

Figure 498: Color Mixture <recipe> tag object saved as <Color Mixture.TAB> file in the <data> filepath, which can be set in *Home\Recipes\General tab*.

7. **Enabling write protection for a recipe tag object**

   If the write protection of a recipe entry is activated, this means that:

   - Name changes are not possible
   - The name of the recipe entry and the corresponding tag values will not be overwritten when any of the following functions are run:
     → "Clear Recipe Entry", page 767
     → "Paste Recipe Entry", page 777
     → "Restore Default Values", page 778
     → "Save Recipe Entry", page 782
     → "Save Recipe Entry As", page 783
     → "Save As/Load Recipe Entry", page 782

   To enable write protection for all the recipe entries in a recipe tag object, follow the steps below:

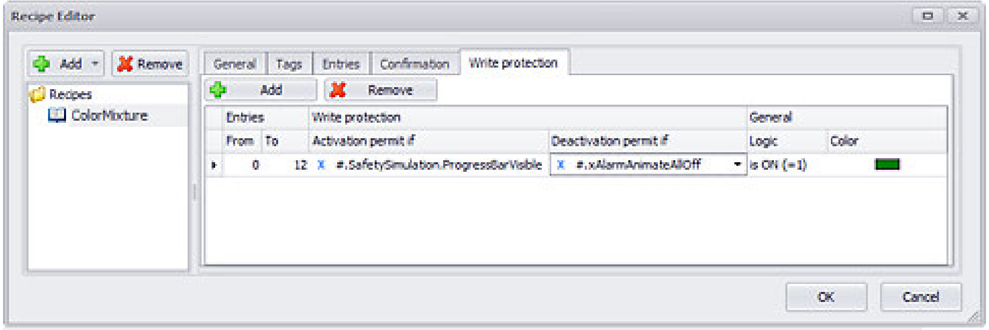
   ▶ Click on *Home\Recipes*.
   ▶ Select the recipe tag object you want from the Recipes folder, e.g., **<Color Mixture>**.
   ▶ Open the *Write protection* tab and click on the Add button.
   ▶ Enter a value of *0* into the <Entries\From> column.

You should have already entered the number of entries in the *No. of entries* field in the General tab. You will be using this number for the following step.

- ▶ Enter a value of *number of entries - 1* into the <Entries\To> column, e.g., <3>.
- ▶ In this particular case, say that you want write protection to be enabled permanently instead of being linked to a condition. Accordingly, do not enter anything into the Write protection subcolumns.
- ▶ Click on the drop-down menu arrow for the Color cell to select a text color for write-protected recipe entries, e.g., a red color.
- ▶ Start the Inspector by pressing the F5 key.

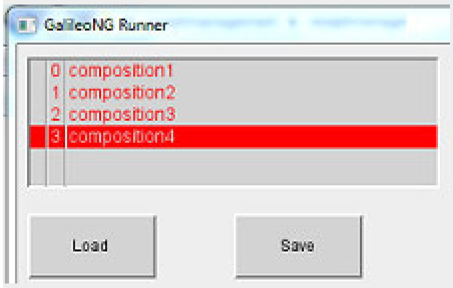
All the recipe entries in the recipe tag object are now write-protected and will be shown with the selected text color on the device screen. In addition, every write-protected recipe entry will be identified by an asterisk in the first column.

Example Write-protect Recipe



Entries		Write protection		General	
From	To	Activation permit if	Deactivation permit if	Logic	Color
0	12	X #.SafetySimulation.ProgressBarVisible	X #.xAlarmAnimateAllOff	is ON (=1)	🟩

Figure 499: Recipe Editor dialog box with write protection settings for a number of entries of <4>.



0 composition1
1 composition2
2 composition3
3 composition4

Figure 500: Device screen at runtime

Example showing a recipe visualization object on the device screen

The screenshot below shows a recipe visualization object with a selection list containing recipe entries. This particular example features eight function key visualization objects linked to navigation functions that can be used to scroll through the recipe entries.

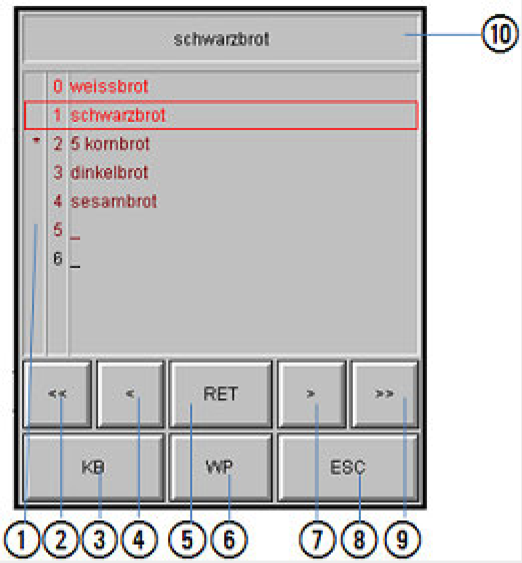


Figure 501: Device screen with recipe visualization object and function key visualization objects

The currently selected recipe entry will be highlighted with a color frame. The individual columns can be hidden if required; please refer to → "Write protection", page 760, → "Entry number", page 749, and → "Entry name", page 749.

① List of all recipe entries
Contains the "Write protection," "Entry number," and "Entry name" columns (an asterisk in the "Write protection" column indicates that the corresponding entry is write-protected; the "Entry name" column shows the corresponding recipe entry's name)

② << Used to move one page up within the selection list

③ KB Pressing the KB Button causes an alphanumeric keyboard to appear for entering or modifying a name of the selected recipe entry.

④ < Moves inside the selection list one Item up.

⑤ RET Quits this dialog box and returns to the selected recipe entry.

⑥ WP This button can be used to enable and disable write protection for a selected recipe entry. If write protection is currently disabled and the button is clicked on tapped on, write protection for the selected recipe entry will be enabled and vice versa.

⑦ Button: > Moves inside the selection list one Item down.

⑧ ESC Quits this Window without a recipe change, i.e. the currently enabled recipe entry will be retained. Changes to the recipe entry Type name and Write protection are not undone.

⑨ >> Moves inside the selection list one page bottom.

⑩ Title
Name of the currently selected recipe entry

About RET: If a recipe entry is not read-only, the current tag values are saved in the old recipe entry before being added to the new recipe entry. If a recipe entry is not write-protected, the current tag values are saved in the old recipe entry before moving to the new recipe entry. If the same recipe entry is selected a second time, the recipe entry will be loaded without saving the current tag values to the file. If a recipe entry without a Type name is selected, the current values of the tags are maintained and the values from the recipe file are not read.

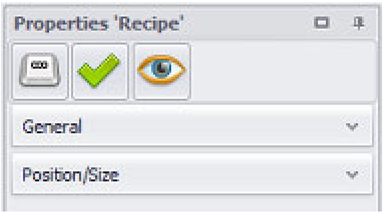
8.2 Recipe

Recipe visualization objects are used to visually represent recipe tag objects on the device screen at runtime in the form of a selection list of recipe entries. The properties for these visualization objects can be used to change the way in which they are displayed, e.g., in terms of position and size

→ Recipe entries can be selected directly within the window.

Prerequisites
In order to be able to follow the descriptions below, you must first have clicked on a recipe visualization object on the current screen.

Controls\ tab



Properties 'Recipe'

General

Position/Size

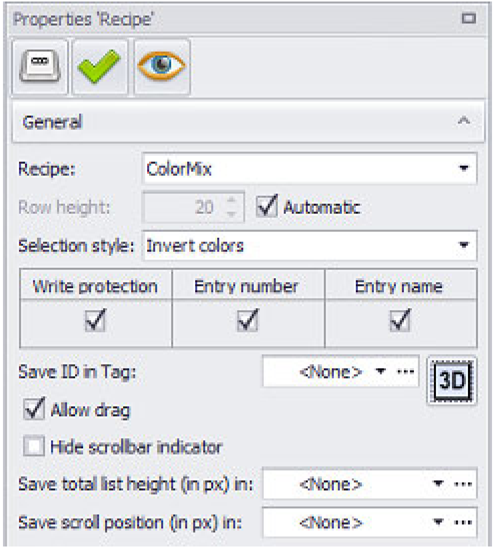
'Recipe' properties

→ " States", page 751

→ "General", page 748

General

Objects\ Recipe



Properties 'Recipe'

General

Recipe: ColorMix

Row height: 20 ☑ Automatic

Selection style: Invert colors

Write protection	Entry number	Entry name
☑	☑	☑

Save ID in Tag: <None>

☑ Allow drag

☐ Hide scrollbar indicator

Save total list height (in px) in: <None>

Save scroll position (in px) in: <None>

Figure 502: Properties 'Recipe' pane, General section

Recipe

Used to select the recipe tag object that will be visualized with the visualization object. In order to be able to use this field, you must first create at least one recipe tag object with the Recipe Editor.

Row height

Used to define the row height in pixels independently of the selected font size. If the row height is smaller than the font size, the text will be cut at the top and bottom.

☑ Automatic adaptation

Automatic

If this option is enabled ☑, the row height will be automatically calculated in such a way that it will be tall enough for showing all the entries in the list. This can come in handy, since you can use different font sizes for the "Enabled" and "Disabled" states.

Selection style

The appearance can be defined for each list.

Invert colors and frames are available.

Depending on the selection, the active line on the device is either inverted, i.e. differently colored text line background or a frame line at the top and bottom marks the active line.

Write protection

If this option is enabled, the Write protection column will be shown in the visualization object's selection list for all recipe entries; otherwise, the column will not be shown in the selection list. In addition, every write-protected recipe entry will be identified by an asterisk in the first column. Write protection can be configured in *Home\Recipes\Write protection tab*.

Entry number

If this option is enabled, the Entry number column will be shown in the visualization object's selection list for all recipe entries; otherwise, the column will not be shown in the selection list. Entry numbers are generated automatically by GALILEO and can be found in *Home\Recipes\Entries tab*.

Entry name

If this option is enabled, the Entry name column will be shown in the visualization object's selection list for all recipe entries; otherwise, the column will not be shown in the selection list. Entry names are defined in *Home\Recipes\Entries tab\Entry column*.

Save ID in Tag

If this checkbox is enabled, the ID of the entry selected at runtime will be stored in the selected tag. As soon as the checkbox is enabled, a tag must be selected from the drop-down menu – otherwise, an error message will appear during compilation (<There is no defined tag for storing the ID.>).

→ The actual ID will be the value passed. The value displayed in the list may be different from the ID if a number format is being used for the display.

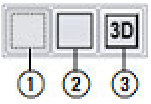
Selection mode on panel Single Click

The following modes can be adjusted for visualization objects with a keyboard:

- Single Click
  Tapping once on a list item or a button on the device screen will activate the selection.
- Delayed
  This option is ideal for implementing a safety function designed to protect against accidental actions. In order for a button on the device screen to be activated, it will first have to be pressed and held (long press) for the configured delay time. By default, this delay time will be 0.5 s, but it can be changed in *Home\Settings\Runtime tab\Touch Settings*.
- Double Click
  Tapping twice on a list item or button on the device screen will select the item / activate the button. By default, the double-click speed will be 0.5 s for a double-click, but it can be changed in*Home\Settings\Runtime tab\Touch Setting*.

Frame 3D

The visualization object can be displayed with any one of three different designs.



①	no frame
②	with frame
③	3D frame

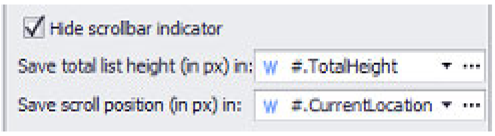
Allow drag

These touch gestures can only be used with PTC devices such as the XV300 and XP500.
This function is also supported by GALILEO Open applications.
If this checkbox is enabled, it will be possible to scroll through the entries in the list on the device screen at runtime. To do so, the person using the device will need to swipe over the list in the desired scrolling direction.

Hide scrollbar indicator

In order to work with objects for which a list is stored (parameter lists, alarm lists, recipes, and containers) more efficiently, you can save the position in the list.

☑ If the checkbox is enabled, the following settings for the list object will be hidden.

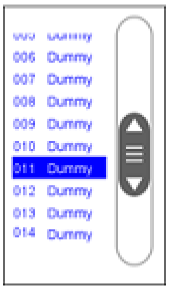


☑ Hide scrollbar indicator
Save total list height (in px) in: W #.TotalHeight
Save scroll position (in px) in: W #.CurrentLocation

Save list height (in px) in:	Saves the total list height (total height with all items added up) in the specified tag
Save scroll position (in px) in:	Saves the current scroll position

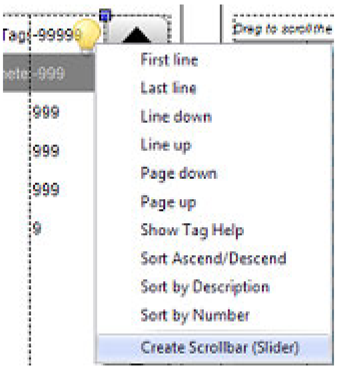
within the total list height in the specified tag.

Together with Save list height (in px) in, this tag can be used, for example, in order to use the slider in a slider object as a scrollbar in the list.



To make it easier to create this slider object, you can follow the steps below:

1. Assign tags to the Save list height (in px) in and Save scroll position (in px) in settings
2. In the screen designer, move the cursor to the upper right corner of the object. The 💡 icon will appear
3. Click on the 💡 icon and select the Generate scrollbar (slider) option in the drop-down menu

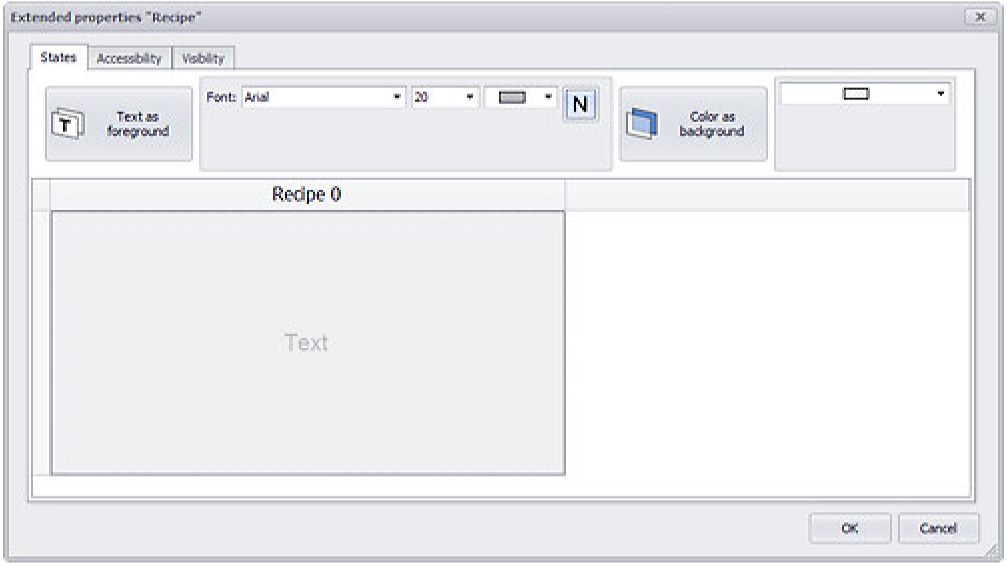


4. A slider object will be created and the tags will already be assigned correctly.

States

Recipe visualization objects only have one state, and the only foreground option for this state is using text.

Objects\ Recipe



The basic configuration parameters for the visualization object's foreground and background are described in detail in the following section: Defining text series

8.3 Recipes (Recipe Editor)

The recipes tag object can be used to manage multiple recipe objects.

In manufacturing, a Recipe usually refers to a specific product model or a specific method. In GALILEO, a recipe is a way to group various tags into a tag set. The tags in a tag set are filled with values, resulting in one or more value sets. In turn, each value set counts as one recipe entry. Finally, multiple recipe entries are grouped into a single Recipe object. This Recipe tag object is not a visualization object, but is instead an object used to create and manage the various value sets. Each Recipe tag object, together with all the values for the various corresponding recipe entries, can be saved on the relevant device in a <RecipeObjectName.TAB> file.

The values for a recipe entry can be preset. It is also possible to allow device operators to enter them.

Each recipe contains multiple recipe entries.

Each recipe entry in a recipe consists of the following components:

- Name of recipe entry
- The tag set is filled with values; each of the resulting value sets corresponds to a recipe entry in GALILEO.
- Option: Write protection on / off
  If write protection is enabled, it will not be possible to overwrite the corresponding recipe entry's data

There are several functions that can be applied to the recipe object, making it possible to save and load a recipe tag object. There are also additional functions that can be used to define how recipe entries in a recipe visualization object will be sorted, as well as functions for navigating in the selection list displayed with a recipe visualization object. Triggering these functions always requires the use of a function key. For a description of the function key, please refer to the → "Using function keys to trigger actions", page 389 section. For a description of these functions, please refer to the → section "Recipe functions", page 764 section.

The Recipe Editor can be used to create recipes and folders.
If you will be using a large number of recipes, you should set up a manageable folder structure in advance.
Clicking on the Add button will give you the option of selecting whether you want to create a folder or a recipe.

Clicking on the Remove button will delete the selected recipe or folder without a confirmation prompt.

The ▦ button will open a table overview of the tags being used in the recipes you have created. For more information, please refer to → "Recipe overview", page 761.

Home\ Recipes tab

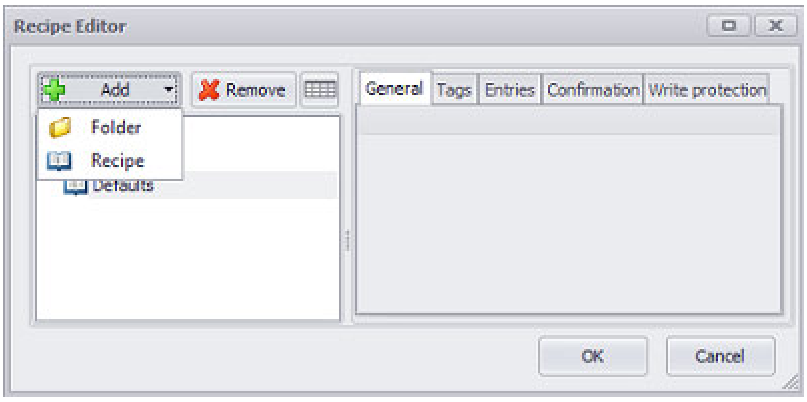


Figure 503: 'Recipe Editor' properties

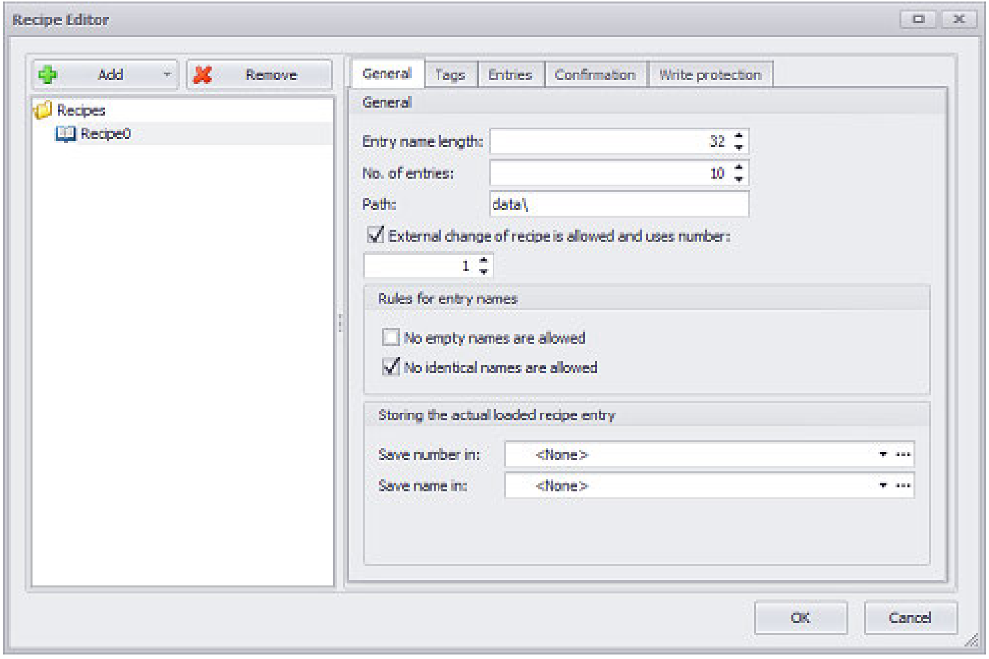
Register General

The properties in the General tab are for the recipe tag object, the recipe file, and, predominantly, the corresponding recipe entries.

Prerequisites

A recipe object from the Recipes folder must already be selected.

Home\ Recipes\General



Entry name length: 32

No. of entries: 10

Path: data\

☑ External change of recipe is allowed and uses number: 1

Rules for entry names

☐ No empty names are allowed

☑ No identical names are allowed

Storing the actual loaded recipe entry

Save number in: <None>

Save name in: <None>

Figure 504: Recipe Editor dialog box with the Color Mixture recipe object selected and the General tab open

Entry name length

The maximum name length for all the recipe entries in the selected recipe object is specified in this field. You can enter any value within a range of 1 to 255. The actual names for the recipe entries are entered under the Entries tab.

Number of entries

GALILEO automatically assigns an index to each recipe entry. This index will be required later on if you want a function to be triggered by the PLC.

The external recipe change, starting from the PLC, can only be made for the first 4096 entries.

The set number of recipe entries should be the number of entries expected for the recipe object even if no values have yet been assigned. When displayed on the device screen at runtime, the selection list will have a row for each recipe entry.

Path

This field is used to enter the path where the recipe tag object should be saved as a recipe file. data\ will be the default path.

→ Users must make sure that the specified path actually exists and that the recipe files will be copied there before the device is started.

External change of recipe is allowed and uses number

The number selected in the corresponding field will be assigned to the selected recipe object. Values between 0 and 15 are permissible. This number will be required later on if you want the PLC to trigger an external recipe change. For detailed information on the things that need to be kept in mind for recipe changes triggered by PLCs, please refer to the → "Example: Loading a recipe entry from the PLC", page 615 section.

Rules for entry names

These options can be used to set up naming rules for recipe entries, making it possible to check and restrict the names entered by device operators.

→ Before saving a recipe, GRS will check the naming rules listed below.
If any of these naming rules are broken, a corresponding system error message will be shown.
The recipe will not be saved.

No empty Type names are allowed

If this checkbox is enabled, it will be mandatory for a name to be assigned to every recipe entry and entered in the recipe file. This requires for a name to be entered for each recipe entry in the Name row, under the Entries tab.

No identical Type names are allowed

If this checkbox is enabled, it will be mandatory to assign a unique name to each recipe entry, i.e., a name that is different from the other recipe entry names. This requires for this name to be entered for each recipe entry in the Name row, under the Entries tab.

Save the currently loaded recipe entry

The number and name stored here are those corresponding to the recipe that is currently loaded.
"Save ID" means the number of the currently selected recipe.

Save number in:

If a tag is selected here, the ID for the currently selected recipe entry will be saved in that numeric tag. The limits for the numeric tag must be set accordingly. For a description of numeric tags, please refer to the→ "Numeric tag limits", page 563 section. The aforementioned ID will be automatically assigned, and can be found in the column for the corresponding recipe entry under the Entries tab.

Save name in:

If a tag of data type STRING is selected here, the name of the currently selected recipe entry will be saved in that tag. The limits for the string tag must be set accordingly. Moreover, the Number of characters defined in the properties for the string tag must be greater than or equal to the number set in Entry name length. For a description of strings tags, please refer to the → "String tag limits ", page 567 section.

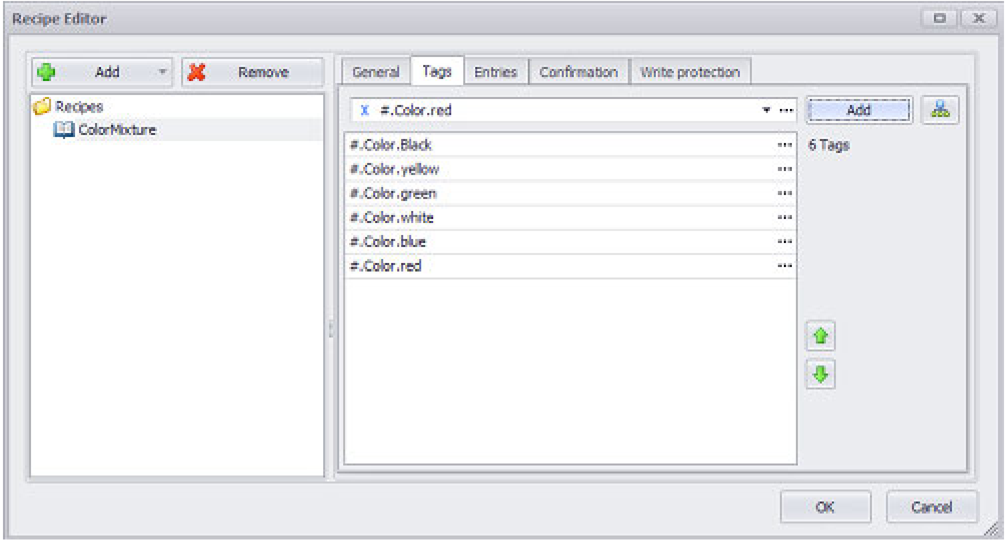
Tags tab

The settings in the Tags tab can be used to assign tags to the selected recipe object. This makes it possible to put together the parameter set that will later be available to all the recipe entries for the selected object and for which values can be preset under the Entries tab.

A maximum of 2000 tags can be assigned to each object.

Prerequisites
If you want to add tags, make sure to first create them in *Project Tree/Tags* or import them.

Home\ Recipes\Tags



#.Color.Black
#.Color.yellow
#.Color.green
#.Color.white
#.Color.blue
#.Color.red

Figure 505: Tag

Clicking on the drop-down menu arrow for the Tag field makes it possible to select tags that you want to add to the tag set for the selected recipe object. Clicking on the Add button will actually add the selected tag, which will then be shown in the list underneath.

Button ...

Limits

Clicking on the ... icon will show the limits for the value tag in a separate dialog box. You can also reach this dialog box via *Project Tree\Tags\<Value tag>\Properties*. When a tag is created, it will already be assigned limits based on the value range defined for it. This value range can be signed or unsigned and will depend on the relevant tag's data type.

The limits can be adjusted as necessary within the value range. In the case of bit tags, you will only be able to specify a default value.

For a description of the tag properties and limits, please refer to: → section "Numeric tag limits", page 563.

Tag Address

This field will only appear if you select a PLC tag, i.e., it will not appear for internal tags.
It will show the PLC address for the selected tag. The properties for PLC tags are described in greater detail in a different section – please refer to → section "Tag address", page 569.

If a single tag is transmitted to more than one communication, the corresponding settings will need to be configured in *Home\Gateway*. For detailed information on how to work with a gateway, please refer to → section "Using the gateway functionality to communicate with multiple PLCs", page 596.

Add

Clicking on this button will add the previously selected tag to the recipe tags.

🖧

Clicking on the 🖧 icon will open the Select tags dialog box, which in turn will have all the tags in *Project Tree\Tags* available for selection.
To select one or more tags, simply enable the corresponding checkbox ☑.

Up/Down ⇧ ⇩

Clicking on ⇧ will move the selected tag one place up in the list; clicking on ⇩ will move it one place down. The order in the list determines the order of the tags in the Entries tab.

(Value) Entries tab

The settings in the Entries tab can be used to preset or edit values for the selected recipe tag object.

Each column in the list corresponds to one recipe entry. This means that the number of columns will be the same as the number entered in the No. of entries field under the General tab. If there is no table in the Entries tab, open the General tab and enter the number of recipe entries you want into the No. of entries field first.

The number of rows will be the same as the number of tags added to the selected recipe object in the Tags tab. Each row can be used to enter the values for the corresponding tag, with each column corresponding to a different recipe entry.

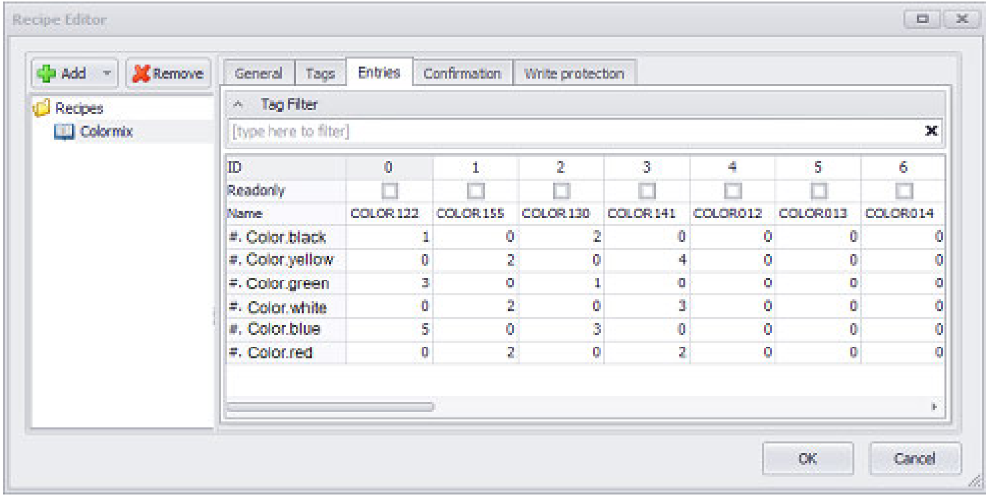
Every recipe entry will automatically be assigned a number in the first row. This number is assigned automatically by GALILEO and cannot be changed.

All other cells can be edited as soon as you click on them.

Float tags can be entered in the following number formats:

- ±x.xxxxxxE±xx (max. 6 decimal places)
- Floating point number (max. 13 digits including decimal point)

Home\ Recipes\Entries



ID	0	1	2	3	4	5	6
Readonly	□	□	□	□	□	□	□
Name	COLOR122	COLOR155	COLOR130	COLOR141	COLOR012	COLOR013	COLOR014
#. Color.black	1	0	2	0	0	0	0
#. Color.yellow	0	2	0	4	0	0	0
#. Color.green	3	0	1	0	0	0	0
#. Color.white	0	2	0	3	0	0	0
#. Color.blue	5	0	3	0	0	0	0
#. Color.red	0	2	0	2	0	0	0

Figure 506: Entries tab with eight recipe entries and values for a total of six tags

Tag filters

The Tag Filter can be used to filter the tags shown in the Name column. If the substring in the [type here to filter] box matches the name of a tag, the corresponding row(s) will be shown with values.

Readonly

When this checkbox is enabled, it will still be possible to edit the corresponding recipe entry at runtime, but it will be write-protected, meaning it will not be possible to save or delete it. Attempting to run any of the following functions will make a system message appear: Clear Recipe Entry, Paste Recipe Entry, Restore Default Values, Save Recipe Entry, Save Recipe Entry As, Save As/Load Rec. Entry.

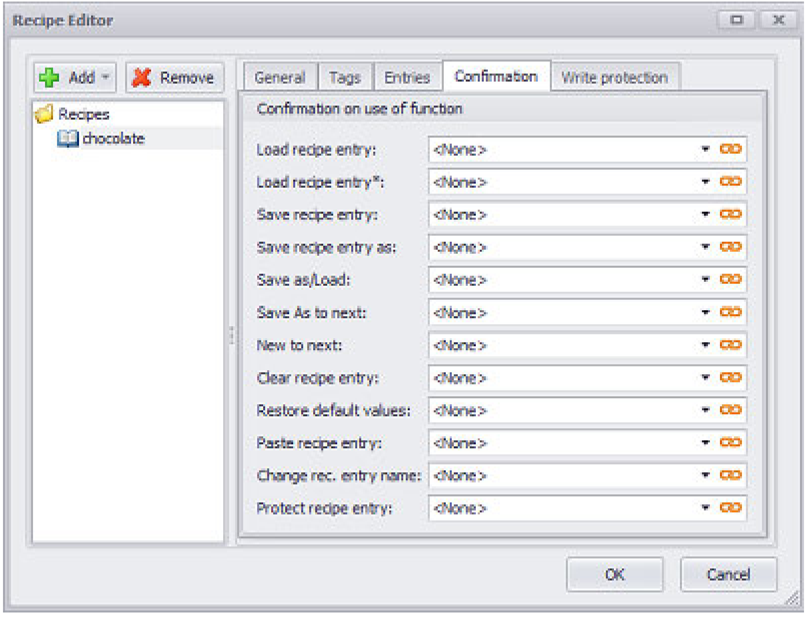
Name

This cell can be used to enter the name for the corresponding recipe entry and edit it. This recipe entry name will be shown at runtime with a recipe visualization object if the Entry name checkbox is enabled in the properties for the visualization object.

Confirmation tab

The Confirmation tab lists all the recipe visualization object functions to which a confirmation prompt can be added.
Confirmation prompts are prompts that require the device operator to confirm a function before the function is executed. The default is for no confirmation prompt to be added, which will be shown with a value of <None> in the corresponding fields.

Home\📖 Recipes\Confirmation



Confirmation on use of function

Load recipe entry: <None>
Load recipe entry*: <None>
Save recipe entry: <None>
Save recipe entry as: <None>
Save as/Load: <None>
Save As to next: <None>
New to next: <None>
Clear recipe entry: <None>
Restore default values: <None>
Paste recipe entry: <None>
Change rec. entry name: <None>
Protect recipe entry: <None>

Figure 507: Register Confirmation

For detailed information on how confirmation prompts are implemented, please refer to → "Using confirmation prompts to confirm whether functions should be run", page 340.

There is a global setting for defining whether a visualization is 🔗 linked or ⊂ ⊃ unlinked. In order for the 🔗 and ⊂ ⊃ chain link icons to be displayed, there must be more than one visualization in a group.
For more information, please refer to the → section " Web visualization", page 1070 section

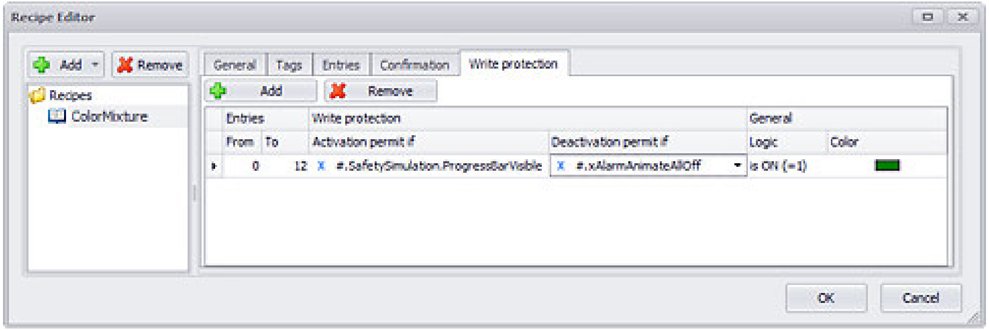
Write protection tab

Write protection enables Ranges of recipe entries to be protected from being overwritten or deleted.

When write protection is enabled, it will still be possible to edit the corresponding recipe entry at runtime, but it will be write-protected, meaning it will not be possible to save or delete it. Attempting to run any of the following functions will make a system message appear: Clear Recipe Entry, Paste Recipe Entry, Restore Default Values, Save Recipe Entry, Save Recipe Entry As, Save As/Load Recipe Entry.

In order to enable and disable write protection, you can use the Protect Recipe Entry function in the Recipes function group. This function can be triggered by the PLC as well.
Write protection will be enabled with the Protect Recipe Entry function and then disabled if the function is run again.

Home\ Recipes\Write protection



Entries		Write protection		General	
From	To	Activation permit if	Deactivation permit if	Logic	Color
0	12	X #.SafetySimulation.ProgressBarVisible	X #.xAlarmAnimateAllOff	is ON (=1)	🟩

Figure 508: Write protection tab specifying that twelve recipe entries (IDs 0 through 11) will be write-protected if the Communication0OK bit tag has a state of 1

Add button

Clicking on the Add button will add a condition for write protection or permanent (unconditional) write protection. In addition

Entries

Used to define the range of recipe entries that should be write-protected.

- From - ID of the recipe entry with which the range that should be write-protected starts
- To - ID of the recipe entry with which the range that should be write-protected ends

Write protection

Activation permit if

Can be used to select a tag of data type BIT. If the state of this tag matches the condition in the Logic cell, this means that the condition is being met. in which case the device operator will be allowed to enable write-protection for the corresponding entries at runtime. This write protection can be implemented using the Protect Recipe Entry function in the Recipes function group. This function can be triggered by the PLC as well.

Deactivation permit if

Can be used to select a tag of data type BIT. If the state of this tag matches the condition in the Logic cell, this means that the condition is being met.

In this case, the device operator will be allowed to disable write-protection for the corresponding entries at runtime. This write protection can be implemented using the Protect Recipe Entry function in the Recipes function group. This function can be triggered by the PLC as well.

Logic

The cells in the logic column are used to specify how the tag under the Write protection column should be interpreted:

- is ON (=1): Positive logic, i.e., if <tag> = 1, write protection will be enabled / disabled.
- is OFF (=0): Negative logic, i.e., if <tag> = 0, write protection will be enabled / disabled.

Color

When recipe entries are write-protected, they can be shown with different colors selected precisely for this purpose. Accordingly, the recipe entries in a recipe object that are specified in the Entries column will be displayed using the color selected here when they are write-protected.

You can change the color if you want. The following options are available for this purpose:

- Transparent
- Standard Colors
- More Colors...

The most recently used colors will be shown (user-specific).

For detailed information on the available colors, please refer to the following section:
→ section "Color choice - Object's Color Palette", page 234

Recipe overview

The ▦ button will open a table overview of all the recipes you have created and the tags used in them. In the corresponding structure, the tag name is used for the rows and its use in the recipe is indicated in the columns.
You can adjust the view by clicking on ⊞ to expand all and ⊟ to collapse all.

If all the tags in a structure are used, the checkmark will be black. If, on other other hand, only some of the tags in the structure are used, the checkmark will be gray.

Please note that this overview cannot be used to make changes to tags.

Home\ Recipes

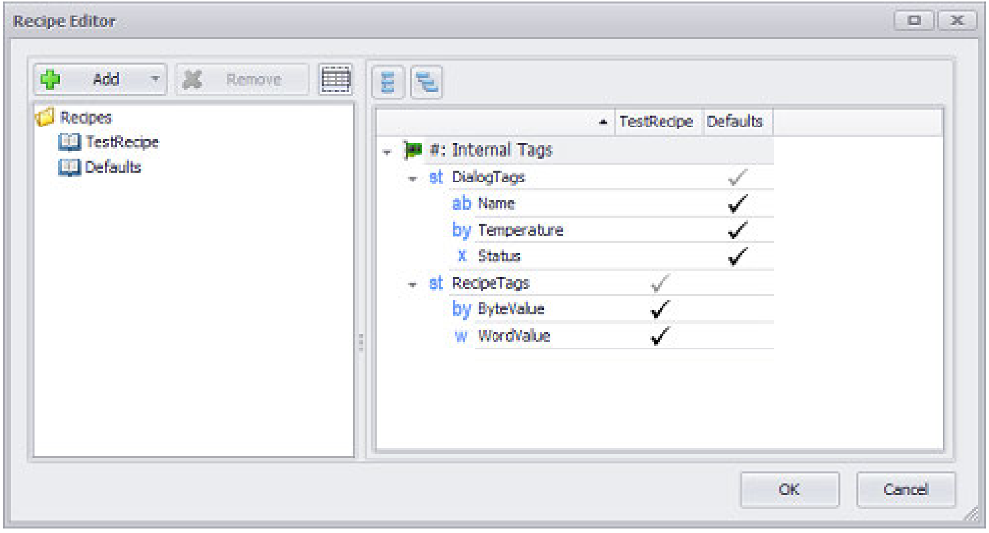


Figure 509: Recipe overview pane

The tree structure will only show tags that are used at least in one recipe.

✓	A grayed out / disabled checkmark indicates that only some elements from the corresponding structure or array are used in the recipe.
✓	A black / enabled checkmark indicates that all the elements in the corresponding structure or array are used in the recipe.

This view is meant exclusively as an overview and cannot be used to make any changes to tags or recipes.

Recipe File (*.TAB)

A recipe file contains all the data for a recipe: names, tag values, and write protection settings for the individual recipe entries.

When the project is compiled, a file with the name of the recipe tag object and the <TAB> extension will be generated. This recipe file will be stored in the path set in *Home\Recipes\General tab\Path field*. If the number of recipe entries or the tags assigned to the recipe change, the file will be converted during compilation. Data such as the names of the various recipe entries and the tag values will be kept during the conversion process.

This File contains all values of the different recipe entries in the following form. The columns are separated by a tab.

Write-protected recipe entries will be identified by an asterisk in the 1st column.

Example showing the content of a recipe file

*	**Recipe entry**	**Tag 1**	**Tag 2**	**Tag 3**	**Tag n**
	White bread	0	543	10	245
	Black bread	1234	-432	-8376	8821

	5 corn bread	8765	-1234	8764	1289
	Spelt bread	872	14	61	0
*	Sesame bread	3456	754	234	-758
	-				
	-				
	-				

Recipe files can be edited with any text editor or with Microsoft Excel®, for example. The program used must support text files with tab characters and must not replace these characters with spaces when saving.

8.4 Recipe functions

Functions can either be triggered by the PLC or by clicking or tapping on the button for a function key visualization object on the device screen. The following functions

The initiation from the PLC is implemented as follows:

- The function number is entered into the 4th data word of the control structure. Together with the status structure, this control structure is used to monitor and indicate the state of the communications between the MMI and PLC.
- Depending on the specific function, a parameter value may need to be passed to the 5th data word of the control structure.
- In the case of functions belonging to the Recipes group (the values for these tags can be saved on the touch panel under a specific name), the number and the recipe entry number must be entered in the 6th data word.
- With functions with an ON/OFF state, bit 14 of the 3rd data word is set according to the required state.
- Set bit 15 of the 3rd data word of the control data block so that the function is executed (see also: Control Data Block).

Other functions like Screen Change, "Recipe Change" etc. can be initiated from the PLC, too.

Recipe entries	Function	Note
Navigate	→ section "First Line (Recipe)", page 772 → section "Last Line", page 773 → section "Line Down", page 773 → section "Line Up", page 773 → section "Page Down", page 776 → section "Page Up", page 776	
(Re)name	→ section "Change Recipe Entry Name", page 766 → section "Change Recipe Entry Name userdef.", page 766	
Load	→ "Load Recipe Entry", page 774 → section "LoadRecipeEntryVar", page 775 → section "LoadSelectedRecipeEntry", page 775 → section "LoadLastRecipeEntry", page 774	Load Recipe Entry (modified) This prompt is shown when unsaved recipe data would be lost if another recipe entry is loaded.
Save	→ section "Save Recipe Entry", page 782 → section "Save Recipe Entry As", page 783 → section "Save Recipe Entry As userdef.", page 781 → section "Save Recipe Entry Var", page 784 → section "Save Recipe Entry As Var",	

	page 783 → section "Save As/Load Recipe Entry", page 782 → section "Save As/Load Recipe Entry userdef.", page 779 → section "SaveActualRecipeEntry", page 782 → section "Save As to next", page 779 → section "Save As to next userdef", page 780 → section "Save As to next Var", page 780	
Delete	→ section "Clear Recipe Entry", page 767	Deletes the name of the recipe file and resets the tags for the recipe entry to the values preset in the tag properties in the recipe file
Recipe entry ID	→ section "Display: Copy Entry", page 768 → section "Display: Loaded Entry", page 769 → section "Display: New Entry", page 769 → section "Display: Selected Entry", page 769	
Copy&Paste	→ section "Copy Recipe Entry", page 768 → section "Copy to next", page 767 → section "Paste Recipe Entry", page 777	
Write-protect	→ section "Protect Recipe Entry", page 777	
Find	→ section "Find Recipe Entry", page 770 → section "Find Recipe Entry (start)", page 771 → section "Find Recipe Entry userdef.", page 771 → section "Find Rcp Entry userdef. (start) ", page 771 → section "Find Next Recipe Entry", page 772 → section "Find Next Recipe Entry (start)", page 770	
Misc.	→ section "New to next", page 775 → section "New to next userdef.", page 776 → section "Restore Default Values", page 778 → section "Select Loaded Recipe Entry", page 784	new - Resets the tags for the recipe entry to the values preset in the tag properties in the recipe file.

	→ section "Refresh Labels", page 778

See also

→ section " Recipe", page 748
→ " Recipes (Recipe Editor)", page 753

8.4.1 Change Recipe Entry Name userdef.

This function can be used to change the name of the selected recipe entry on the device screen.

When triggered from the touch panel

If this function is triggered from the device, the user-defined keyboard will appear so that the operator can enter the recipe entry name they want. The name of the recipe object is passed as a parameter in the definition for the → "Function keys", page 386. Accordingly, the name of the selected recipe entry will be changed.

When triggered from the PLC

The corresponding PLC text with the new name for the recipe entry needs to be written. If the function is then called by the PLC, this will cause the name of the recipe entry to be copied to the recipe file and updated on the device.

→ This function will only be run if the selected recipe entry is not write-protected.

Number of the Function	2070 (816h)
Executable from the PLC	No
Function with ON/OFF state	No
Parameter 1	Name of recipe object ; the recipe and function key visualization objects must be on the same screen
Parameter 2	Selected User Defined Keyboards
Function available in standard screens	Yes
Function available with keyboards	No
Function available in script	No
Function available in application entry script	No

8.4.2 Change Recipe Entry Name

This function can be used to change the name of the selected recipe entry on the device screen.

When triggered from the device

If this function is triggered from the device, an alphanumeric keyboard will appear so that the operator can enter the recipe entry name they want. The name of the recipe object is passed as a parameter in the definition for the function. Accordingly, the name of the selected recipe entry will be changed.

When triggered from the PLC

The corresponding PLC text with the new name needs to be written. If the function is then called, this will cause the name of the recipe entry to be copied to the recipe file and updated on the device.

→ This function will only be run if the selected recipe entry is not write-protected.

Number of the Function	2055 (807h)
Executable from the PLC	Yes
Function with ON/OFF state	No
Parameter 1	Name of recipe object
Function available in standard screens	Yes
Function available with keyboards	No
Function available in script	No
Function available in application entry script	No

8.4.3 Clear Recipe Entry

Using this function will reset the tag values for the selected recipe entry in the recipe file to zero.
In other words, all tag values will be set to "0" and the name of the recipe entry will be deleted.

This will only affect the loaded recipe if the selected recipe is the loaded recipe.

→ This function will only be run if the selected recipe entry is not write-protected.

→ The function key and recipe visualization objects must be found on the same screen.

Number of the Function	2053 (805h)
Executable from the PLC	No
Function with ON/OFF state	No
Parameter 1	Name of recipe object
Function available in standard screens	Yes
Function available with keyboards	No
Function available in script	No
Function available in application entry script	No

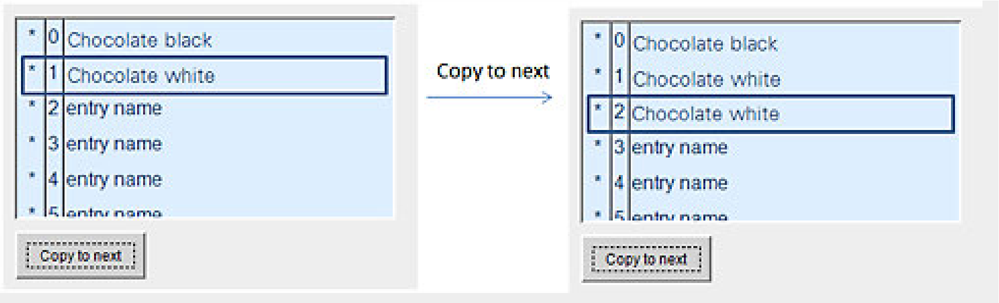
8.4.4 Copy to next

This function will take all the properties and tag values for the selected recipe entry and save them with the same name in the next free entry in the recipe file. Free recipe entries can be identified with the present <entry name> name.

→ The function key and recipe visualization objects must be found on the same screen.

Number of the Function	2081 (821h)
Executable from the PLC	No
Function with ON/OFF state	No
Parameter 1	Name of recipe object
Function available in standard screens	Yes
Function available with keyboards	No
Function available in script	No
Function available in application entry script	No

Example Copy to next



8.4.5 Copy Recipe Entry

This function copies all the properties and tag values for the selected recipe entry to the clipboard. After this, the values can be pasted at any point in the list with recipe entries by using the → "Paste Recipe Entry", page 777 special function. The data in the clipboard will be retained even if the → "First Line (Recipe)", page 772, → "Last Line", page 773, → "Line Down", page 773, → "Line Up", page 773, → "Page Down", page 776, → "Page Up", page 776, or Paste Recipe Entry function is run.

→ The function key and recipe visualization objects must be found on the same screen.

Number of the Function	2051 (803h)
Executable from the PLC	No
Function with ON/OFF state	No
Parameter 1	Name of recipe object
Function available in standard screens	Yes
Function available with keyboards	No
Function available in script	No
Function available in application entry script	No

8.4.6 Display: Copy Entry

This function will show the name of the recipe entry in the clipboard.
Please note that this function can only be used in conjunction with a confirmation prompt when copying recipe entries.

Number of the Function	2064 (810h)
Executable from the PLC	No

Function with ON/OFF state	No
Function available in standard screens	No
Function available with keyboards	Yes
Function available in script	No
Function available in application entry script	No

8.4.7 Display: Loaded Entry

This function will show the name of the recipe entry that is currently loaded. Please note that this function can only be used in conjunction with confirmation prompts.

This function will show the name of the recipe entry that is currently loaded. Please note that this function can only be used in conjunction with confirmation prompts.

Number of the Function	2066 (812h)
Executable from the PLC	No
Function with ON/OFF state	No
Function available in standard screens	No
Function available with keyboards	Yes
Function available in script	No
Function available in application entry script	No

8.4.8 Display: New Entry

This function will show the recipe entry number that will be used when generating the new entry. Please note that this function can only be used in conjunction with security prompts when generating new recipe entries.

This function will show the recipe entry number that will be used when generating the new entry. Please note that this function can only be used in conjunction with security prompts when generating new recipe entries.

Number of the Function	2082 (822h)
Executable from the PLC	No
Function with ON/OFF state	No
Function available in standard screens	No
Function available with keyboards	Yes
Function available in script	No
Function available in application entry script	No

8.4.9 Display: Selected Entry

This function will show the name of the selected recipe entry. Please note that this function can only be used in conjunction with confirmation prompts.

Numbers of the Functions	2065 (811h)
Executable from the PLC	No
Function with ON/OFF state	No
Function available in Standard screens	No

Function available with keyboards	Yes
Function available in loop scripts	No
Function available in event scripts	No
Function available in application entry script	No

8.4.10 Find Next Recipe Entry (start)

This function is used for finding the next recipe entry. The function will search for the string entered using the → "Find Recipe Entry (start)", page 771 function. This entry is highlighted if a recipe entry name from the first character matches with the search string. This function can be carried out several times, with the search always starting from the currently selected recipe entry . If None is found with this Type name, the currently selected entry remains highlighted.

→ The function key and recipe visualization objects must be found on the same screen.

Number of the Function	2075 (81Bh)
Executable from the PLC	No
Function with ON/OFF state	No
Parameter 1	Name of recipe object
Function available in standard screens	Yes
Function available with keyboards	No
Function available in script	No
Function available in application entry script	No

8.4.11 Find Recipe Entry

This function can be used to search for a recipe entry. An alphanumeric keyboard will be shown so that the operator can enter the search string they want. The search will always start from the beginning of the list. If a recipe entry name, or part of it, matches the search string, the corresponding entry will be selected. The→ "Find Next Recipe Entry", page 772 function can be used to find additional entries in the list.

→ The function key and recipe visualization objects must be found on the same screen.

Number of the Function	2068 (814h)
Executable from the PLC	No
Function with ON/OFF state	No
Parameter 1	Name of recipe object
Function available in standard screens	Yes
Function available with keyboards	No
Function available in script	No
Function available in application entry script	No

8.4.12 Find Recipe Entry (start)

This function can be used to search for a recipe entry.
An alphanumeric keyboard will be shown so that the operator can enter the search string they want.
The search will always start from the beginning of the list.
If a recipe entry name matches the search string starting from the first character, the corresponding entry will be selected.

→ The function key and recipe visualization objects must be found on the same screen.

The → "Find Next Recipe Entry (start)", page 770 function can be used to find additional entries in the list.

Number of the Function	2074 (81Ah)
Executable from the PLC	No
Function with ON/OFF state	No
Parameter 1	Name of recipe object
Function available in standard screens	Yes
Function available with keyboards	No
Function available in script	No
Function available in application entry script	No

8.4.13 Find Recipe Entry userdef.

This function can be used to search for a recipe entry. The user-defined keyboard will be shown so that the operator can enter the search string they want. The search will always start from the beginning of the list. If a recipe entry name, or part of it, matches the search string, the corresponding entry will be selected. The→ "Find Next Recipe Entry", page 772 function can be used to find additional entries in the list.

→ The function key and recipe visualization objects must be found on the same screen.

Number of the Function	2073 (819h)
Executable from the PLC	No
Function with ON/OFF state	No
Parameter 1	Name of recipe object
Parameter entered	Selected User Defined Keyboards
Function available in standard screens	Yes
Function available with keyboards	No
Function available in script	No
Function available in application entry script	No

8.4.14 Find Rcp Entry userdef. (start)

This function can be used to search for a recipe entry. The user-defined keyboard will be shown so that the operator can enter the search string they want. The search will always start from the beginning of the list. If a recipe entry name matches the search string starting from the first character, the corresponding entry will be selected. The → "Find Next Recipe Entry (start)", page 770 function can be used to find additional entries in the list.

→ The function key and recipe visualization objects must be found on the same screen.

Number of the Function	2076 (81ch)
Executable from the PLC	No
Function with ON/OFF state	No
Parameter 1	Name of recipe object
Parameter entered	Selected User Defined Keyboards
Function available in standard screens	Yes
Function available with keyboards	No
Function available in script	No
Function available in application entry script	No

8.4.15 Find Next Recipe Entry

This function is used for finding the next recipe entry. It will search for the search string entered with the Find Recipe Entry function. If a recipe entry name, or part of it, matches the search string, the corresponding recipe entry will be selected. This function can be carried out several times, with the search always starting from the currently selected recipe entry . If no additional recipe entries with the specified name are found, the selected recipe entry will remain selected.

→ The function key and recipe visualization objects must be found on the same screen.

Number of the Function	2069 (815h)
Executable from the PLC	No
Function with ON/OFF state	No
Parameter 1	Name of recipe object
Function available in standard screens	Yes
Function available with keyboards	No
Function available in script	No
Function available in application entry script	No

8.4.16 First Line (Recipe)

Using this function will make the selection jump to the first recipe entry in the list of recipe entries.

→ The function key and recipe visualization objects must be found on the same screen.

Number of the Function	2060 (80ch)
Executable from the PLC	No
Function with ON/OFF state	No
Parameter 1	Name of recipe object

Function available in standard screens	Yes
Function available with keyboards	No
Function available in script	Yes
Function available in application entry script	No

8.4.17 Last Line

When this function is used, the last recipe entry in the list of recipe entries will be selected.

→ The function key and recipe visualization objects must be found on the same screen.

Number of the Function	2061 (80dh)
Executable from the PLC	No
Function with ON/OFF state	No
Parameter 1	Name of recipe object
Function available in standard screens	Yes
Function available with keyboards	No
Function available in script	Yes
Function available in application entry script	No

8.4.18 Line Down

When this function is used, the next recipe entry in the list of recipe entries will be selected.

If the entry currently selected is the bottommost entry in the window, the list will be moved one line up.

→ The function key and recipe visualization objects must be found on the same screen.

Number of the Function	2057 (809h)
Executable from the PLC	No
Function with ON/OFF state	No
Parameter 1	Name of recipe object
Function available in standard screens	Yes
Function available with keyboards	No
Function available in script	Yes
Function available in application entry script	No

8.4.19 Line Up

When this function is used, the previous recipe entry in the list of recipe entries will be selected.

If the entry currently selected is the topmost entry in the window, the list will be moved one line down.

→ The function key and recipe visualization objects must be found on the same screen.

Number of the Function	2056 (808h)
Executable from the PLC	No
Function with ON/OFF state	No
Parameter 1	Name of recipe object
Function available in standard screens	Yes
Function available with keyboards	No
Function available in script	Yes
Function available in application entry script	No

8.4.20 LoadLastRecipeEntry

This function loads and refreshes the values for the last loaded recipe entry from the recipe file, i.e., the tag values (**please refer to the following as well:** Recipe tags), the current name of the recipe entry, and the current recipe entry number will be set and, if necessary, transmitted to the PLC. **No values will be saved** in the loaded recipe.

This function can be used after starting the device in order to load the recipe entry that was last loaded. A loop script that is only called if theOnPresetOK function returns "true" needs to be defined for this purpose.

Parameter entered	Name of recipe object
Function available in script	Yes
Function available in application entry script	Yes

8.4.21 Load Recipe Entry

This function loads and updates the values for the currently selected recipe entry from the recipe file,
i.e., the tag values, the current name of the recipe entry, and the current number of the recipe entry will be transferred to the PLC if necessary.
No values will be saved in the previously loaded recipe.

The recipe entry number is the number of an element in a specific recipe.

→ The device's loop will be interrupted when loading a recipe entry, and will only be resumed once all the tags in the recipe have been transferred to the PLC!

For a description of an external recipe change triggered by the PLC, please refer to:
→ section "Example: Loading a recipe entry from the PLC", page 615

Number of the Function	2050 (802h)
Executable from the PLC	Yes
Function with ON/OFF state	No
Parameter 1	Name of recipe object
Function available in standard screens	Yes

Function available with keyboards	No
Function available in script	No
Function available in application entry script	No

8.4.22 LoadRecipeEntryVar

This function loads and refreshes the values of the Recipe Entry Number from the recipe file, the tag values (see also Recipe Tags), the current Recipe Entry Name and the current Recipe Entry Number are set and transferred to the PLC if required. **No values are saved** in the previously loaded recipe.

Parameter entered	Name of recipe object
Parameter entered	Name of the Tags: Recipe entry number
Function available in script	Yes
Function available in application entry script	Yes

8.4.23 LoadSelectedRecipeEntry

This function loads and refreshes the values of the currently selected recipe entry from the recipe file, i.e. the tag values (see Recipe Tags), the current Recipe Entry Name and the current Recipe Entry Number are set and transferred to the PLC if required.

Parameter entered	Name of recipe object (The object must be in the same screen)
Function available in script	Yes
Function available in application entry script	Yes

8.4.24 New to next

If this function is triggered from the device, an alphanumeric keyboard will appear so that the operator can enter the recipe entry name they want. With this function, the tag preset values are stored under the entered Type name in the next free Item (without recipe entry name) in the recipe file. This function will not affect the recipe entry that is currently loaded (the corresponding data will not be changed).

→ The function key and recipe visualization objects must be found on the same screen.

Number of the Function	2079 (81Fh)
Executable from the PLC	No
Function with ON/OFF state	No
Parameter 1	Name of recipe object
Function available in standard screens	Yes
Function available with keyboards	No
Function available in script	No

Function available in application entry script	No

8.4.25 New to next userdef.

When this function is triggered on the touch panel, a User Defined Keyboard appears for entering the recipe entry name. With this function, the tag preset values are stored under the entered Type name in the next free Item (without recipe entry name) in the recipe file. This function will not affect the recipe entry that is currently loaded (the corresponding data will not be changed).

→ The function key and recipe visualization objects must be found on the same screen.

Number of the Function	2080 (820h)
Executable from the PLC	No
Function with ON/OFF state	No
Parameter 1	Name of recipe object
Parameter 2	Selected User Defined Keyboards
Function available in standard screens	Yes
Function available with keyboards	No
Function available in script	No
Function available in application entry script	No

8.4.26 Page Down

This function is used to move the selection bar one page (window height) to the bottom. The selection bar is then positioned inside the list of all recipe entries on the top line or at the first Item if the window is not full.

→ The function key and recipe visualization objects must be found on the same screen.

Number of the Function	2059 (80Bh)
Executable from the PLC	No
Function with ON/OFF state	No
Parameter 1	Name of recipe object
Function available in standard screens	Yes
Function available with keyboards	No
Function available in script	Yes
Function available in application entry script	No

8.4.27 Page Up

This function is used to move the selection bar one page (window height) to the top. The selection bar is then positioned inside the list of all recipe entries on the bottom line or at the last Item if the window is not full.

→ The function key and recipe visualization objects must be found on the same screen.

Number of the Function	2058 (80Ah)
Executable from the PLC	No
Function with ON/OFF state	No
Parameter 1	Name of recipe object
Function available in standard screens	Yes
Function available with keyboards	No
Function available in script	Yes
Function available in application entry script	No

8.4.28 Paste Recipe Entry

With this function, the selected recipe entry is overwritten with the tag values and recipe entry name of the Clipboard data.
The data needs to be stored previously in the clipboard with the → section "Copy Recipe Entry", page 768 function.
The data in the Clipboard is retained when executing the functions First Line, Last Line, Line Down, Line Up, Page Down, Page Up and → "Paste Recipe Entry", page 777.

Copying recipe entries is supported only within the Recipe Editor.
This function only affects the selected recipe if it has been currently loaded.

→ This function will only be run if the selected recipe entry is not write-protected.

→ The function key and recipe visualization objects must be found on the same screen.

Number of the Function	2052 (804h)
Executable from the PLC	No
Function with ON/OFF state	No
Parameter 1	Name of recipe tag object
Function available in standard screens	Yes
Function available with keyboards	No
Function available in script	No
Function available in application entry script	No

8.4.29 Protect Recipe Entry

This function will enable / disable write protection for the currently selected recipe entry in the recipe visualization object.

Running the Protect Recipe Entry function will enable write protection. Running it again will disable it.

Write protection is taken into account for the following functions:

→ section "Clear Recipe Entry", page 767
→ section "Change Recipe Entry Name", page 766
→ section "Change Recipe Entry Name userdef.", page 766
→ section "Paste Recipe Entry", page 777

→ section "Restore Default Values", page 778
→ section "Save Recipe Entry", page 782
→ section "Save Recipe Entry Var", page 784
→ section "Save Recipe Entry As", page 783
→ section "Save Recipe Entry As Var", page 783
→ section "Save Recipe Entry As userdef.", page 781
→ section "Save As/Load Recipe Entry userdef.", page 779

Write protection activated

The values of this recipe entry cannot be overwritten until write protection has been removed.

Write protection not activated

When the functions listed above are run, the values for the recipe entry will be changed.

Number of the Function	2048 (800h)
Executable from the PLC	Yes
Function with ON/OFF state	No
Parameter 1	Name of recipe object
Function available in standard screens	Yes
Function available with keyboards	No
Function available in script	No
Function available in application entry script	No

8.4.30 Refresh Labels

This function reads the recipe entry names from the recipe file again.

→ This function should only be used if the recipe files (.*tab files) are edited externally during ongoing operation.

Number of the Function	2085 (825h)
Executable from the PLC	No
Function with ON/OFF state	No
Function available in standard screens	Yes
Function available with keyboards	Yes
Function available in script	Yes
Function available in application entry script	Yes

8.4.31 Restore Default Values

This function sets the tags for the selected recipe entry in the recipe file to the corresponding default values. These default values are set in the properties for the tags, in the Limits section – please refer to the → " Tags", page 563, → section "Value tag default", page 565, and → section "String tag default", page 568 sections as well. The recipe entry's name will not be changed. This will only affect the loaded recipe if the selected recipe is the loaded recipe.

→ This function will only be run if the selected recipe entry is not write-protected.

→ The function key and recipe visualization objects must be found on the same screen.

Numbers of the Functions	2054 (806h)
Executable from the PLC	No
Function with ON/OFF state	No
Parameter 1	Name of recipe tag object
Function available in Standard screens	Yes
Function available with keyboards	No
Function available in script	No
Function available in application entry script	No

8.4.32 Save As/Load Recipe Entry userdef.

If this function is triggered from the device, the user-defined keyboard will appear so that the operator can enter the recipe entry name they want. This Load Recipe Entry function will take the data at the selected position of the currently loaded recipe and save it in the recipe file. In addition, the selected position will become the currently loaded recipe entry (see Load Recipe Entry). The previously loaded recipe entry will not be modified by this function.

→ The function key and recipe visualization objects must be found on the same screen.

→ This function will only be run if the selected recipe entry is not write-protected.

Number of the Function	2072 (818h)
Executable from the PLC	No
Function with ON/OFF state	No
Parameter 1	Name of recipe tag object
Parameter 2	Selected User Defined Keyboards
Function available in standard screens	Yes
Function available with keyboards	No
Function available in script	No
Function available in application entry script	No

8.4.33 Save As to next

If this function is triggered from the device, an alphanumeric keyboard (Alpha numeric) will appear so that the operator can enter the recipe entry name they want. This function will take the current tag data and save it in the next free recipe entry in the recipe file under the name entered by the operator. This function will not affect the recipe entry that is currently loaded (the corresponding data will not be changed).

→ The function key and recipe visualization objects must be found on the same screen.

Number of the Function	2077 (81dh)
Executable from the PLC	No
Function with ON/OFF state	No
Parameter 1	Name of recipe tag object
Function available in standard screens	Yes
Function available with keyboards	No
Function available in script	No
Function available in application entry script	No

8.4.34 Save As to next userdef

When this function is triggered on the touch panel, a User Defined Keyboard appears for entering the recipe entry name. This function stores the current tag values under the selected name in the next free recipe entry in the recipe file. This function will not affect the recipe entry that is currently loaded (the corresponding data will not be changed).

→ The function key and recipe visualization objects must be found on the same screen.

Number of the Function	2078 (81Eh)
Executable from the PLC	No
Function with ON/OFF state	No
Parameter 1	Name of recipe tag object
Parameter 2	Selected User Defined Keyboards
Function available in standard screens	Yes
Function available with keyboards	No
Function available in script	No
Function available in application entry script	No

8.4.35 Save As to next Var

This function stores the current tag values under the selected name in the next free recipe entry in the recipe file. This function will not affect the recipe entry that is currently loaded (the corresponding data will not be changed).

→ The function key and recipe visualization objects must be found on the same screen.

Number of the Function	2084 (824h)
Executable from the PLC	No
Function with ON/OFF state	No
Parameter 1	Name of recipe tag object
Parameter 2	Type name of recipe entry
Function available in standard screens	Yes
Function available with keyboards	No

Function available in script	Yes
Function available in application entry script	Yes

Sample script:

Say that a recipe entry has been loaded from the <Color Mixture> tag object, e.g., <COLOR155>, and that the tag data has been changed. Now say that you want this tag set to be stored again in the <Color Mixture> tag object under the next free recipe entry with the name <COLOR015>.

▶ Add the following lines to the script:

```
Recipe.SaveAsToNextVar(Color Mixture, Recipe.COLOR015);
```

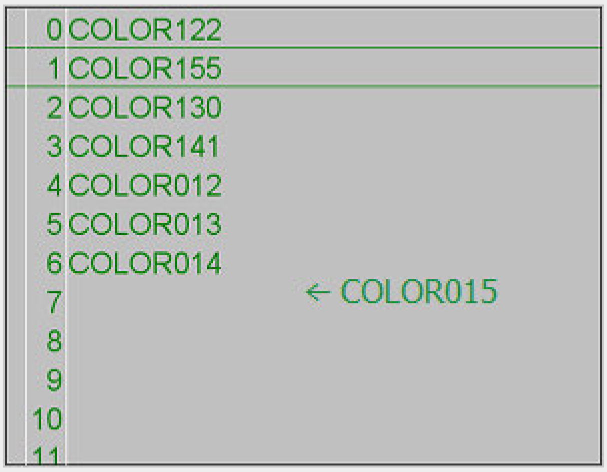

0 COLOR122
1 COLOR155
2 COLOR130
3 COLOR141
4 COLOR012
5 COLOR013
6 COLOR014
7
8
9
10
11

Figure 510: Recipe <Color Mixture>

8.4.36 Save Recipe Entry As userdef.

If this function is triggered from the device, the user-defined keyboard will appear so that the operator can enter the recipe entry name they want. This function will take the current tag data and save it in the recipe file under the name entered by the operator. This function will not affect the recipe entry that is currently loaded (the corresponding data will not be changed).

→ This function will only be run if the selected recipe entry is not write-protected.

→ The function key and recipe visualization objects must be found on the same screen.

Number of the Function	2071 (817h)
Executable from the PLC	No
Function with ON/OFF state	No
Parameter 1	Name of recipe tag object
Parameter 2	Selected User Defined Keyboards
Function available in standard screens	Yes
Function available with keyboards	No
Function available in script	No
Function available in application entry script	No

8.4.37 SaveActualRecipeEntry

This function saves the current tag values under the currently loaded recipe entry in the recipe file.

→ This function will only be run if the loaded recipe entry is not write-protected.

Parameter entered	Name of recipe object
Function available in script	Yes
Function available in application entry script	Yes

8.4.38 Save As/Load Recipe Entry

If this function is triggered from the device, an alphanumeric keyboard will appear so that the operator can enter the recipe entry name they want. This function will take the data at the selected position of the currently loaded recipe and save it in the recipe file. In addition, the selected position will become the currently loaded recipe entry (see Load Recipe Entry). The previously loaded recipe entry will not be modified by this function.

→ This function will only be run if the selected recipe entry is not write-protected.

→ The function key and recipe visualization objects must be found on the same screen.

Number of the Function	2067 (813h)
Executable from the PLC	No
Function with ON/OFF state	No
Parameter 1	Name of recipe tag object
Function available in standard screens	Yes
Function available with keyboards	No
Function available in script	No
Function available in application entry script	Yes

8.4.39 Save Recipe Entry

This function saves the current tag values under the currently loaded recipe entry in the recipe file.

→ This function will only be run if the selected recipe entry is not write-protected.

Number of the Function	2049 (801h)
Executable from the PLC	No
Function with ON/OFF state	No
Parameter 1	Name of recipe tag object
Only when the script is called: Parameter 2	Recipe entry number
Function available in standard screens	Yes

Function available with keyboards	No
Function available in script	No
Function available in application entry script	No

8.4.40 Save Recipe Entry As

If this function is triggered from the device, an alphanumeric keyboard will appear so that the operator can enter the recipe entry name they want. This function will take the current tag data and save it in the recipe file under the name entered by the operator. This function will not affect the recipe entry that is currently loaded (the corresponding data will not be changed).

→ This function will only be run if the selected recipe entry is not write-protected.

→ The function key and recipe visualization objects must be found on the same screen.

Number of the Function	2063 (80Fh)
Executable from the PLC	No
Function with ON/OFF state	No
Parameter 1	Name of recipe tag object
Function available in standard screens	Yes
Function available with keyboards	No
Function available in script	No
Function available in application entry script	No

8.4.41 Save Recipe Entry As Var

This function stores the current tag values under the selected name into the recipe file. This function will not affect the recipe entry that is currently loaded (the corresponding data will not be changed).

→ This function will only be run if the selected recipe entry is not write-protected.

→ The function key and recipe visualization objects must be found on the same screen.

Number of the Function	2083 (823h)
Executable from the PLC	No
Function with ON/OFF state	No
Parameter 1	Name of recipe tag object
Function available in standard screens	Yes
Function available with keyboards	No
Function available in script	No
Function available in application entry script	No

8.4.42 Save Recipe Entry Var

This function will take the current tag data and save it under the specified recipe entry number.

→ This function will only be run if the selected recipe entry is not write-protected.

→ The function key and recipe visualization objects must be found on the same screen.

Number of the Function	–
Executable from the PLC	No
Function with ON/OFF state	No
Parameter 1	Name of recipe tag object
Parameter 2	Tag with recipe entry number
Function available in standard screens	No
Function available with keyboards	No
Function available in script	Yes
Function available in application entry script	Yes

8.4.43 Select Loaded Recipe Entry

Thus function returns the selection bar in the display to the currently loaded recipe entry.

→ The function key and recipe visualization objects must be found on the same screen.

Number of the Function	2062 (80Eh)
Executable from the PLC	No
Function with ON/OFF state	No
Parameter 1	Name of recipe tag object
Function available in standard screens	Yes
Function available with keyboards	No
Function available in script	Yes
Function available in application entry script	No

9. Parameter list

A parameter list is a list of tags that can be grouped freely. A parameter list will be shown on the device screen as soon as a screen containing a parameter list object is shown.

A visualization object is used to graphically represent parameter lists on the screen of the device being used. The properties for this visualization object can be used to customize how the object will be displayed, including features such as its position and size. As the name implies, each parameter list consists of one or more parameter list entries, with each entry also being described with properties.
For example, the Keyboard property can be used to make the tag in the entry editable by device operators. Tags are assigned to parameter lists either automatically with the ad-hoc group used by default or with custom-defined tag groups. Working with tag groups has the advantage that the groups only need to be defined once in the project and can then be used in all the visualization objects used for the parameter list. In other words, this makes it easy to display various views of data.

Each tag can be assigned multiple times. The tags being shown can be of different data types, and can also be from a tag array.

If a keyboard is assigned to a parameter list entry, the keyboard will be shown at runtime and soon as the device user taps twice on the corresponding parameter list row. At that point, it will be possible to edit the tag value using the keyboard. Please note that values edited at runtime will be transferred directly to the PLC if the the PLC tag has been assigned an address.
If the Allow drag property for a parameter list object is enabled, it will be possible to scroll through the parameter list at runtime. There are additional parameter list functions available that can be used to sort a parameter list's entries. Using these functions always requires the use of a function key visualization object. A triggering element is not required.

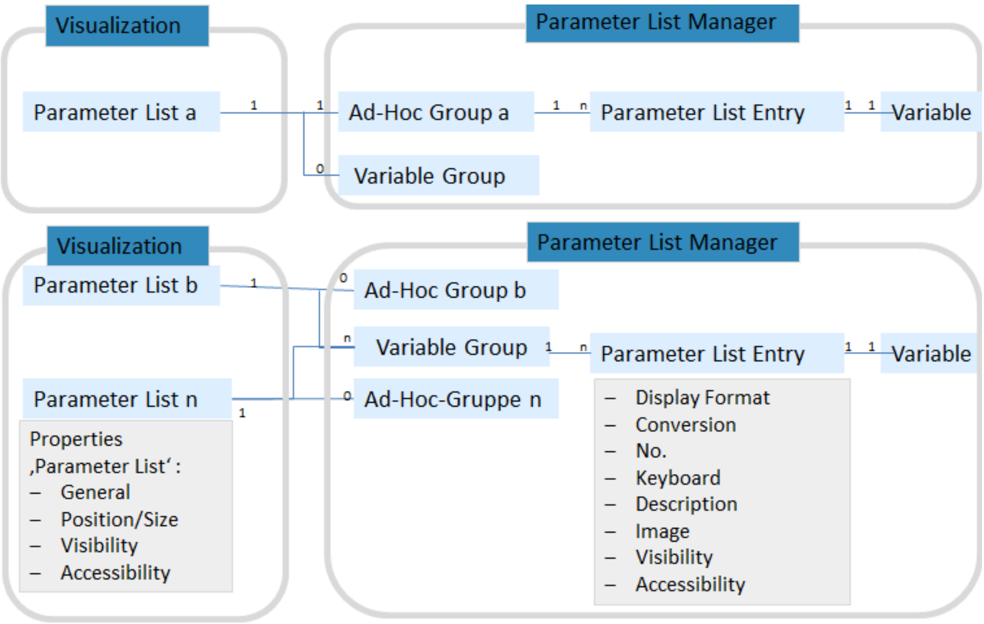


Figure 511: How parameter lists work: how the visualization object, the tag groups, and the various tags interact with each other

9.1 Using parameter lists to display values

In real-life applications, parameter list objects are used to subdivide large data lists into smaller lists and to display and edit them on device screens. One of the advantages of using this approach is that smaller lists keep things manageable for device operators.

The following steps show how to create a parameter list. The use case that follows describes a typical scenario, as well as the results expected on the device screen or computer.

Prerequisites

- A device must already be selected, e.g., XV-300-xxx
- A communication system must already be selected, e.g., XSOFT-CODESYS-3
- A screen must have already been created, e.g., <Start screen>.
- You must have already imported the tags you want to display and edit from the PLC or created them in *Project Tree\Tags*.

To create a parameter list, follow the steps below:

1. **Creating a parameter list visualization object**

   To create a parameter list visualization object, follow the steps below:

   - Create a new screen if necessary, e.g., <Sensors>.
   - Click on *Controls\Parameter List*.
   - Position the object
     Move the cursor to the screen and draw the area for the visualization object.
   - In the Properties 'Parameter List' pane, check to make sure that the Image, No., Description, and Value checkboxes are enabled. If they are not, the corresponding properties will not be shown at runtime.
   - Select the No Frame □ option.
   - Enable the Allow drag checkbox. This will make it possible to scroll through the parameter list visualization object at runtime by using a swipe gesture.
     These touch gestures can only be used with PTC devices such as the XV300 and XP500.
     This function is also supported by GALILEO Open applications.
   - Double-click on the parameter list visualization object.
   - The Extended properties "Parameter List" dialog box will open.
   - Open the States tab.
   - Select <Arial> as the font, <26> pt as the font size, and a shade of green as the text color.
   - Select the Color as background option and then select <transparent> as

the color.

- Click on OK.

2. **VariablenAssigning tags to a parameter list and displaying them**

To assign tags to a parameter list visualization object, they need to be assigned using tag groups. There are two types of tags groups: The first is the ad-hoc group, which is selected by default. Use the Ad-Hoc Group option if you want to enter and/or output tags in the project using solely the parameter list visualization object being configured.

On the other hand, select the Project Group option if you want to display various groupings of tag values for the device operator.

Adding project groups

You can create project groups by clicking on *Home\Parameter Lists* or directly with the parameter list visualization object that will be displaying the project groups at runtime.

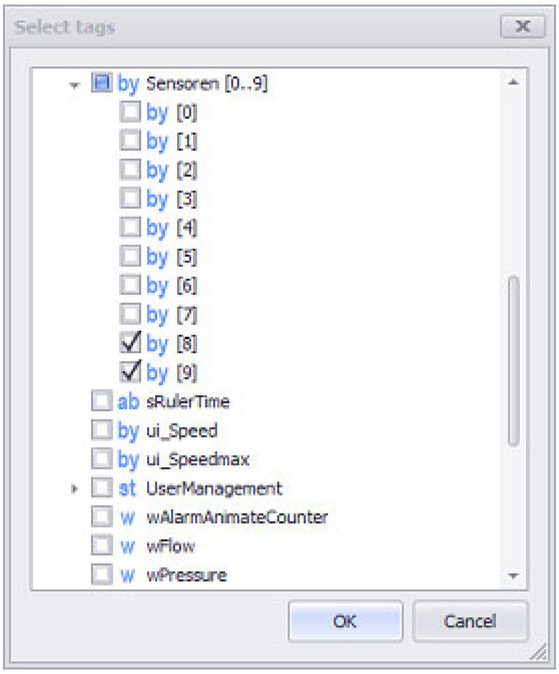
To create a project group using the parameter list visualization object, follow the steps below:

- Select the parameter list visualization object on the screen and click on the button in the Properties 'Parameter List' pane.
- The Tag Groups tab will open. The Ad-Hoc Group option will be selected by default.
- Click on Project Group.
- Click on the Add button and enter the name you want for the project group, e.g., <ColorSensor_1>.
- If you want to create several project groups, repeat the previous step, entering the names you want for the additional groups, e.g., <ColorSensor_2>, <ThicknessSensor>. If you create several project groups, they will be displayed on the device screen at runtime only if the corresponding checkboxes are enabled.

Assigning tags to the tag group

To create entries in a tag group and assign the tags you want, follow the steps below:

- Enable the Active checkbox for the tag group you want, e.g., <ThicknessSensor>. Make sure to disable the checkboxes for all other tag groups; otherwise, the buttons required to add tags to tag groups will remain grayed out.
- Click on and select the tags you want by enabling the corresponding checkboxes.



Select tags

- ☐ by Sensoren [0..9]
  - ☐ by [0]
  - ☐ by [1]
  - ☐ by [2]
  - ☐ by [3]
  - ☐ by [4]
  - ☐ by [5]
  - ☐ by [6]
  - ☐ by [7]
  - ☑ by [8]
  - ☑ by [9]
- ☐ ab sRulerTime
- ☐ by ui_Speed
- ☐ by ui_Speedmax
- ☐ st UserManagement
- ☐ w wAlarmAnimateCounter
- ☐ w wFlow
- ☐ w wPressure

Figure 512: Two tags – SensorParameters [6] and SensorParameters [7] – being assigned to a tag group, e.g., <ThicknessSensor>

- Click on OK.
- Check the Properties 'Parameter List' pane to make sure the Value checkbox is enabled.
- Click on OK.
- Click on the drop-down menu arrow for the Description column to enter a text of your choice for the tags. This text will be displayed on the device screen at runtime.
- Check the Properties 'Parameter List' pane to make sure the Description checkbox is enabled.

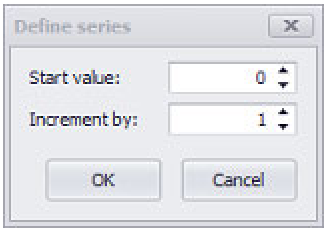
Define Series

By default, the list entries will be sorted in ascending order at runtime based on the value in the No. field. You can use the No. field as necessary to set up the list entry sorting you want. Either enter a number into every No. field or define series.

To define series, follow the steps below:

- For the first list entry, click on the No. column.
- While holding down the <Shift> key, click on the No. column for the last list entry.
- Multiple list entries will be selected in the No. column.
- Right-click to open the context menu and click on the <Define series> option.

The Define series dialog box will appear.



Define series

Start value: 0

Increment by: 1

- ▶ Enter the value with which the first list entry should start into the "Start value" field.
- ▶ Enter the increments for the series into the Increment by field.
- ▶ Click on OK.

Final check

- ▶ If you want to work with multiple project groups, use the procedure above to assign tags to all the project groups, e.g., ColorSensor_1 and ColorSensor_2
- ▶ Check to make sure that the Active checkbox is enabled for all the project groups you want displayed.
- ▶ Start the Inspector by pressing the F5 key.

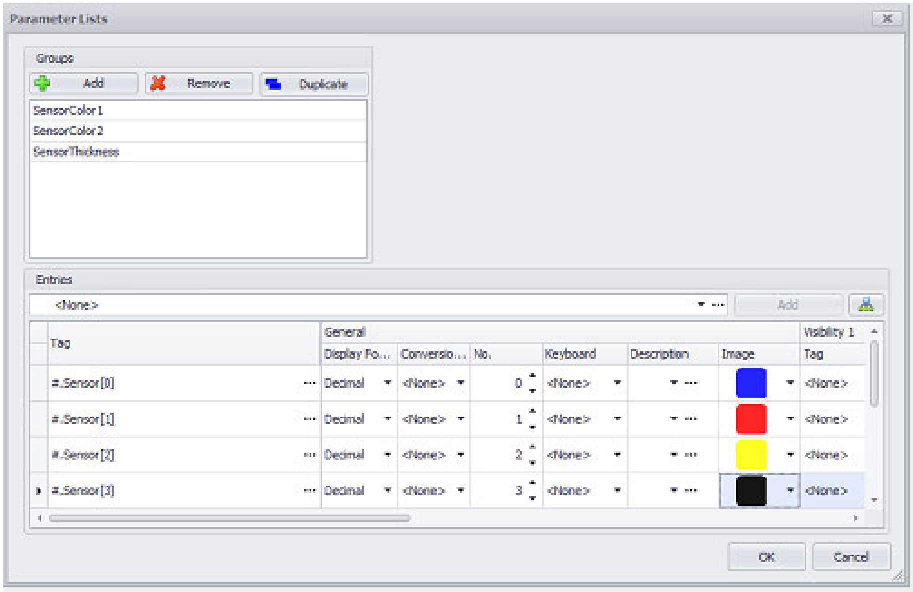
→ Once a project group is created, it will be available to all the parameter list visualization objects in the project. Also, a tag can be assigned to more than one project group.

→ The settings configured in a tag group, such as Description, Keyboard, and Image, will be linked to the corresponding parameter list entry and not to the tag itself!
This means that the settings for a tag, e.g., an image assigned to project group A, will not be automatically applied when the tag is added to another project group, e.g., project group B.

Assigning tags to a parameter list and displaying them

- ▶ Create three project groups named ThicknessSensor, ColorSensor_1, and ColorSensor_2 that will be used to group values when they are shown to the device operator.
- ▶ Add four entries to the ColorSensor_1 project group by clicking on the Add button. Assign a tag for each entry inColorSensor_1 as shown in the screenshot below.

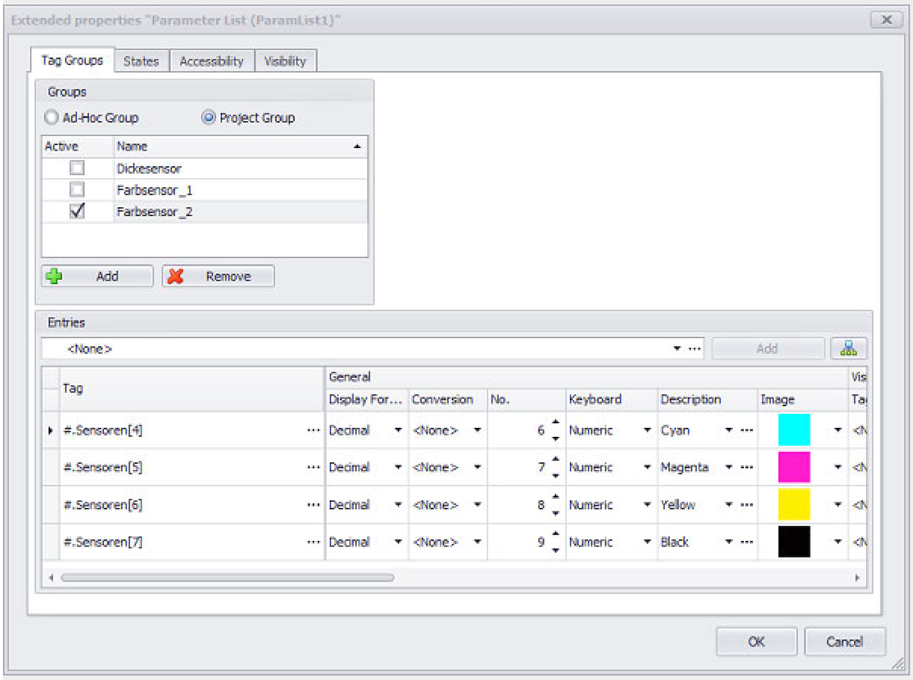
9. Parameter list
9.1 Using parameter lists to display values



SensorColor1
SensorColor2
SensorThickness

	Tag	General						Visibility 1
		Display Fo...	Conversio...	No.	Keyboard	Description	Image	Tag
	#.Sensor[0]	Decimal	<None>	0	<None>		🟦	<None>
	#.Sensor[1]	Decimal	<None>	1	<None>		🟥	<None>
	#.Sensor[2]	Decimal	<None>	2	<None>		🟨	<None>
▸	#.Sensor[3]	Decimal	<None>	3	<None>		⬛	<None>

- ▶ Repeat the last step for ColorSensor_2

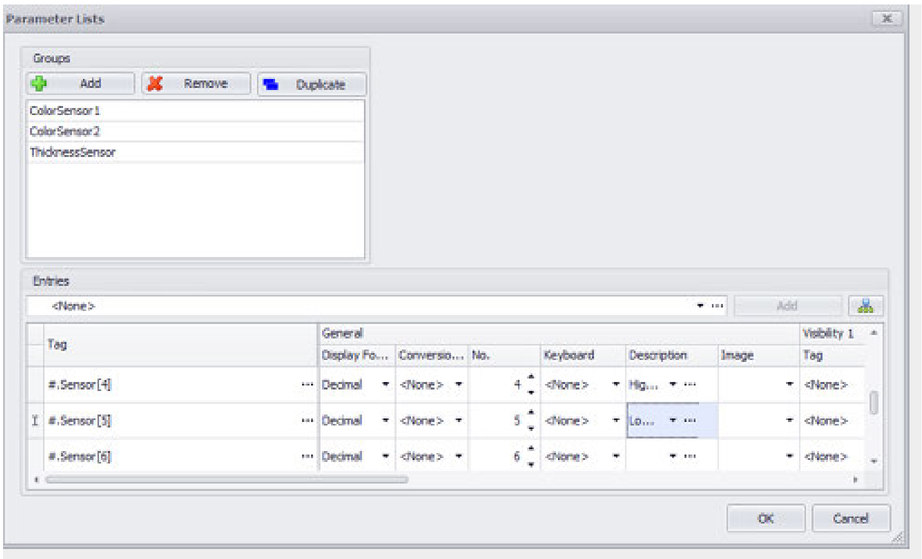


○ Ad-Hoc Group ● Project Group

Active	Name
□	Dickesensor
□	Farbsensor_1
☑	Farbsensor_2

	Tag	General						Vis
		Display For...	Conversion	No.	Keyboard	Description	Image	Ta
▸	#.Sensoren[4]	Decimal	<None>	6	Numeric	Cyan	🟦	<N
	#.Sensoren[5]	Decimal	<None>	7	Numeric	Magenta	🟪	<N
	#.Sensoren[6]	Decimal	<None>	8	Numeric	Yellow	🟨	<N
	#.Sensoren[7]	Decimal	<None>	9	Numeric	Black	⬛	<N

- ▶ Repeat the last step for ThicknessSensor.



ColorSensor1
ColorSensor2
ThicknessSensor

<None>

	General						Visibility 1
Tag	Display Fo...	Conversio...	No.	Keyboard	Description	Image	Tag
#.Sensor[4]	Decimal	<None>	4	<None>	Hig...		<None>
#.Sensor[5]	Decimal	<None>	5	<None>	Lo...		<None>
#.Sensor[6]	Decimal	<None>	6	<None>			<None>

3. **Displaying an image in the parameter list (optional)**

In certain cases, using images to identify the entries in a parameter list on the device screen at runtime may help device operators by making their work easier.

Prerequisites

The image files that you want to use must already be found in the `Project images` folder.

You must have already assigned a tag group containing tags to the visualization object.

For detailed information on how to add images to *Project Tree\Project images* and use them in a project, please refer to → section "Step 5: Adding an image", page 174.

To add images to the entries in a parameter list visualization object, follow the steps below:

- Select the parameter list visualization object.
- Click on the button in the Properties 'Parameter List' pane.

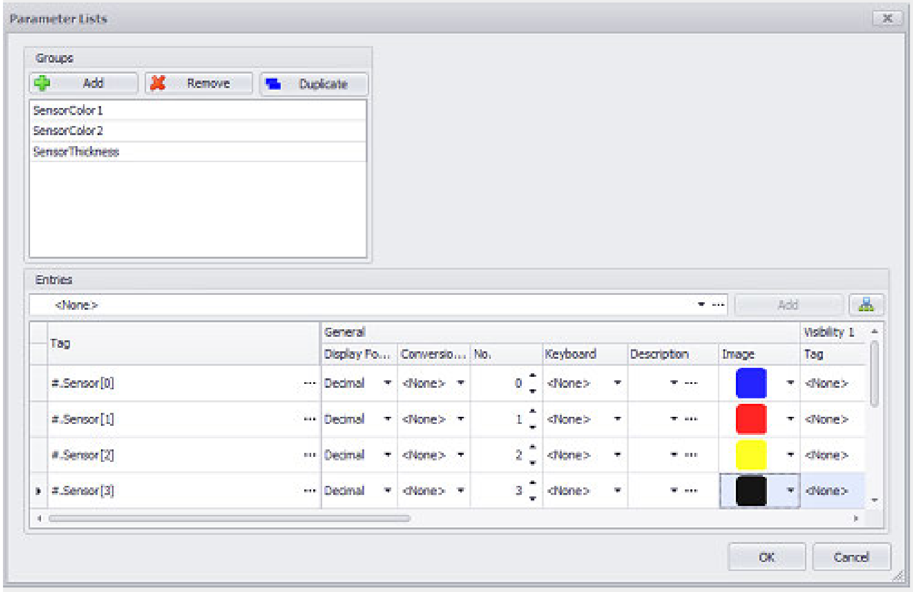
The Extended properties "Parameter list" dialog box will appear with the Tag Groups tab open.

- Click on the drop-down menu arrow under the Image column to select the image you want for each entry. To open the Project images folder, make sure to click on the arrow next to it and not on its name. For example, select the image in <Project images\Icons\sensorcolors\>cyan< *for the >SensorParameters[0]* entry.
- Check the Properties 'Parameter List' pane to make sure the Image checkbox is enabled.
- Start the Inspector by pressing the F5 key.

The parameter list with the image for the entry will be displayed.

Displaying an image in a parameter list

- You must have already saved the color images as bitmaps with the PNG image file format in the `Project images` folder.
- Click on the drop-down menu arrow under the Image column to select the image you want for each entry. To open the Project images folder, make sure to click on the arrow next to it and not on its name.



SensorColor1
SensorColor2
SensorThickness

	General						Visibility 1
Tag	Display Fo...	Conversio...	No.	Keyboard	Description	Image	Tag
#.Sensor[0]	Decimal	<None>	0	<None>		🟦	<None>
#.Sensor[1]	Decimal	<None>	1	<None>		🟥	<None>
#.Sensor[2]	Decimal	<None>	2	<None>		🟨	<None>
#.Sensor[3]	Decimal	<None>	3	<None>		⬛	<None>

- Repeat this step for ColorSensor_2

4. **Editable values in the parameter list (optional)**

In order to make it possible for operators to edit the values in a parameter list on the device screen at runtime, a keyboard needs to be assigned to the corresponding entries in the parameter list.

- Click on the relevant parameter list.
- Click on the button in the Properties 'Parameter List' pane.

The *Tag Groups* tab will open.

- Under the *Keyboard* column, select a keyboard (e.g., <Numeric>) for the entries that device operators should be able to edit at runtime.
- To assign the same keyboard to multiple entries, start by assigning the keyboard (e.g., <Numeric>) to the first entry, then double-click on the keyboard cell for that entry.
- Press <Ctrl+C> to copy the entry to the clipboard.
- Select the keyboard cells for the other entries you want. For example, by holding down the Shift key while clicking on the first and last cells.

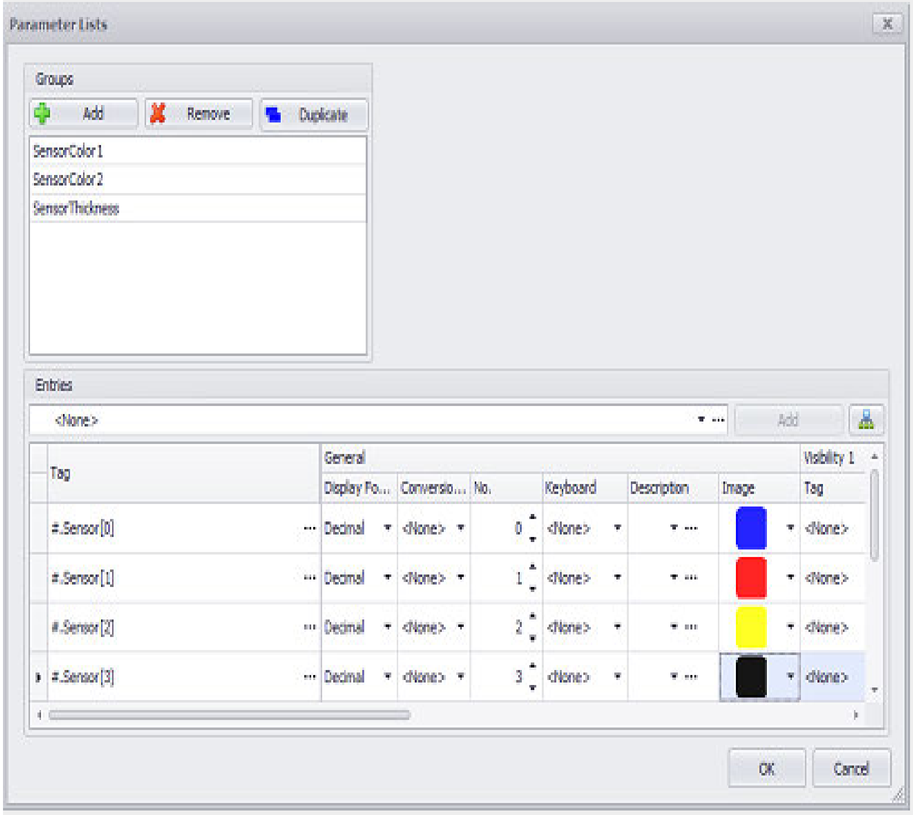
- Press <Ctrl+V> to copy the clipboard's contents to the selected cells.
- Start the Inspector by pressing the F5 key.

The tags' values will be displayed in the parameter list on the device screen. Double-clicking on an entry will open the configured keyboard, making it possible to edit the corresponding value.

Changes to tag values will be transmitted immediately to the PLC, provided that the modified value corresponds to a PLC tag.

How to make it possible to edit values in a parameter list

- In order to make it possible for device operators to edit the values in a parameter list on the device screen at runtime, it is necessary to specify a keyboard for every entry that should be editable. To do this, go to the Keyboard cell and select the Numeric keyboard option.



SensorColor1
SensorColor2
SensorThickness

Tag	General						Visibility 1
	Display Fo...	Conversio...	No.	Keyboard	Description	Image	Tag
#.Sensor[0]	Decimal	<None>	0	<None>		🟦	<None>
#.Sensor[1]	Decimal	<None>	1	<None>		🟥	<None>
#.Sensor[2]	Decimal	<None>	2	<None>		🟨	<None>
#.Sensor[3]	Decimal	<None>	3	<None>		⬛	<None>

- Repeat this step for **<ColorSensor_2>** and **<ThicknessSensor>**.

The current value identified with **<Cyan>** in the parameter list is assigned to the **<SensorParameters[0]>** tag. In turn, the **<SensorParameters[0]>** tag is assigned to tag address **<sensor_param>** in a PLC. If the device operator changes the value for **<Cyan>** in the parameter list entry at runtime, the PLC will immediately see the value as a a result of the change at the **<sensor_param>** tag address.

5. **Save ID in Tag**

   When the ID is saved in a tag, the PLC is directly notified of which entry is currently selected. This makes it possible to dynamically set and modify other tags in the PLC based on the selected entry.

   In order to pass the values from a parameter list to a PLC tag at runtime, the parameter list values must be linked to the PLC tags.

   ▶ Go to *Properties* pane, General section.

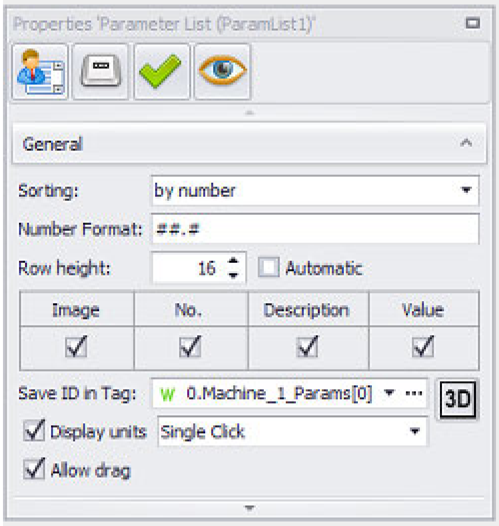
   ▶ Enable the Save ID in Tag option.

   ▶ Click on the drop-down menu arrow and select a PLC tag.

   Linking parameter list values to PLC tags

   ▶ In the General section of the *Properties* pane, enable the Save ID in Tag option.

   ▶ Click on the drop-down menu arrow and select the Machine_1.Param PLC tag you imported previously from the list of PLC tags.



Properties 'Parameter List (ParamList1)'

General

Sorting: by number

Number Format: ##.#

Row height: 16 □ Automatic

Image	No.	Description	Value
☑	☑	☑	☑

Save ID in Tag: W 0.Machine_1_Params[0]

☑ Display units Single Click

☑ Allow drag

   If the device operator changes the value of the tags at runtime, the PLC will immediately see the value as a a result of the change in the Machine_1.Param tag.

6. Add navigation functions

   a. → "First Line (Parameter list)", page 828 / → "Last Line", page 830

   b. → "Line up", page 834 / → "Line down", page 832

   c. → "Page up", page 837 / → "Page down", page 835

7. Adding sorting functions

   a. → "Sort by Description", page 841 / → "Sort by Number", page 843

   b. → "Sort ascend/descend", page 839

8. Showing the help window for the parameter list entry

a. **Showing the help for tags in a list**

The following example builds on the parameter list example. Before proceeding with this section, make sure you go through the parameter list sections and complete the corresponding exercise – please refer to → "Using parameter lists to display values ", page 787

Prerequisites
In order to be able to follow the descriptions below, you must have already created the <Sensors> screen, and the screen must contain the corresponding parameter list visualization object.

To add help information for entries in a parameter list, follow the steps below:

Example Show Tag Help

This example is shown in DemoApplication2.prj.

This example shows how to add help information to entries in a parameter list so that operators will be able to access this help. To do so, follow the steps below:

Create a function key visualization object

- Click on the parameter list visualization object.
- Move the cursor to the visualization object's top right corner.
- Move the cursor over the 💡 icon and let it stay there until the quick access menu appears.
- Select the <Show Tag Help> option in the quick access menu. A function key visualization object will be automatically created and linked to the corresponding function.

Configure the function key visually

To add an image to the button for the function key visualization object, follow the steps below:

- Go to the Position/Size section in the Properties 'Function Key' pane and enter <40> for the width and <40> for the height.
- Double-click on the function key to open the Extended properties "Function Key" dialog box.
- Select Icon as foreground in the States tab.
- In the field to the right (the one that says <None>), click on the drop-down menu arrow and select the image you want, e.g., *Project images\Icons\<help>*.
- Enable the Stretched checkbox.
- Select the Color as background option and then select

<Transparent>

▶ Click on OK.

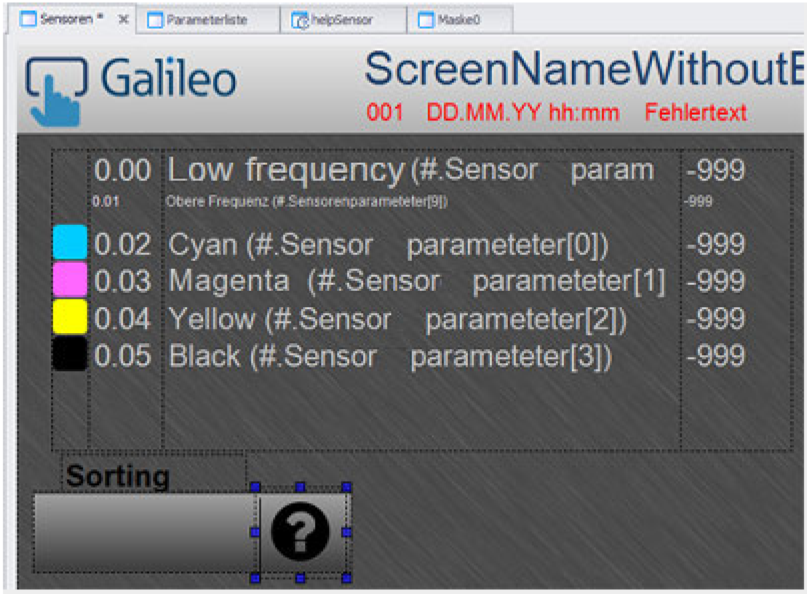
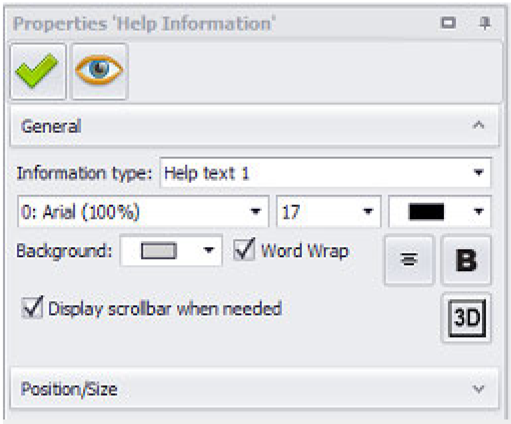

Sensoren *
Parameterliste
helpSensor
Maske0
Galileo
ScreenNameWithoutE
001 DD.MM.YY hh:mm Fehlertext
0.00 Low frequency (#.Sensor param -999
0.01 Obere Frequenz (#.Sensorenparameteter[9]) -999
0.02 Cyan (#.Sensor parameteter[0]) -999
0.03 Magenta (#.Sensor parameteter[1] -999
0.04 Yellow (#.Sensor parameteter[2]) -999
0.05 Black (#.Sensor parameteter[3]) -999
Sorting
?

Figure 513: Section from the <Sensors> screen with parameter list visualization object and selected function key visualization object with <Show Tag Help> function.

Create and configure a help screen

- Open the *Project Tree\Screens\Help Screens folder* and double-click on the folder to open the context menu and select the *Add new screen* option.
- Create a new help screen, e.g., helpSensor.
- In the Properties '<help screen>' pane, select Transparent as the <Background>.
- Click on *Controls\Help Information* and draw a Help information visualization object on the helpSensor help screen.
- In the Properties 'Help Information' pane, select Help text 1 using the <Information type> drop-down menu. You will be filling this help text with text in the tag help in a later step .
- In the Properties 'Help Information' pane, select Transparent as the <Background> and enable the Word Wrap and Display scrollbar when needed checkboxes.
- Click on *Controls\Help Information* and draw a Help Information visualization object on the helpSensor help screen.
- In the Properties 'Help Information' pane, select Image using the <Information type> drop-down menu. You will be selecting the corresponding image in the tag help in a later step .
- Click on OK.



Properties 'Help Information'

General

Information type: Help text 1

0: Arial (100%) | 17

Background: ☑ Word Wrap

☑ Display scrollbar when needed

Figure 514: Properties for the MotorHelp screen. This pane can be used to select which of the three pieces of help information should be shown and in which display area. The three available options are Help text 1, Help text 2, and Image.

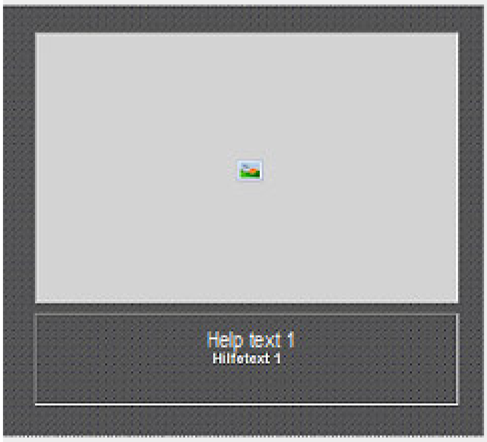


Figure 515: Section from the <helpSensor> screen

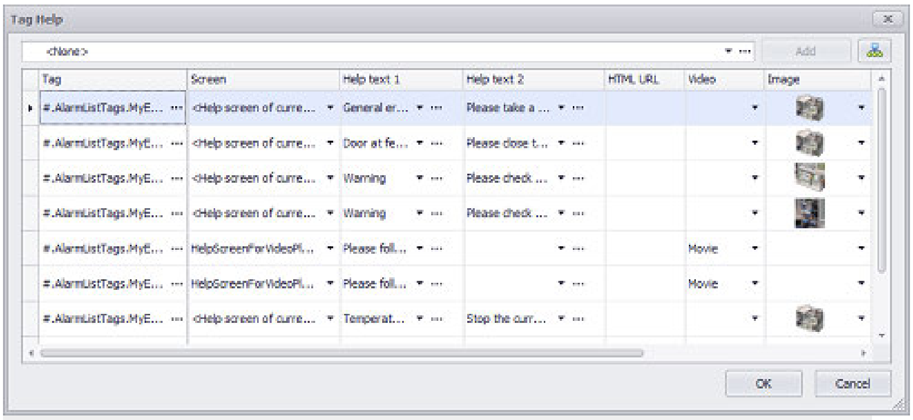
Fill out the help information for the help screen with content

By default, the <Help screen of current screen> will be used as the help screen. In other words, the help screen entered as such in the properties for the screen on which the visualization object with the tag is found is the help screen that will be used.

Two help texts and one text image can be entered for each entry in the tag help.

- Open the tag help by clicking on *Home\Tag Help*.
- Click on the drop-down menu arrow to select the tag to which you want to assign the help texts, e.g., <SensorParameters[0]>.
- In the Help text 1 column, enter the text you want, e.g., <Enter the value for cyan in the CMYK system.>
- In the Image column, select an image for the tag, e.g., *Project images\images\Machine\<service1>.*
- In the Background color column, select the <Transparent> option for all entries.

▶ In the Screen column for the relevant tags, select the help screen you created previously, e.g., <helpSensor>.
Alternatively, you can also use the default <Help screen of current screen> option. You will then need to use the Help screen field in the Properties 'Screen' pane to look for screen specified there and configure it accordingly.



Tag Help

<None>

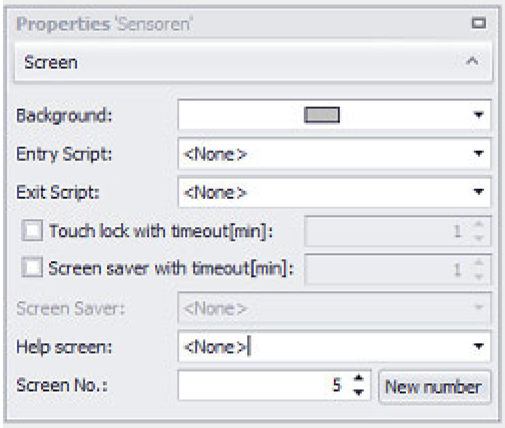
Tag	Screen	Help text 1	Help text 2	HTML URL	Video	Image
#.AlarmListTags.MyE...	<Help screen of curre...	General er...	Please take a ...			
#.AlarmListTags.MyE...	<Help screen of curre...	Door at fe...	Please close t...			
#.AlarmListTags.MyE...	<Help screen of curre...	Warning	Please check ...			
#.AlarmListTags.MyE...	<Help screen of curre...	Warning	Please check ...			
#.AlarmListTags.MyE...	HelpScreenForVideoPl...	Please foll...			Movie	
#.AlarmListTags.MyE...	HelpScreenForVideoPl...	Please foll...			Movie	
#.AlarmListTags.MyE...	<Help screen of curre...	Temperat...	Stop the curr...			

Figure 516: Tag Help dialog box; the contents for the three pieces of help information are defined here: help text 1, help text 2, and the help image shown in the help screen. In the screenshot, <helpSensor> has already been defined as the help screen for two tags

Join Helpscreen

The last step consists of going to the screen containing the visualization objects with the tag and setting the help screen you created as the corresponding help screen.

- ▶ Go to *Project Tree\Standard screens* and click on the screen with the tags, e.g., Sensors.
- ▶ In the Properties 'Sensors' pane, use the Help screen field to select the help screen you want, e.g., <helpSensor>.



Properties 'Sensoren'

Screen

Background:

Entry Script: <None>

Exit Script: <None>

□ Touch lock with timeout[min]: 1

□ Screen saver with timeout[min]: 1

Screen Saver: <None>

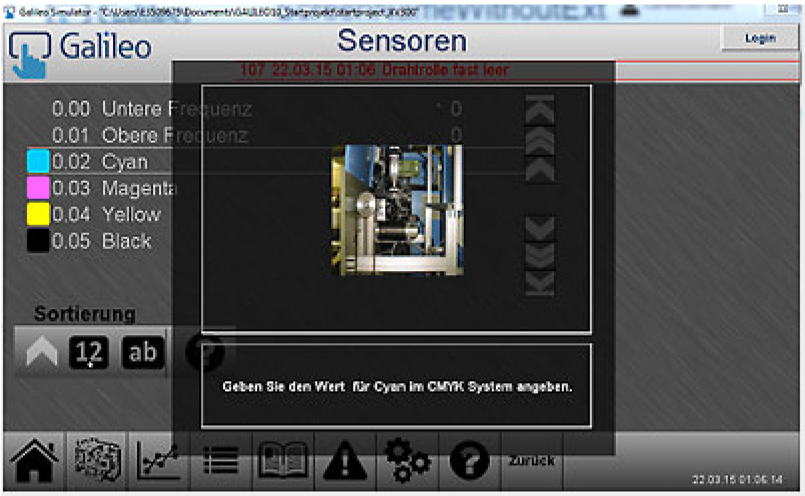
Help screen: <None>

Screen No.: 5

Figure 517: Properties pane for the Sensors screen with the helpSensor help screen as the <Help screen of current screen>

- ▶ Start the Inspector by pressing the F5 key.

The runtime result will look as shown below:



Sensoren

107 22.03.15 01:06 Drahtrolle fast leer

0.00 Untere Frequenz 0
0.01 Obere Frequenz 0
0.02 Cyan
0.03 Magenta
0.04 Yellow
0.05 Black

Geben Sie den Wert für Cyan im CMYK System angeben.

22.03.15 01:06:14

Device screen showing help screen after the ? button is tapped on.

Parameter list application example

Prerequisites

For this exercise, you must have already created the following tag:<Sensors[0..9]> of data type Byte Array.

Job

Say that you have a printing machine featuring three sensors. Two color sensors are used to ensure the required color quality, while the third sensor is used to measure the paper thickness.

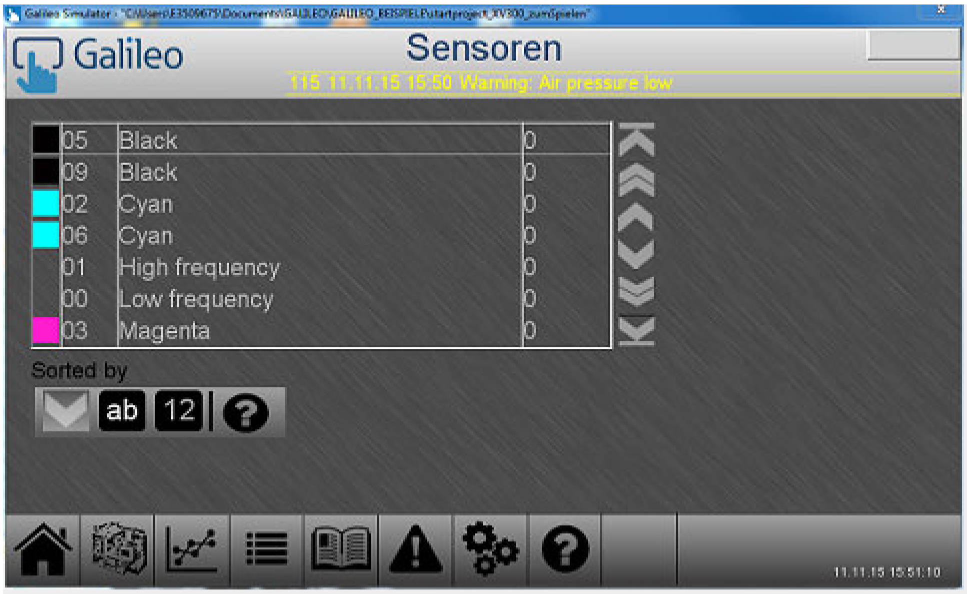
In real-life applications, sensors feature a large number of parameters. For simplification purposes, this example assumes that each color sensor only has four configurable parameters and that the thickness sensor only has two configurable parameters. In addition, it assumes that you want to give the device operation the option of displaying the parameters for all the sensors on the device screen and adjusting them as necessary.

In this example, you will be setting up controls with the following functions so that device operators can use them: switching between ascending and descending order, jumping to the first entry, jumping to the last entry, scrolling by line and by page, sorting by index or by description text. In addition, the tenth parameter list entry will be set up in such a way that device operators can access the help function for the parameter list.

If you follow steps 1 through 8 as listed above, the result on the device screen will look as shown below.

This example is shown in DemoApplication2.prj.

Result on device screen



Galileo Simulator - "C:\Users\E3509679\Documents\GALILEO\GALILEO_BEISPIEL\Startproject_XV300_zumSpielen"

Sensoren

115 11.11.15 15:50 Warning: Air pressure low

⬛	05	Black	0
⬛	09	Black	0
🟦	02	Cyan	0
🟦	06	Cyan	0
	01	High frequency	0
	00	Low frequency	0
🟪	03	Magenta	0

Sorted by

Figure 518: Application example: device screen with parameter list at runtime

9.2 Parameter List

Parameter lists are used to visualize the tags in one or more tag groups on a device screen.

The actual visualization object will be purely an output object unless it is configured with a keyboard. Using a keyboard will make it possible for device operators to enter a new value for the tags of data type BIT, BYTE, WORD, DWORD, QWORD, FLOAT, or STRING being shown. This keyboard will be displayed on the PC after clicking on the visualization object or on the device after tapping the object.

The maximum text length is limited to 256 characters, i.e., the number of characters that a tag of data type STRING with 256 bytes can show.

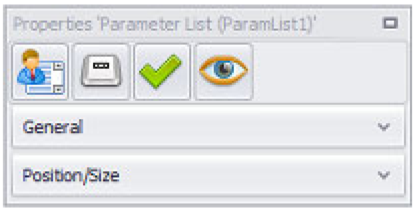
There are multiple navigation functions that can be applied to this visualization object.

It is possible to define two different states for this visualization object. This makes it possible to change the visualization object's appearance as soon as the object is enabled or disabled.

Prerequisites

In order to be able to follow the descriptions below, you must first have clicked on a parameter list visualization object on the current screen.

Controls tab



'Parameter List' properties

→ "Extended properties 'Parameter List'", page 804

→ "States", page 817

✓ → "Accessibility", page 818

→ "Visibility", page 818

→ section "General", page 812

Extended properties 'Parameter List'

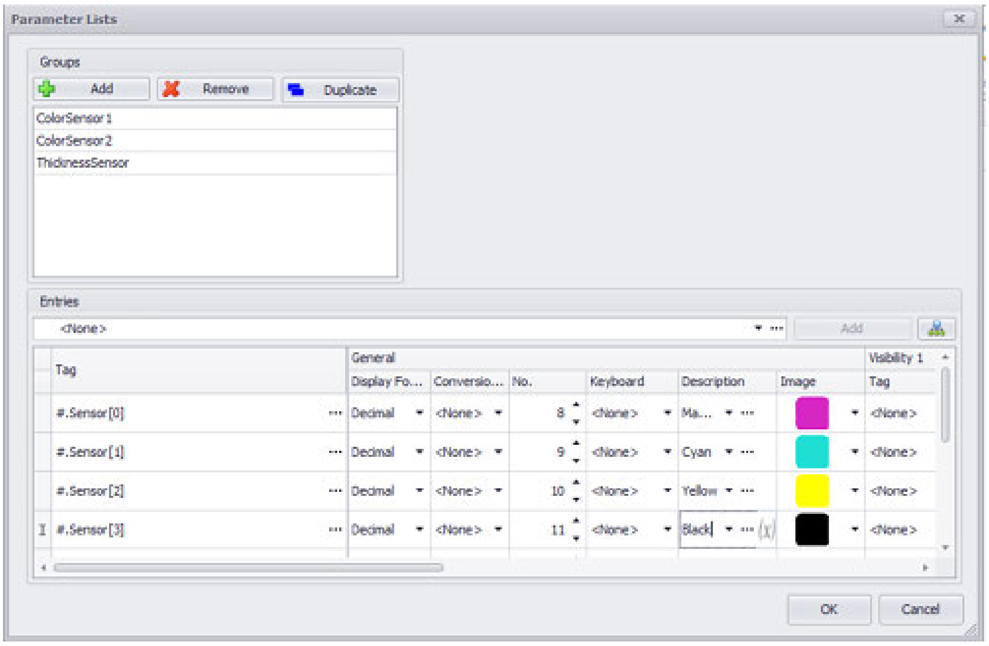
The settings in the Tag Groups tab are used to define which tag groups will be displayed in a parameter list. The Ad-Hoc Group option will be pre-selected by default. Before a tag can be displayed, it must be assigned to a tag group.

The Ad-Hoc Group option should be used when working with various groups is not particularly necessary in a project and the parameter list will always be used to display the exact same tags. In contrast to project groups, an ad-hoc group will only be available to the currently selected parameter list.

Project groups are used to make it possible to display various views of tags using a parameter list. Each project group will be available, together with its tags, to all the parameter lists in a project.

Properties defining the editability, accessibility, and visibility of individual parameter list entries at runtime are also available.

Controls\ Parameter List



	General						Visibility 1
Tag	Display Fo...	Conversio...	No.	Keyboard	Description	Image	Tag
#.Sensor[0]	Decimal	<None>	8	<None>	Ma...	🟪	<None>
#.Sensor[1]	Decimal	<None>	9	<None>	Cyan	🟦	<None>
#.Sensor[2]	Decimal	<None>	10	<None>	Yellow	🟨	<None>
#.Sensor[3]	Decimal	<None>	11	<None>	Black	⬛	<None>

Figure 519: Tag Groups tab in the Extended properties dialog box for a parameter list

9.2.0.1 Tag Groups of the Parameter List

Ad-Hoc Group

The Ad-Hoc Group option will be selected by default when a new parameter list is created. This ad-hoc group will only be available to the corresponding parameter list visualization object, i.e., other parameter list visualization objects will not be able to use it.

Project groups, on the other hand, behave differently: each project group only needs to be created once within a project, Once it is created, it will be available to all parameter list visualization objects so that it can be displayed as necessary.

Project Group

A project group only needs to be created once within a project. Once it is created, it will be available to all parameter list visualization objects so that it can be displayed as necessary. To access all project groups, you can also click on *Home\Parameter Lists*.

Active

When the checkbox for a project group is enabled, the project group will be displayed in the Parameter List visualization object at runtime.

→ If the parameter list is blank at runtime, this means that no project group has been enabled.

Name

All the groups defined in a project will be shown under the Name column. To add additional groups, press the F6 key.

Add

Adds a new project group.

Remove

Deletes the selected project group.

9.2.0.2 Entries of the Parameter List

Entries

Clicking on the drop-down menu arrow for the Entries field will allow you to select the tag that you want to add to the tag set for the selected group. Clicking on the Add button will add the selected tag, which will then be shown in the table below, in the Tag column.

Clicking on the icon will open the Select tags dialog box, which in turn will have all the tags in *Project Tree\Tags* available for selection.
To select one or more tags, simply enable the corresponding checkbox ☑.

Variable

The tags shown in the Tag column are the tags assigned to the selected group. Tags can be assigned to more than one group.

Button ...

Limits

Clicking on the ... icon will show the limits for the value tag in a separate dialog box. You can also reach this dialog box via *Project Tree\Tags\<Value tag>\Properties*.
When a tag is created, it will already be assigned limits based on the value range defined for it. This value range can be signed or unsigned and will depend on the relevant tag's data type.

The limits can be adjusted as necessary within the value range. In the case of bit tags, you will only be able to specify a default value.

For a description of the tag properties and limits, please refer to: → section "Numeric tag limits", page 563.

Tag Address

This field will only appear if you select a PLC tag, i.e., it will not appear for internal tags.
It will show the PLC address for the selected tag. The properties for PLC tags are described in greater detail in a different section – please refer to → section "Tag address", page 569.

If a single tag is transmitted to more than one communication, the corresponding settings will need to be configured in *Home\Gateway*. For detailed information on how to work with a gateway, please refer to → section "Using the gateway functionality to communicate with multiple PLCs", page 596.

Display Format

The following display formats are available for the tag value:

- Decimal, e.g., 248_{dec}
- Hexadecimal, e.g., $F8_{hex}$
- Octal, e.g., 370_{oct}
- Binary, e.g., $1111\ 1000_{bin}$
- BCD; $0000\ 0010\ 0100\ 1000_{BCD}$
- hh:mm / mm:ss

Conversion

Values do not necessarily have to be displayed or entered the way the PLC needs them or the way they are stored in the tags. The Value Conversion Manager can be used to convert tag values and assign a variety of units to them before they are shown by visualization objects. The way value conversion works is described in greater detail in a different section – please refer to → "Conversions", page 921.

No.

A number needs to be entered under the No. column. These numbers will be used to determine the order of the entries in the parameter list visualization object at runtime, making it possible to define series.

In order for this to occur, the No. checkbox must be enabled in the properties for the corresponding parameter list visualization object; please refer to → "Image / No. / Description / Value", page 814.

Keyboard

Clicking on the drop-down menu arrow will open a list of available keyboards. If a keyboard is selected, the visualization object will turn into a touch zone at runtime and the person using the device will be able to enter or change a value. If a keyboard is not selected, the visualization object will be a display object without a touch zone.

A numeric system keyboard can be assigned to numeric tags.

The following are the available numeric system keyboards:

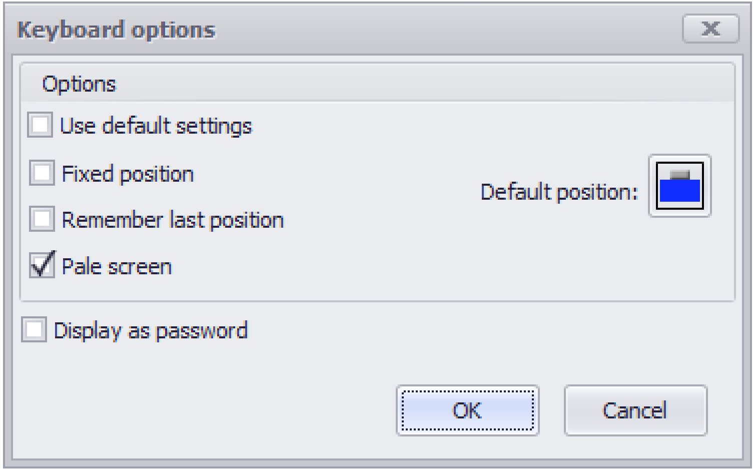
- inc/dec 1
- inc/dec 10
- Numeric
- Numeric Digit Min/Max
- Numeric Min/Max
- Numeric Password

- Select Numeric
- Select Numeric Digit

If you select Select Numeric or Select Numeric Digit as a keyboard, it will be possible to adjust the value with a function from the Keyboards or Characters/Stream characters group, e.g., with the Inc/dec 1 function. For a description of how to use functions, please refer to the → "Controlling functions with the PLC", page 602 section.

To configure the display options for the keyboard, click on the ⚙ icon:

Controls\...\Properties\Keyboard⚙


Keyboard options

Options

□ Use default settings

□ Fixed position

□ Remember last position

☑ Pale screen

Default position:

□ Display as password

Figure 520: Keyboard options dialog box

Use default settings: If this option is enabled, the other options will be hidden and the project settings will be applied instead.

Fixed Position If this option is enabled, an active keyboard cannot be moved around

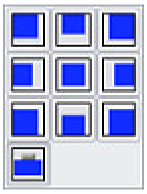
Remember last position If this option is enabled, the keyboard's position will be saved when closed.
The next time the same keyboard is opened, it will be automatically moved to this saved position.

Gray out screen If this option is enabled, the view behind the keyboard will be dimmed.

Default position: Used to specify the keyboard's default position relative to the object calling it.
The alignments available for selection include "top left," "center center," "bottom center," etc. plus "docked."
If you select the "docked" option, the Galileo runtime system will decide where the keyboard can best be displayed. If the "Remember last position" option is enabled, this setting will only be applied the very first time the keyboard is opened.

Default position

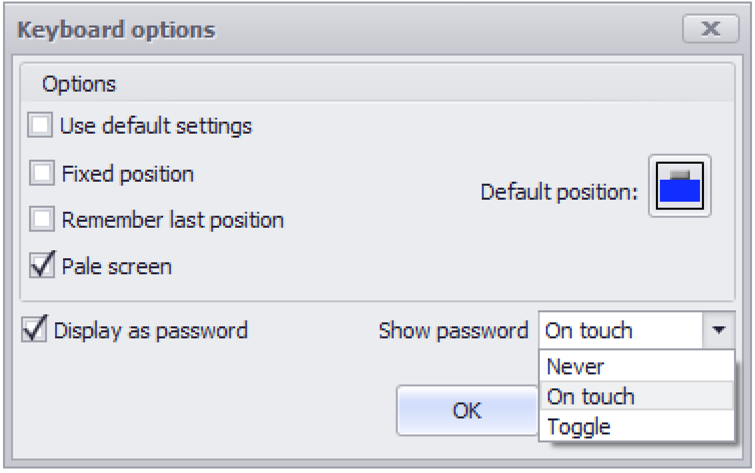
The keyboard will be positioned based on the option selected here. The default position on the screen can be defined by clicking inside the selection box as follows:



Top left ■ , top center ■ or top right ■ Center left ■ , center center ■ or center right ■ Bottom left ■ , bottom center ■ or bottom right ■ or ■ Docked on the button

Display as password: If this option is enabled, only "****" will be displayed instead of the actual value. The following additional options will be available when the option is enabled:

Controls\...\Properties\Keyboard



Keyboard options

Options

□ Use default settings

□ Fixed position

□ Remember last position

☑ Pale screen

Default position: ■

☑ Display as password

Show password: On touch

Figure 521: Keyboard options dialog box

Never There will be no way to show the entered password in cleartext at runtime

On touch As soon as a number or letter is entered, an 👁 icon will appear in the input field. If this icon is tapped, the password will be shown in cleartext.

As soon as the 👁 icon is no longer being touched, the password will be shown as "****" again.

Toggle As soon as a number or letter is entered, an 👁 icon will appear in the input field. If this icon is tapped, the password will be shown in cleartext. The password will continue to be shown in cleartext until the icon is tapped again.

→ A previously entered value cannot be shown in cleartext and will remain secret. Only current value input can be shown as cleartext.

For detailed information on how to put together keyboards made up of multiple individual function key visualization objects, please refer to the → section "User Defined Keyboards", page 316 section.

→ In order for this field to be enabled, a tag needs to be selected in the Tag field first.

Description

The text in the Description column will be shown at runtime in the parameter list visualization object. It can be used to describe the corresponding tag and/or what it does. Please note, however, that in order for the description text to be shown, the Description checkbox must be enabled in the properties for the corresponding parameter list visualization object; please refer to → "Image / No. / Description / Value", page 814.

Image

An icon can be displayed at runtime for each entry in a parameter list, This significantly expands the visual possibilities available when designing a visualization object.

In order to be able to take advantage of this option, the Image checkbox must be enabled in the properties for the corresponding parameter list visualization object; please refer to → "Image / No. / Description / Value", page 814.

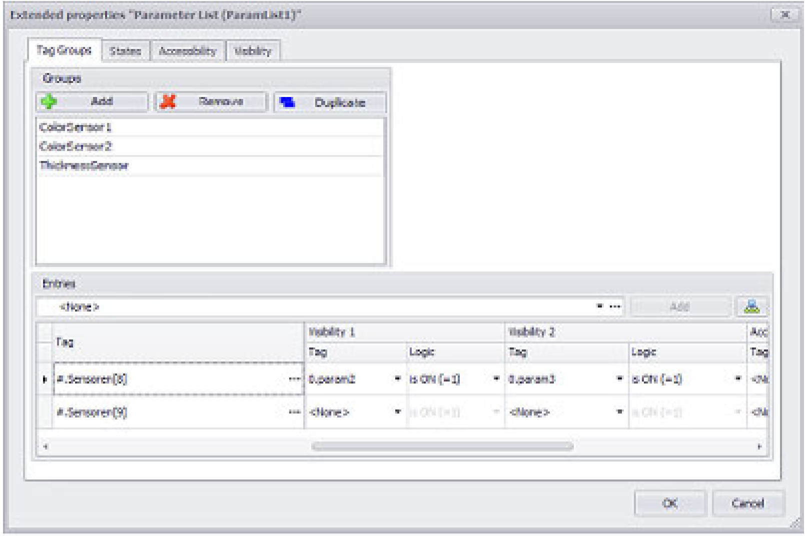
Click on the drop-down menu arrow under the Image column to select the image you want for each entry. Make sure that the image is found in*Project Tree\Media\Project images*. If it is not there, it will not appear in the drop-down menu.
For more detailed information on how to add an image to the project tree, please refer to → section "Step 5: Adding an image", page 174.

To remove a previously selected image so that it is no longer assigned to an entry, simply select the <None> option instead.

Visibility 1 and Visibility 2

You can enter a maximum of two conditions for tag states. These conditions will be OR'd, meaning that the corresponding tag will be INVISIBLE as soon as any of them are TRUE.

Controls\Parameter List\Properties\Tag Groups .



Extended properties "Parameter List (ParamList1)"

ColorSensor1
ColorSensor2
ThicknessSensor

Tag	Visibility 1		Visibility 2		Acc
	Tag	Logic	Tag	Logic	Tag
#.Sensoren[8]	0.param2	is ON (=1)	0.param3	is ON (=1)	<N
#.Sensoren[9]	<None>	is ON (=1)	<None>	is ON (=1)	<N

Figure 522: Extended properties dialog box for a parameter list showing the Tag Groups tab; in this screenshot, a project group containing one single tag has been created

→ Please note that the visibility conditions for the parameter list visualization object itself are not the same as the visibility conditions defined for the individual entries in a tag group!

These columns are used to define conditions under which the corresponding tag will not be visible in the parameter list. In contrast, the Visibility tag for the parameter list visualization object is used to define conditions under which the object itself will not be visible. This means, for example, that it is possible for the parameter list object to be displayed while a tag assigned to it is not.

Example for the Visibility 1 and Visibility 2 columns

The Sensors[8] tag will be shown if

param2 =0; param3=0;

The Sensors[8] tag will not be shown if

param2 =1; param3=0;

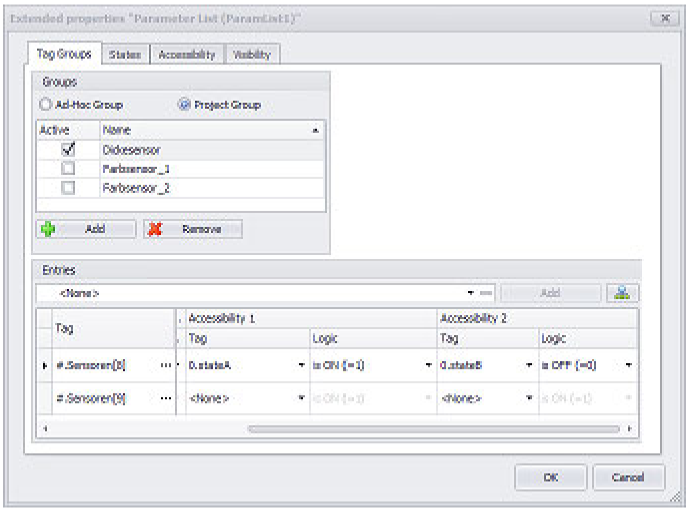
param2 =0; param3=1;

param2 =1; param3=1;

Accessibility 1 and Accessibility 2

You can enter a maximum of two conditions for the accessibility of individual tags. These conditions will be OR'd, meaning that the corresponding tag will be INVISIBLE as soon as any of them are TRUE.

Controls\Parameter List\Properties\Tag Groups .



○ Ad-Hoc Group ● Project Group

Active	Name
☑	Dickesensor
□	Farbsensor_1
□	Farbsensor_2

<None>

Tag	Accessibility 1		Accessibility 2	
	Tag	Logic	Tag	Logic
#.Sensoren[8]	0.stateA	is ON (=1)	0.stateB	is OFF (=0)
#.Sensoren[9]	<None>	is ON (=1)	<None>	is ON (=1)

→ Please note that the accessibility conditions for the parameter list visualization object itself are not the same as the accessibility conditions defined for the individual entries in a tag group.

These columns are used to define conditions under which the corresponding tag will not be accessible in the parameter list. In contrast, the Accessibility tag for the parameter list visualization object is used to define conditions under which the object itself will not be accessible. This means, for example, that it is possible for one of the tags in a tag group not to be accessible even when the corresponding parameter list object itself is accessible.

Example for the Accessibility 1 and Accessibility 2 columns

The Sensors[8] tag from the example above will be accessible if

stateA =0; stateB=1;

The Sensors[8] tag will not be accessible if

stateA =1; stateB=1;

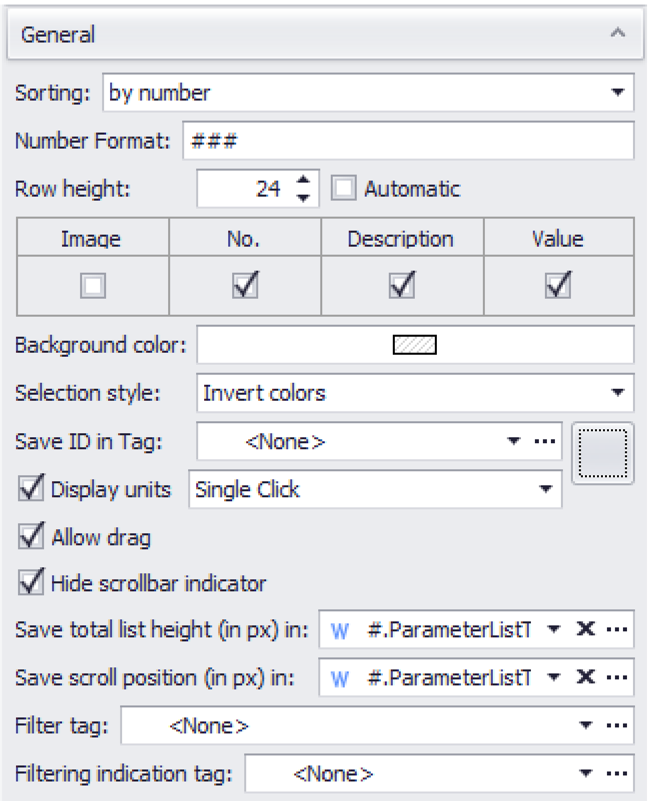
stateA =0; stateB=0;

stateA =1; stateB=0;

General

The General section can be used to configure all the properties that affect how the visualization object will be displayed, with the exception of its size and position.

Controls\Parameter List\General



General

Sorting: by number

Number Format: ###

Row height: 24 □ Automatic

Image	No.	Description	Value
□	☑	☑	☑

Background color: ▨

Selection style: Invert colors

Save ID in Tag: <None>

☑ Display units Single Click

☑ Allow drag

☑ Hide scrollbar indicator

Save total list height (in px) in: W #.ParameterListT

Save scroll position (in px) in: W #.ParameterListT

Filter tag: <None>

Filtering indication tag: <None>

Figure 523: Window Properties of the 'Parameter List'

Sorting

The Sorting field is used to specify the order in which the parameter list entries will be shown on the device screen.

- by numbers
  The parameter list entries will be shown in ascending order by default. This order will be based on the values in the No. column in the Tag Groups tab.
- by description
  The parameter list entries will be shown in ascending alphabetical order by default. This order will be based on the values in the Description column.

The Sort Descend , Sort Ascend, Sort by Number, and Sort by Description functions can also be used to change the way in which entries are sorted. For a description of these functions, please refer to the → " Parameter List", page 804 section.

Number Format

The Number Format field can be used to define the output format for the ID in the *No.* column of the parameter list object shown on the device screen at runtime. Each "#" character in this field is a placeholder for one number. , ' / - |

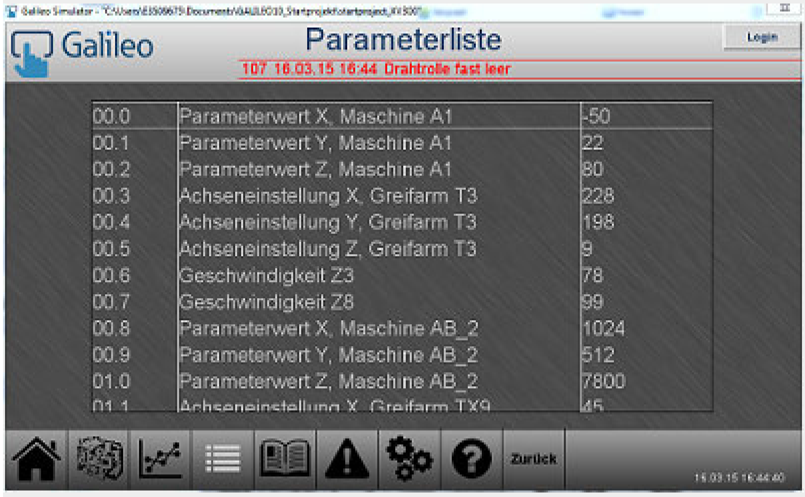
If the number has fewer decimal places than the defined format, the places are filled with leading zeros.

If the number has more decimal places than the defined format, the additional decimal places of the number are truncated.

Example Number Format

Number Format: ##.#

How the IDs for parameter list entries 0001 to 0100 will be displayed at runtime:



107 16.03.15 16:44 Drahtrolle fast leer

00.0	Parameterwert X, Maschine A1	-50
00.1	Parameterwert Y, Maschine A1	22
00.2	Parameterwert Z, Maschine A1	80
00.3	Achseneinstellung X, Greifarm T3	228
00.4	Achseneinstellung Y, Greifarm T3	198
00.5	Achseneinstellung Z, Greifarm T3	9
00.6	Geschwindigkeit Z3	78
00.7	Geschwindigkeit Z8	99
00.8	Parameterwert X, Maschine AB_2	1024
00.9	Parameterwert Y, Maschine AB_2	512
01.0	Parameterwert Z, Maschine AB_2	7800
01.1	Achseneinstellung X, Greifarm TX9	45

Row height

Used to define the row height in pixels independently of the selected font size. If the row height is smaller than the font size, the text will be cut at the top and bottom.

☑ Automatic adaptation

Automatic

If this option is enabled ☑, the row height will be automatically calculated in such a way that it will be tall enough for showing all the entries in the list. This can come in handy, since you can use different font sizes for the "Enabled" and "Disabled" states.

Image / No. / Description / Value

The checkboxes in this section can be enabled in order to specify which of these four possible parameter list columns will be shown on the device screen at runtime. The corresponding contents are assigned to the parameter list object in the Parameter List Manager.

Controls\Parameter List\Properties\General

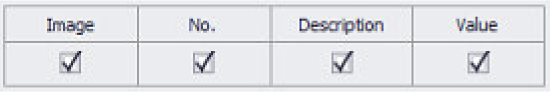
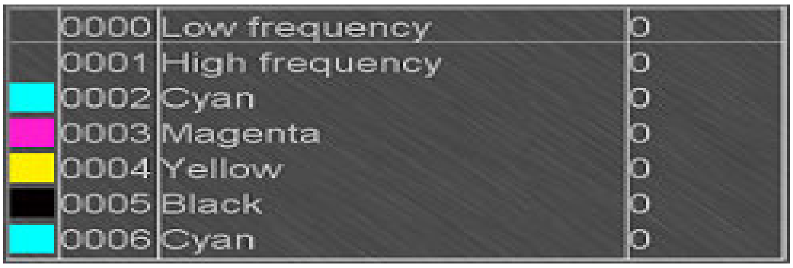


Image	No.	Description	Value
☑	☑	☑	☑

If all four checkboxes are enabled, the corresponding parameter list will be displayed on the device screen as shown below:

Display on device screen



	0000	Low frequency	0
	0001	High frequency	0
🟦	0002	Cyan	0
🟪	0003	Magenta	0
🟨	0004	Yellow	0
⬛	0005	Black	0
🟦	0006	Cyan	0

Background color

A color can be selected as the background for the list.

You can change the color if you want. The following options are available for this purpose:

- Transparent
- Standard Colors
- More Colors...

The most recently used colors will be shown (user-specific).

For detailed information on the available colors, please refer to the following section:
→ section "Color choice - Object's Color Palette", page 234

Selection style

The appearance can be defined for each list.

Invert colors and frames are available.

Depending on the selection, the active line on the device is either inverted, i.e. differently colored text line background or a frame line at the top and bottom marks the active line.

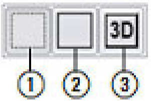
Save ID in Tag

If this checkbox is enabled, the ID of the entry selected at runtime will be stored in the selected tag. As soon as the checkbox is enabled, a tag must be selected from the drop-down menu – otherwise, an error message will appear during compilation (<There is no defined tag for storing the ID.>).

→ The actual ID will be the value passed. The value displayed in the list may be different from the ID if a number format is being used for the display.

Frame 3D

The visualization object can be displayed with any one of three different designs.



① no frame

② with frame

③ 3D frame

Display units

If this checkbox ☑ is enabled, the units for the tags will be displayed as well.

Units are configured in the Conversion Manager. A conversion group can be assigned to each entry so that units will be linked to the tag display.

Selection mode on panel Single Click

The following modes can be adjusted for visualization objects with a keyboard:

- Single Click
  Tapping once on a list item or a button on the device screen will activate the selection.
- Delayed
  This option is ideal for implementing a safety function designed to protect against accidental actions. In order for a button on the device screen to be activated, it will first have to be pressed and held (long press) for the configured delay time. By default, this delay time will be 0.5 s, but it can be changed in *Home\Settings\Runtime tab\Touch Settings*.
- Double Click
  Tapping twice on a list item or button on the device screen will select the item / activate the button. By default, the double-click speed will be 0.5 s for a double-click, but it can be changed in*Home\Settings\Runtime tab\Touch Setting*.

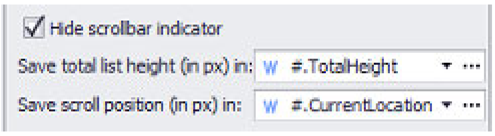
Allow drag

These touch gestures can only be used with PTC devices such as the XV300 and XP500.
This function is also supported by GALILEO Open applications.
If this checkbox is enabled, it will be possible to scroll through the entries in the list on the device screen at runtime. To do so, the person using the device will need to swipe over the list in the desired scrolling direction.

Hide scrollbar indicator

In order to work with objects for which a list is stored (parameter lists, alarm lists, recipes, and containers) more efficiently, you can save the position in the list.

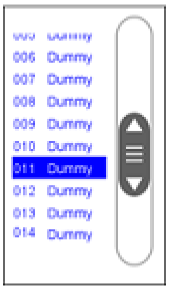
☑ If the checkbox is enabled, the following settings for the list object will be hidden.



☑ Hide scrollbar indicator
Save total list height (in px) in: W #.TotalHeight
Save scroll position (in px) in: W #.CurrentLocation

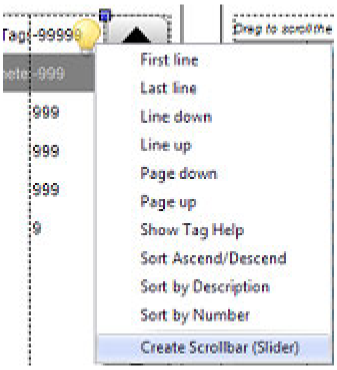
Save list height (in px) in:	Saves the total list height (total height with all items added up) in the specified tag
Save scroll position (in px) in:	Saves the current scroll position within the total list height in the specified tag.

Together with Save list height (in px) in, this tag can be used, for example, in order to use the slider in a slider object as a scrollbar in the list.



To make it easier to create this slider object, you can follow the steps below:

1. Assign tags to the Save list height (in px) in and Save scroll position (in px) in settings
2. In the screen designer, move the cursor to the upper right corner of the object. The 💡 icon will appear
3. Click on the 💡 icon and select the Generate scrollbar (slider) option in the drop-down menu



4. A slider object will be created and the tags will already be assigned correctly.

Filter tag

Specification of a tag of the type String. Together with the special function Filter the parameter list can then be filtered according to the content of the string variable. Only the column Description is considered for the filter.

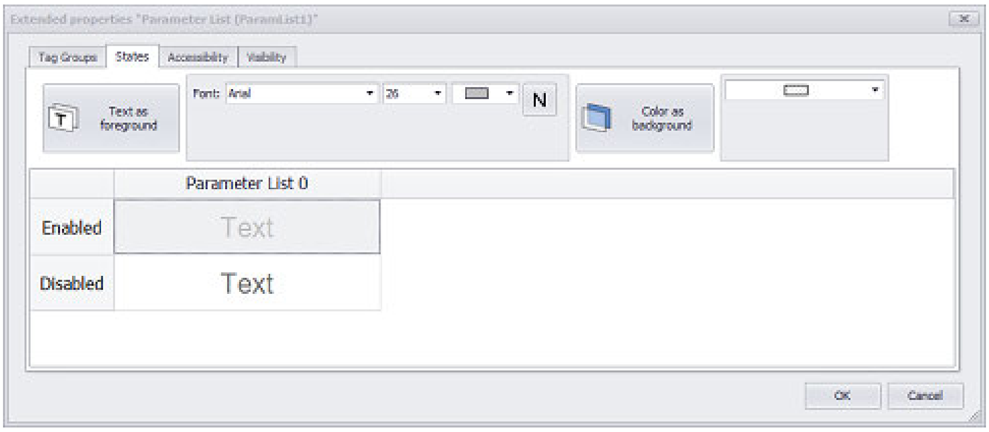
Filtering-indication-tag

Specification of a tag of the type bit. If filtering of the parameter list is active, the bit is set to **1**. Otherwise it has the value 0.

States

The states available for parameter list objects are Enabled and Disabled.

Controls\ Parameter List


Extended properties "Parameter List (ParamList1)"

Font: Arial 26

	Parameter List 0
Enabled	Text
Disabled	Text

Figure 524: Extended properties "Parameter List" dialog box, States tab

Enabled/ Disabled

The conditions used to determine whether a parameter list will be Enabled or Disabled are defined under the Accessibility tab. If these accessibility conditions are met, the parameter list will be enabled and will be displayed on the device screen using the settings for the Enabled state. If the conditions are not met, the parameter list will be disabled and will be displayed on the device screen using the settings for the Disabled state.

The basic configuration parameters for the visualization object's foreground and background are described in detail in the following section: Defining text series

Accessibility

→ Please note that the accessibility conditions for the parameter list itself are not the same as the accessibility conditions defined for the individual entries in a tag group!

The settings in this tab are used to define the conditions under which the visualization object will not be accessible. In contrast, the settings in the Accessibility columns in the *Tag Groups* tab are used to define the conditions under which the specific tags in the parameter list will not be accessible. This means, for example, that it is possible for the parameter list object to be accessible while a tag assigned to it is not.

For detailed information on the object's ✓ accessibility, please refer to: → section "Accessibility", page 258

Visibility

→ Please note that the visibility conditions for the parameter list visualization object itself are not the same as the visibility conditions defined for the individual entries in a tag group!

The settings in this tab are used to define the conditions under which the visualization object will not be visible. In contrast, the settings in the Parameter List

Manager and in the Tag Groups tab are used to define the conditions under which the specific tags in the parameter list will not be visible. This means, for example, that it is possible for the parameter list object to be displayed while a tag assigned to it is not.

For detailed information on the object's 👁 visibility, please refer to: → section "Visibility", page 259

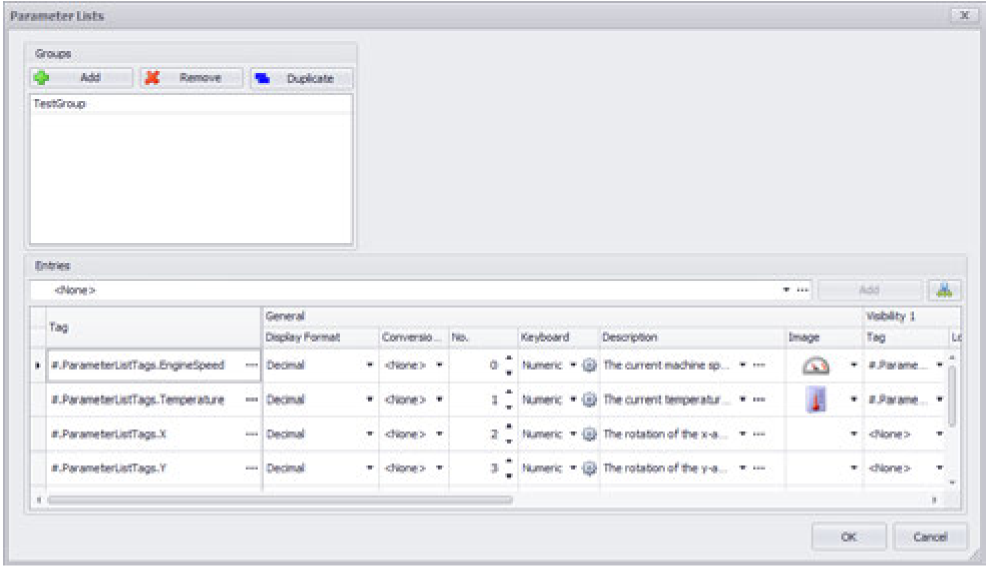
9.3 Parameter Lists

Parameter lists tag objects can be used to group tags and make these groups available to all parameter list visualization objects within a project so that they can be displayed.
Please note that unlike "parameter list" objects, "parameter lists" objects are not a visualization object.

Groups, also referred to as "project groups," are used to make it possible to display various views of tags using a parameter list. Each project group will be available, together with its tags, to all the parameter lists in a project.

Properties defining the editability, accessibility, and visibility of individual parameter list entries at runtime are available.

Home\Project Configuration\ Parameter Lists



	Tag	General						Visibility 1	
		Display Format	Conversio...	No.	Keyboard	Description	Image	Tag	Lc
▸	#.ParameterListTags.EngineSpeed	Decimal	<None>	0	Numeric	The current machine sp...		#.Parame...	
	#.ParameterListTags.Temperature	Decimal	<None>	1	Numeric	The current temperatur...		#.Parame...	
	#.ParameterListTags.X	Decimal	<None>	2	Numeric	The rotation of the x-a...		<None>	
	#.ParameterListTags.Y	Decimal	<None>	3	Numeric	The rotation of the y-a...		<None>	

Figure 525: Window Parameter Lists

Groups

A project group only needs to be created once within a project. Once it is created, it will be available to all parameter list visualization objects so that it can be displayed as necessary. To access all project groups, you can also click on *Controls\Properties 'Parameter List'*. Up to 65535 groups are allowed.

Add

Adds a new group

Remove

Selected group will be deleted.

Duplicate

The selected group will be added as a copy in its entirety. The group copy number will be incremented in the suggested name.

Entries

Clicking on the drop-down menu arrow for the Entries field will allow you to select the tag that you want to add to the tag set for the selected group. Clicking on the Add button will add the selected tag, which will then be shown in the table below, in the Tag column.

Clicking on the icon will open the Select tags dialog box, which in turn will have all the tags in *Project Tree\Tags* available for selection.
To select one or more tags, simply enable the corresponding checkbox ☑.

Tag

The tags shown in the Tag column are the tags assigned to the selected group. Tags can be assigned to more than one group. Each group can have a maximum of 65,535 tags.

Button ...

Limits

Clicking on the ... icon will show the limits for the value tag in a separate dialog box. You can also reach this dialog box via *Project Tree\Tags\<Value tag>\Properties*. When a tag is created, it will already be assigned limits based on the value range defined for it. This value range can be signed or unsigned and will depend on the relevant tag's data type.

The limits can be adjusted as necessary within the value range. In the case of bit tags, you will only be able to specify a default value.

For a description of the tag properties and limits, please refer to: → section "Numeric tag limits", page 563.

Tag Address

This field will only appear if you select a PLC tag, i.e., it will not appear for internal tags.
It will show the PLC address for the selected tag. The properties for PLC tags are described in greater detail in a different section – please refer to → section "Tag address", page 569.

If a single tag is transmitted to more than one communication, the corresponding settings will need to be configured in *Home\Gateway*. For detailed information on how to work with a gateway, please refer to → section "Using the gateway functionality to communicate with multiple PLCs", page 596.

Display Format

The following display formats are available for the tag value:

- Decimal, e.g., 248_{dec}
- Hexadecimal, e.g., $F8_{hex}$
- Octal, e.g., 370_{oct}
- Binary, e.g., $1111\ 1000_{bin}$
- BCD; $0000\ 0010\ 0100\ 1000_{BCD}$
- hh:mm / mm:ss

Conversion

Values do not necessarily have to be displayed or entered the way the PLC needs them or the way they are stored in the tags. The Value Conversion Manager can be used to convert tag values and assign a variety of units to them before they are shown by visualization objects. The way value conversion works is described in greater detail in a different section – please refer to → "Conversions", page 921.

No.

By default, a value of 0 will be entered into the <No.> column as soon as the first tag is added, with the value being incremented by one for every new tag.

By default, the parameter list entries will be sorted in ascending order at runtime based on the value in the No. field.

In order for this to occur, the No. checkbox must be enabled in the properties for the corresponding parameter list visualization object.

You can change the order in which entries will be displayed by changing the values in the No. column accordingly. GALILEO supports the use of series. For more detailed information on series, please refer to the → section "Define Series", page 789 section.

Keyboard

Clicking on the drop-down menu arrow will open a list of available keyboards. If a keyboard is selected, the visualization object will turn into a touch zone at runtime and the person using the device will be able to enter or change a value. If a keyboard is not selected, the visualization object will be a display object without a touch zone.

A numeric system keyboard can be assigned to numeric tags.

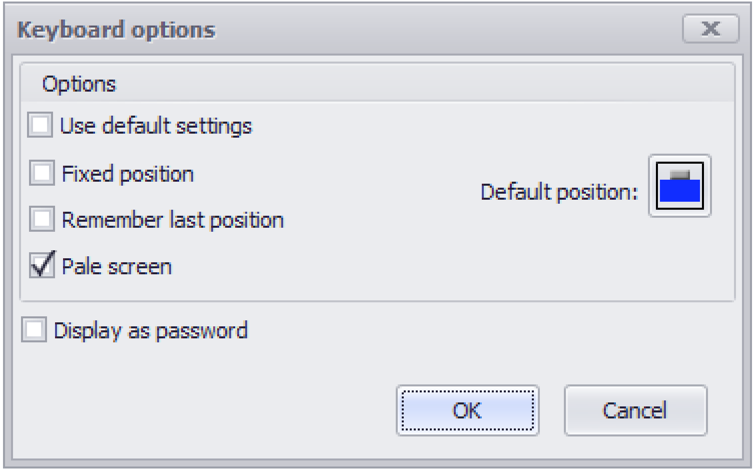
The following are the available numeric system keyboards:

- inc/dec 1
- inc/dec 10
- Numeric
- Numeric Digit Min/Max
- Numeric Min/Max
- Numeric Password
- Select Numeric
- Select Numeric Digit

If you select Select Numeric or Select Numeric Digit as a keyboard, it will be possible to adjust the value with a function from the Keyboards or Characters/Stream characters group, e.g., with the Inc/dec 1 function. For a description of how to use functions, please refer to the → "Controlling functions with the PLC", page 602 section.

To configure the display options for the keyboard, click on the ⚙ icon:

Controls\...\Properties\Keyboard⚙



Keyboard options

Options

□ Use default settings

□ Fixed position

□ Remember last position

☑ Pale screen

Default position:

□ Display as password

Figure 526: Keyboard options dialog box

Use default settings: If this option is enabled, the other options will be hidden and the project settings will be applied instead.

Fixed Position If this option is enabled, an active keyboard cannot be moved around

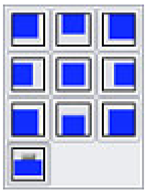
Remember last position If this option is enabled, the keyboard's position will be saved when closed.
The next time the same keyboard is opened, it will be automatically moved to this saved position.

Gray out screen If this option is enabled, the view behind the keyboard will be dimmed.

Default position: Used to specify the keyboard's default position relative to the object calling it.
The alignments available for selection include "top left," "center center," "bottom center," etc. plus "docked."
If you select the "docked" option, the Galileo runtime system will decide where the keyboard can best be displayed. If the "Remember last position" option is enabled, this setting will only be applied the very first time the keyboard is opened.

Default position

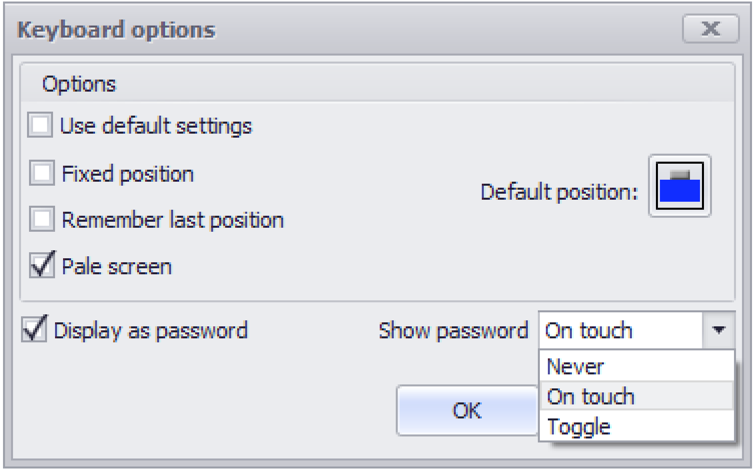
The keyboard will be positioned based on the option selected here. The default position on the screen can be defined by clicking inside the selection box as follows:



Top left ■, top center ■ or top right ■Center left ■, center center ■ or center right ■ Bottom left ■, bottom center ■ or bottom right ■ or ■ Docked on the button

Display as password: If this option is enabled, only "****" will be displayed instead of the actual value. The following additional options will be available when the option is enabled:

Controls\...\Properties\Keyboard



Keyboard options

Options

☐ Use default settings

☐ Fixed position

☐ Remember last position

☑ Pale screen

Default position: ■

☑ Display as password

Show password: On touch

Figure 527: Keyboard options dialog box

Never There will be no way to show the entered password in cleartext at runtime

On touch As soon as a number or letter is entered, an ◉ icon will appear in the input field. If this icon is tapped, the password will be shown in cleartext.

As soon as the ◉ icon is no longer being touched, the password will be shown as "****" again.

Toggle As soon as a number or letter is entered, an ◉ icon will appear in the input field. If this icon is tapped, the password will be shown in cleartext. The password will continue to be shown in cleartext until the icon is tapped again.

→ A previously entered value cannot be shown in cleartext and will remain secret. Only current value input can be shown as cleartext.

For detailed information on how to put together keyboards made up of multiple individual function key visualization objects, please refer to the → section "User Defined Keyboards", page 316 section.

→ In order for this field to be enabled, a tag needs to be selected in the Tag field first.

Description

The text in the Description column will be shown at runtime in the parameter list visualization object. It can be used to describe the corresponding tag and/or what it does. Please note, however, that in order for the description text to be shown, the Description checkbox must be enabled in the properties for the corresponding parameter list visualization object; please refer to → "Description", page 825.

Image

An icon can be displayed at runtime for each entry in a parameter list, This significantly expands the visual possibilities available when designing a visualization object.

In order for this to occur, the Image checkbox must be enabled in the properties for the corresponding parameter list visualization object – please refer to → "Description", page 825.

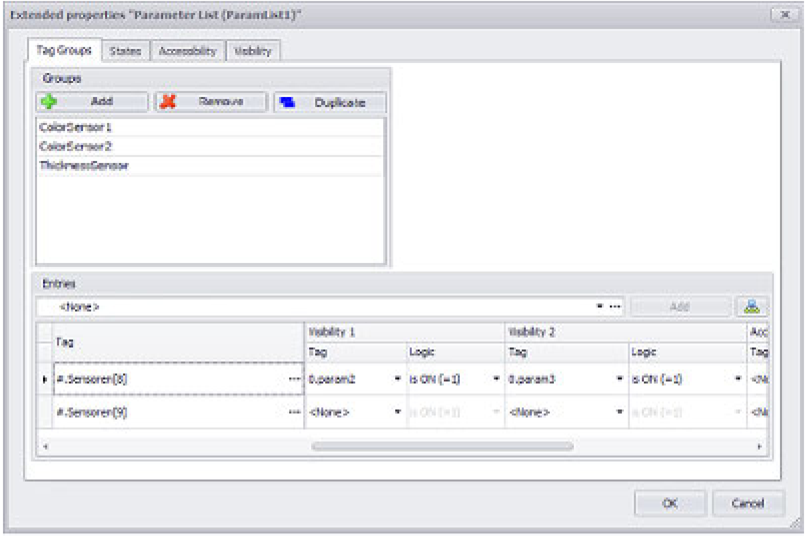
Click on the drop-down menu arrow under the Image column to select the image you want for each entry. Make sure that the image is found in*Project Tree\Media\Project images*. If it is not there, it will not appear in the drop-down menu.
For more detailed information on how to add an image to the project tree, please refer to → section "Step 5: Adding an image", page 174.

To remove a previously selected image so that it is no longer assigned to an entry, simply select the <None> option instead.

Extended properties "Parameter List"
Visibility 1 and Visibility 2

You can enter a maximum of two conditions for tag states. These conditions will be OR'd, meaning that the corresponding tag will be INVISIBLE as soon as any of them are TRUE.

Controls\ Parameter List\ \Tag Groups



Extended properties "Parameter List (ParamList1)"

Tag Groups | States | Accessibility | Visibility

Groups

ColorSensor1
ColorSensor2
ThicknessSensor

Entries

<None>

	Tag	Visibility 1		Visibility 2		Acc
		Tag	Logic	Tag	Logic	Tag
▸	#.Sensoren[8]	0.param2	is ON (=1)	0.param3	is ON (=1)	<N
	#.Sensoren[9]	<None>	is ON (=1)	<None>	is ON (=1)	<N

Figure 528: Extended properties dialog box for a parameter list showing the Tag Groups tab; in this screenshot, a project group containing one single tag has been created

→ Please note that the visibility conditions for the parameter list visualization object itself are not the same as the visibility conditions defined for the individual entries in a tag group!

These columns are used to define conditions under which the corresponding tag will not be visible in the parameter list. In contrast, the Visibility tag for the parameter list visualization object is used to define conditions under which the object itself will not be visible. This means, for example, that it is possible for the parameter list object to be displayed while a tag assigned to it is not.

Example for the Visibility 1 and Visibility 2 columns

The Sensors[8] tag will be shown if

param2 =0; param3=0;

The Sensors[8] tag will not be shown if

param2 =1; param3=0;

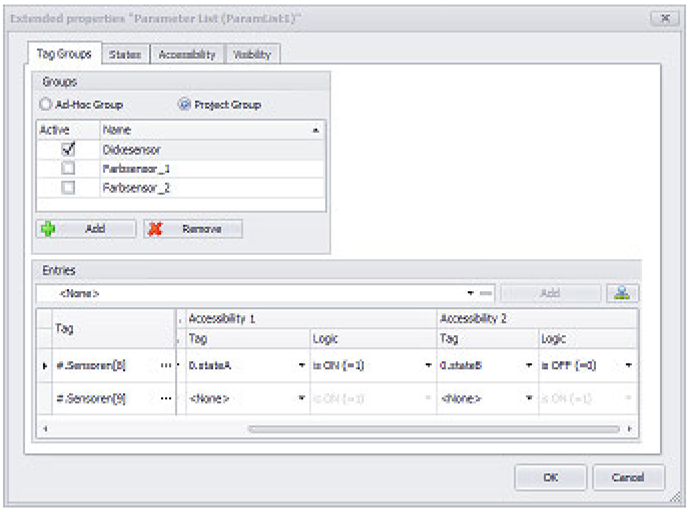
param2 =0; param3=1;

param2 =1; param3=1;

Accessibility 1 and Accessibility 2

You can enter a maximum of two conditions for the accessibility of individual tags. These conditions will be OR'd, meaning that the corresponding tag will be INVISIBLE as soon as any of them are TRUE.

Controls\ Parameter List\ \Tag Groups



Extended properties "Parameter List (ParamList1)"

○ Ad-Hoc Group ● Project Group

Active	Name
☑	Dickesensor
□	Farbsensor_1
□	Farbsensor_2

<None>

Tag	Accessibility 1		Accessibility 2	
	Tag	Logic	Tag	Logic
#.Sensoren[8]	0.stateA	is ON (=1)	0.stateB	is OFF (=0)
#.Sensoren[9]	<None>	is ON (=1)	<None>	is ON (=1)

→ Please note that the accessibility conditions for the parameter list visualization object itself are not the same as the accessibility conditions defined for the individual entries in a tag group!

These columns are used to define conditions under which the corresponding tag will not be accessible in the parameter list. In contrast, the Accessibility tag for the parameter list visualization object is used to define conditions under which the object itself will not be accessible. This means, for example, that it is possible for one of the tags in a tag group not to be accessible even when the corresponding parameter list object itself is accessible.

Example for the Accessibility 1 and Accessibility 2 columns

The Sensors[8] tag from the example above will be accessible if

stateA =0; stateB=1;

The Sensors[8] tag will not be accessible if

stateA =1; stateB=1;

stateA =0; stateB=0;

stateA =1; stateB=0;

9.4 Parameter list functions

The functions for parameter lists focus mostly on ways to sort parameter list entries. Meanwhile, the Show Tag Help function can be used to enter a help text for a selected parameter list entry so that device operators will be able to access the text if necessary. In order to be able to run functions for a parameter list, you will always need a function key. You can design the button for this function key according to your specific needs, e.g., by adding a background color or image to it.

→ To quickly insert the object, use a function key with a special function on the screen 💡

9.4.1 Filter

Filters the entries of the referenced parameter list based on the content of the specified tag at "→ "Filter tag", page 817" of the control → " Parameter List", page 804 .

The filtering remains active until the function is executed again. Thus "live filtering" is possible, by changing the string tag.

Number of the Function	3596 (E0Ch)
Executable from the PLC	No
Function with ON/OFF state	Yes
Parameter 1	Name of parameter list
Function available in standard screens	Yes
Function available with keyboards	No
Function available in script	Yes
Function available in application entry script	No

9.4.2 First Line (Parameter list)

Calling the First line function for a *parameter list* visualization object will make the highlight bar jump to the first entry in the parameter list. This call can only be triggered by a function key visualization object.

→ The function key and parameter list visualization objects must be found on the same screen.

To do so, follow the steps below:

- Click on *Controls\Function Key*.
- Position the object
  Move the cursor to the screen and draw the area for the visualization object.
- In the *Properties 'Function Key'* pane, select *ParameterList* as the group and *Last line* as the function.
- In the Parameter 1 field, select the parameter list visualization object that should be displayed on the screen at runtime.
- Start the Galileo Simulator with function key F5 or
  by clicking on Home\

. Build.

Regardless of which parameter list entry is selected on the device screen at runtime, the highlight bar will jump to the first entry when the button for the function key visualization object is clicked or tapped on.

Number of the Function	3588 (E04h)
Executable from the PLC	No
Function with ON/OFF state	No
Parameter 1	Name of parameter list (X-axis coordinates, Y-axis coordinates)
Function available in standard screens	Yes
Function available with keyboards	No
Function available in script	Yes, name of parameter list
Function available in application entry script	No

Example First line

This example is shown in DemoApplication2.prj.

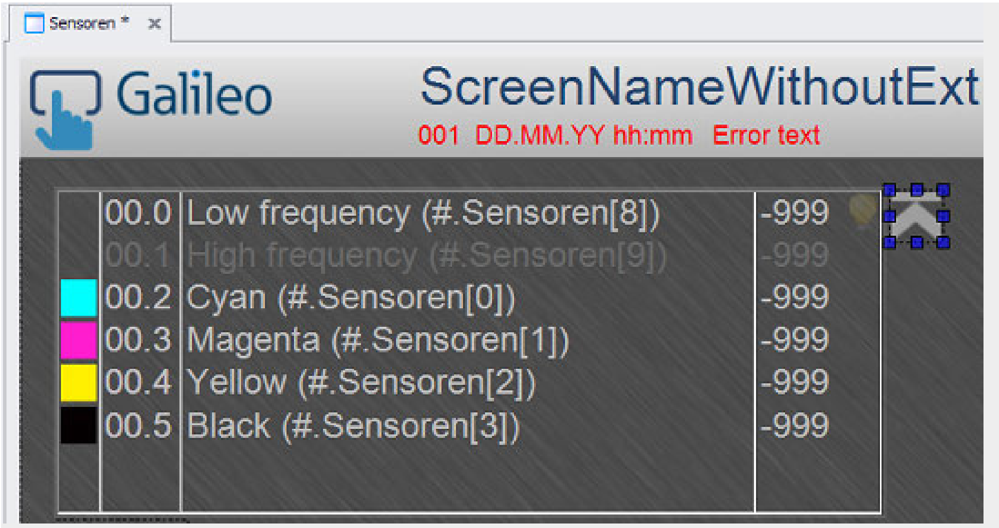
Create a function key visualization object

- Click on *Controls\Function Key*
- Position the object
  Move the cursor to the screen and draw the area for the visualization object.
- In the Properties 'Function Key' pane, select ParameterList in the <Group> field and First line in the <Function > field.

Configure the function key visually

To add an image to the button for the function key visualization object, follow the steps below:

- Go to the Position/Size section in the Properties 'Function Key' pane and enter <40> for the width and <40> for the height.
- Double-click on the function key visualization object.
- Select Icon as foreground in the States tab.
- In the field to the right, click on the drop-down menu arrow and select the image you want, e.g., *Project images\Icons\Navigation\firstLine*.
- Enable the Stretched checkbox.
- Select the Color as background option and then select <Transparent>
- Click on OK.



	00.0	Low frequency (#.Sensoren[8])	-999
	00.1	High frequency (#.Sensoren[9])	-999
🟦	00.2	Cyan (#.Sensoren[0])	-999
🟪	00.3	Magenta (#.Sensoren[1])	-999
🟨	00.4	Yellow (#.Sensoren[2])	-999
⬛	00.5	Black (#.Sensoren[3])	-999

Figure 529: Section from the <Sensors> screen with parameter list visualization object and function key visualization object with <First line> function.

9.4.3 Last Line

Calling the Last line function for a parameter list visualization object will make the highlight bar jump to the last entry in the parameter list. This call can only be triggered by a function key visualization object.

→ The function key and parameter list visualization objects must be found on the same screen.

To do so, follow the steps below:

- Click on *Controls\Function Key*.
- Position the object
  Move the cursor to the screen and draw the area for the visualization object.
- In the *Properties 'Function Key'* pane, select *ParameterList* as the group and *Last line* as the function.
- In the Parameter 1 field, select the parameter list visualization object that should be displayed on the screen at runtime.
- Start the Galileo Simulator with function key F5 or
  by clicking on Home\ . Build.

Regardless of which parameter list entry is selected on the device screen at runtime, the highlight bar will jump to the last entry when the button for the function key visualization object is clicked or tapped on.

Number of the Function	3589 (E05h)
Executable from the PLC	No
Function with ON/OFF state	No
Parameter 1	Name of parameter list (X-axis coordinates, Y-axis coordinates)

Function available in standard screens	Yes
Function available with keyboards	No
Function available in script	Yes
Function available in application entry script	No

Example Last line

This example is shown in DemoApplication2.prj.

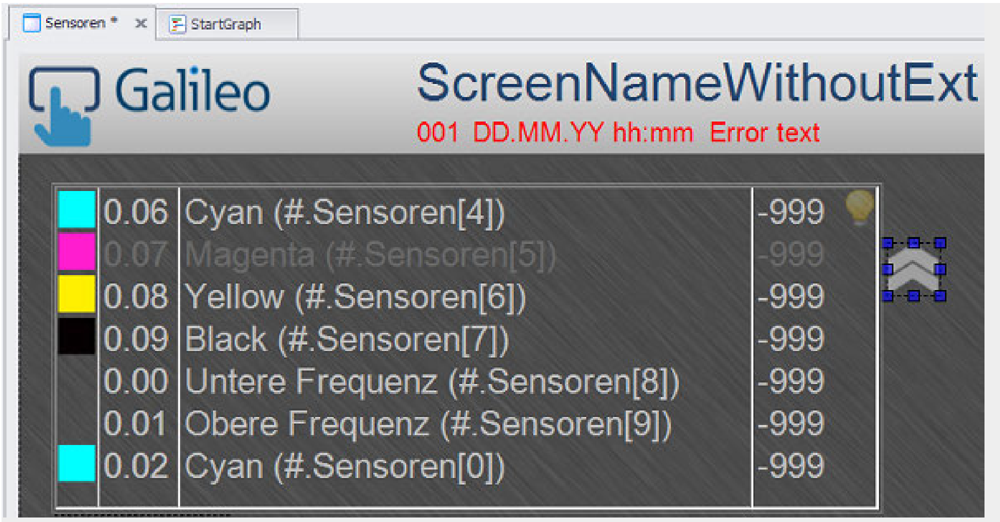
Create a function key visualization object

- Click on the parameter list visualization object.
- Move the cursor to the visualization object's top right corner.
- Move the cursor over the 💡 icon and let it stay there until the quick access menu appears.
- Select the <Last line> option in the quick access menu. A function key visualization object will be automatically created and linked to the corresponding function.

Configure the function key visually

To add an image to the button for the function key visualization object, follow the steps below:

- Go to the Position/Size section in the Properties 'Function Key' pane and enter <40> for the width and <40> for the height.
- Double-click on the function key to open the Extended properties "Function Key" dialog box.
- Select Icon as foreground in the States tab.
- In the field to the right, click on the drop-down menu arrow and select the image you want, e.g., *Project images\Icons\Navigation\lastLine*.
- Enable the Stretched checkbox.
- Select the Color as background option and then select <Transparent>
- Click on OK.



ScreenNameWithoutExt

001 DD.MM.YY hh:mm Error text

🟦	0.06	Cyan (#.Sensoren[4])	-999
🟪	0.07	Magenta (#.Sensoren[5])	-999
🟨	0.08	Yellow (#.Sensoren[6])	-999
⬛	0.09	Black (#.Sensoren[7])	-999
	0.00	Untere Frequenz (#.Sensoren[8])	-999
	0.01	Obere Frequenz (#.Sensoren[9])	-999
🟦	0.02	Cyan (#.Sensoren[0])	-999

Figure 530: Section from the <Sensors> screen with parameter list visualization object and function key visualization object with <Last line> function.

9.4.4 Line down

Calling the Line down function for a parameter list visualization object will make the highlight bar move one entry down in the parameter list.

To create a function key that can be used to call the function, follow the steps below:

- ▶ Click on *Controls\Function Key*.
- ▶ Position the object
  Move the cursor to the screen and draw the area for the visualization object.
- ▶ In the *Properties 'Function Key'* pane, select <ParameterList> as the group and <Line down> as the function.
- ▶ In the Parameter 1 field, select the parameter list visualization object that should be displayed on the screen at runtime.
- ▶ Start the Galileo Simulator with function key F5 or
  by clicking on Home\ . Build.

Regardless of which parameter list entry is selected on the device screen at runtime, the highlight bar will jump to the next entry when the button for the function key visualization object is clicked or tapped on.

Number of the Function	3585 (E01h)
Executable from the PLC	No
Function with ON/OFF state	No
Parameter 1	Name of parameter list (X-axis coordinates, Y-axis coordinates)
Function available in standard screens	Yes
Function available with keyboards	No
Function available in script	Yes
Function available in application entry script	No

Example Line down

This example is shown in DemoApplication2.prj.

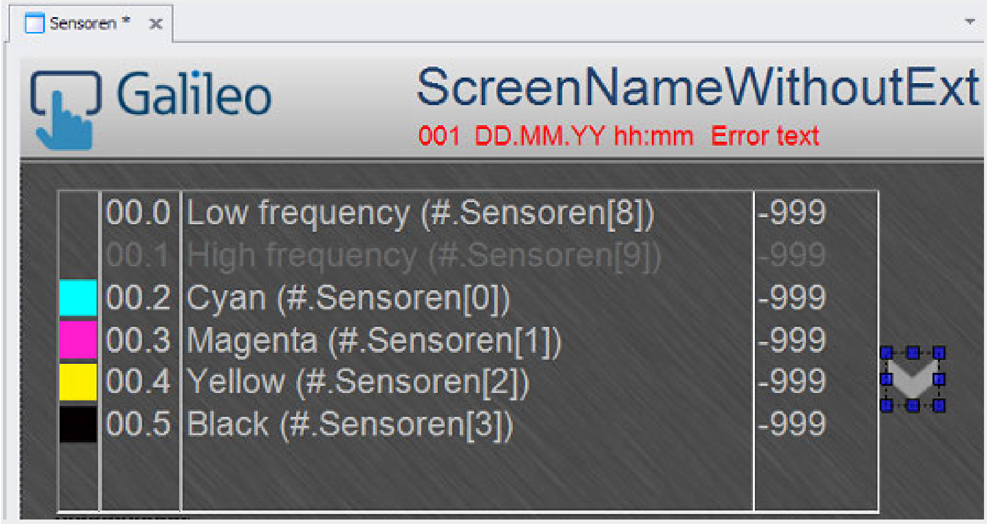
Create a function key visualization object

- Click on the parameter list visualization object.
- Move the cursor to the visualization object's top right corner.
- Move the cursor over the 💡 icon and let it stay there until the quick access menu appears.
- Select the <Line down> option in the quick access menu. A function key visualization object will be automatically created and linked to the corresponding function.

Configure the function key visually

To add an image to the button for the function key visualization object, follow the steps below:

- Go to the Position/Size section in the Properties 'Function Key' pane and enter <40> for the width and <40> for the height.
- Double-click on the function key to open the Extended properties "Function Key" dialog box.
- Select Icon as foreground in the States tab.
- In the field to the right, click on the drop-down menu arrow and select the image you want, e.g., *Project images\Icons\Navigation\Line down*.
- Enable the Stretched checkbox.
- Select the Color as background option and then select <Transparent>
- Click on OK.



ScreenNameWithoutExt

001 DD.MM.YY hh:mm Error text

	00.0	Low frequency (#.Sensoren[8])	-999
	00.1	High frequency (#.Sensoren[9])	-999
🟦	00.2	Cyan (#.Sensoren[0])	-999
🟪	00.3	Magenta (#.Sensoren[1])	-999
🟨	00.4	Yellow (#.Sensoren[2])	-999
⬛	00.5	Black (#.Sensoren[3])	-999

Figure 531: Section from the <Sensors> screen with parameter list visualization object and function key visualization object with <Line down> function.

9.4.5 Line up

Calling the Line up function for a parameter list visualization object will make the highlight bar move one entry up in the parameter list.

This call can only be triggered by a function key visualization object.

> → The function key and parameter list visualization objects must be found on the same screen.

To do so, follow the steps below:

- Click on *Controls\Function Key*.
- Position the object
  Move the cursor to the screen and draw the area for the visualization object.
- In the *Properties 'Function Key'* pane, select <ParameterList> as the group and <Line up> as the function.
- In the Parameter 1 field, select the parameter list visualization object that should be displayed on the screen at runtime.
- Start the Galileo Simulator with function key F5 or
  by clicking on Home\ . Build.

Regardless of which parameter list entry is selected on the device screen at runtime, the highlight bar will jump to the previous entry when the button for the function key visualization object is clicked or tapped on.

Number of the Function	3584 (E00h)
Executable from the PLC	No
Function with ON/OFF state	No
Parameter 1	Name of parameter list (X-axis coordinates, Y-axis coordinates)
Function available in standard screens	Yes
Function available with keyboards	No
Function available in script	Yes
Function available in application entry script	No

Example Line up

This example is shown in DemoApplication2.prj.

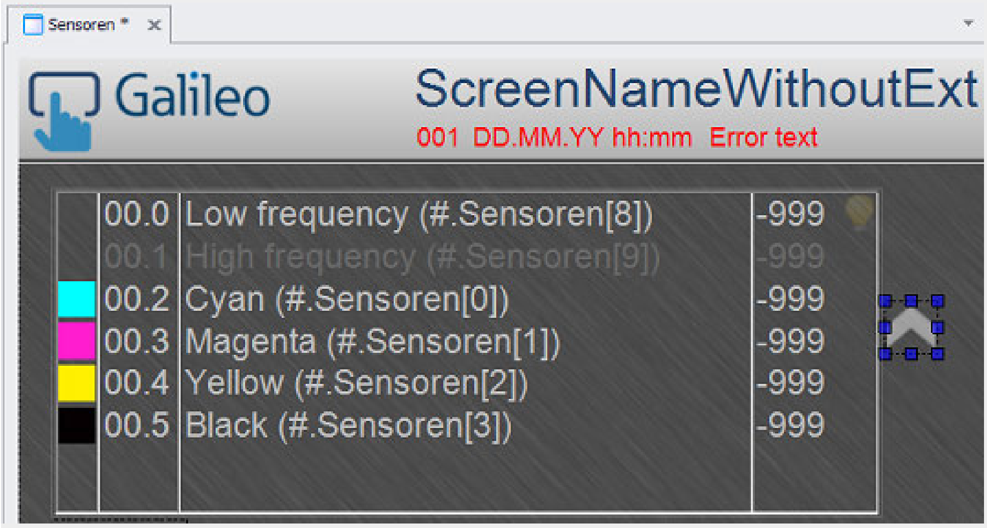
Create a function key visualization object

- Click on the parameter list visualization object.
- Move the cursor to the visualization object's top right corner.
- Move the cursor over the 💡 icon and let it stay there until the quick access menu appears.
- Select the <Line up> option in the quick access menu. A function key visualization object will be automatically created and linked to the function.

Configure the function key visually

To add an image to the button for the function key visualization object, follow the steps below:

- Go to the Position/Size section in the Properties 'Function Key' pane and enter <40> for the width and <40> for the height.
- Double-click on the function key to open the Extended properties "Function Key" dialog box.
- Select Icon as foreground in the States tab.
- In the field to the right, click on the drop-down menu arrow and select the image you want, e.g., *Project images\Icons\Navigation\Line up*.
- Enable the Stretched checkbox.
- Select the Color as background option and then select <Transparent>
- Click on OK.



	00.0	Low frequency (#.Sensoren[8])	-999
	00.1	High frequency (#.Sensoren[9])	-999
🟦	00.2	Cyan (#.Sensoren[0])	-999
🟪	00.3	Magenta (#.Sensoren[1])	-999
🟨	00.4	Yellow (#.Sensoren[2])	-999
⬛	00.5	Black (#.Sensoren[3])	-999

Figure 532: Section from the <Sensors> screen with parameter list visualization object and function key visualization object with <Line up> function.

9.4.6 Page down

Calling the Page down function for a parameter list visualization object will scroll down one page in the parameter list.

This call can only be triggered by a function key visualization object.

→ The function key and parameter list visualization objects must be found on the same screen.

To do so, follow the steps below:

- Click on *Controls\Function Key*.
- Position the object
  Move the cursor to the screen and draw the area for the visualization object.
- In the *Properties 'Function Key'* pane, select *ParameterList* as the group and Page down as the function.
- In the Parameter 1 field, select the parameter list visualization object that should be displayed on the screen at runtime.

- ▶ Start the Galileo Simulator with function key F5 or by clicking on Home\ . Build.

Regardless of which parameter list entry is selected on the device screen at runtime, the next page in the parameter list will be shown when the button for the function key visualization object is clicked or tapped on.

Number of the Function	3587 (E03h)
Executable from the PLC	No
Function with ON/OFF state	No
Parameter 1	Name of parameter list (X-axis coordinates, Y-axis coordinates)
Function available in standard screens	Yes
Function available with keyboards	No
Function available in script	Yes
Function available in application entry script	No

Example Page down

This example is shown in DemoApplication2.prj.

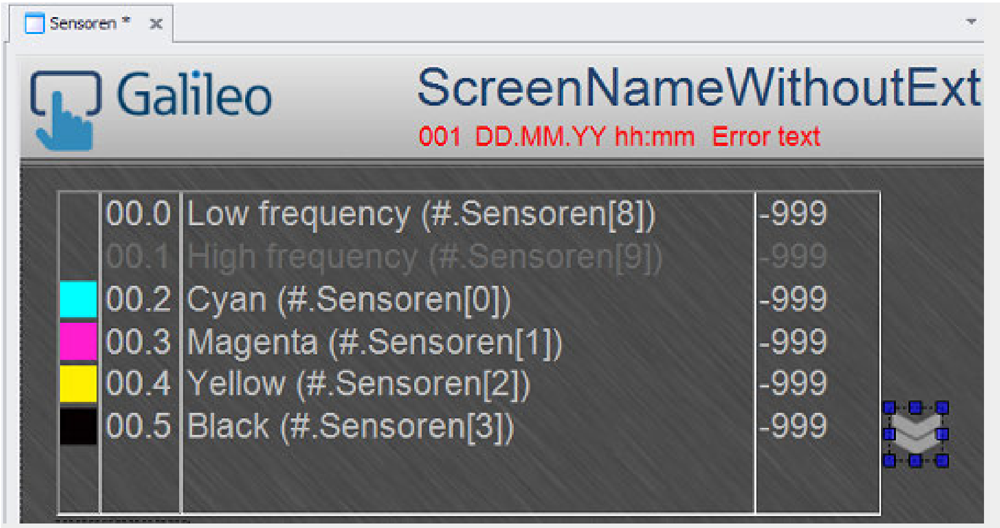
Create a function key visualization object

- ▶ Click on the parameter list visualization object.
- ▶ Move the cursor to the visualization object's top right corner.
- ▶ Move the cursor over the icon and let it stay there until the quick access menu appears.
- ▶ Select the <Page down> option in the quick access menu. A function key visualization object will be automatically created and linked to the corresponding function.

Configure the function key visually

To add an image to the button for the function key visualization object, follow the steps below:

- ▶ Go to the Position/Size section in the Properties 'Function Key' pane and enter <40> for the width and <40> for the height.
- ▶ Double-click on the function key to open the Extended properties "Function Key" dialog box.
- ▶ Select Icon as foreground in the States tab.
- ▶ In the field to the right, click on the drop-down menu arrow and select the image you want, e.g., *Project images\Icons\Navigation\pageDown*.
- ▶ Enable the Stretched checkbox.
- ▶ Select the Color as background option and then select <Transparent>
- ▶ Click on OK.



Sensoren *

Galileo

ScreenNameWithoutExt

001 DD.MM.YY hh:mm Error text

	00.0	Low frequency (#.Sensoren[8])	-999
	00.1	High frequency (#.Sensoren[9])	-999
🟦	00.2	Cyan (#.Sensoren[0])	-999
🟪	00.3	Magenta (#.Sensoren[1])	-999
🟨	00.4	Yellow (#.Sensoren[2])	-999
⬛	00.5	Black (#.Sensoren[3])	-999

Figure 533: Section from the <Sensors> screen with parameter list visualization object and function key visualization object with <Page down> function.

9.4.7 Page up

Calling the Page up function for a parameter list visualization object will scroll down one page in the parameter list. This call can only be triggered by a function key visualization object.

→ The function key and parameter list visualization objects must be found on the same screen.

To do so, follow the steps below:

- Click on *Controls\Function Key*.
- Position the object
  Move the cursor to the screen and draw the area for the visualization object.
- In the *Properties 'Function Key'* pane, select <ParameterList> as the group and <Page up> as the function.
- In the Parameter 1 field, select the parameter list visualization object that should be displayed on the screen at runtime.
- Start the Galileo Simulator with function key F5 or
  by clicking on Home\ . Build.

Regardless of which parameter list entry is selected on the device screen at runtime, the previous page in the parameter list will be shown when the button for the function key visualization object is clicked or tapped on.

Number of the Function	3586 (E02h)
Executable from the PLC	No
Function with ON/OFF state	No
Parameter 1	Name of parameter list (X-axis coordinates, Y-axis coordinates)

Function available in standard screens	Yes
Function available with keyboards	No
Function available in script	Yes
Function available in application entry script	No

Example Page up

This example is shown in DemoApplication2.prj.

Create a function key visualization object

- Click on the parameter list visualization object.
- Move the cursor to the visualization object's top right corner.
- Move the cursor over the 💡 icon and let it stay there until the quick access menu appears.
- Select the <Page up> option in the quick access menu. A function key visualization object will be automatically created and linked to the function.

Configure the function key visually

To add an image to the button for the function key visualization object, follow the steps below:

- Go to the Position/Size section in the Properties 'Function Key' pane and enter <40> for the width and <40> for the height.
- Double-click on the function key to open the Extended properties "Function Key" dialog box.
- Select Icon as foreground in the States tab.
- In the field to the right, click on the drop-down menu arrow and select the image you want, e.g., *Project images\Icons\Navigation\PageUp*.
- Enable the Stretched checkbox.
- Select the Color as background option and then select <Transparent>
- Click on OK.

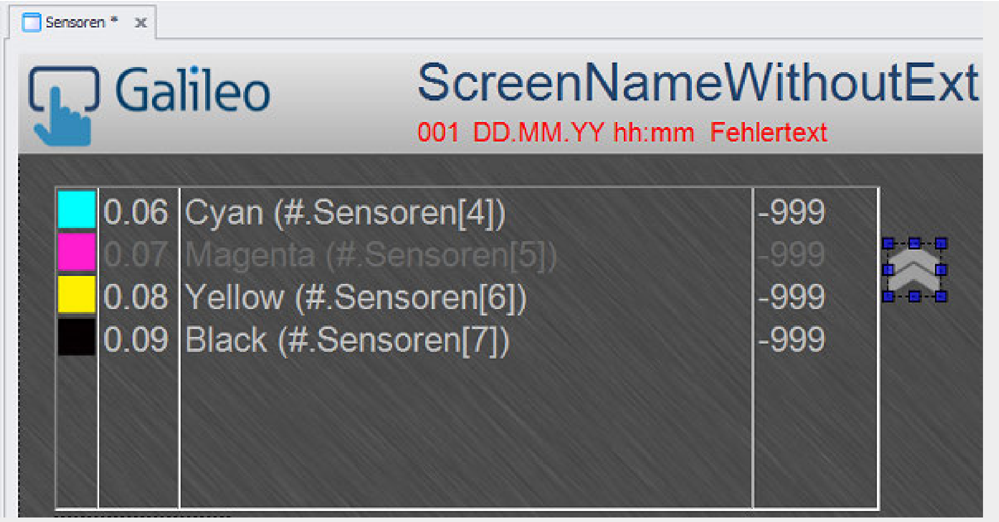

Sensoren *
Galileo
ScreenNameWithoutExt
001 DD.MM.YY hh:mm Fehlertext
0.06 Cyan (#.Sensoren[4]) -999
0.07 Magenta (#.Sensoren[5]) -999
0.08 Yellow (#.Sensoren[6]) -999
0.09 Black (#.Sensoren[7]) -999

Figure 534: Section from the <Sensors> screen with parameter list visualization object and function key visualization object with <Page up> function.

9.4.8 Sort ascend/descend

By default, the entries in parameter list visualization objects are sorted in ascending order, with the entries' index being used as the default sorting criterion. with the entries' index being used as the default sorting criterion. This sorting criterion can be changed using the Sort by Description and Sort by Number functions. Meanwhile, the *Sort Ascend/Descend* function can be used to switch between sorting in ascending order and sorting in descending order for the entries in a parameter list. This call can only be triggered by a function key visualization object.

→ The function key and parameter list visualization objects must be found on the same screen.

To do so, follow the steps below:

- Click on *Controls\Function Key*.
- Position the object
  Move the cursor to the screen and draw the area for the visualization object.
- In the *Properties 'Function Key'* pane, select <ParameterList> as the group and <Sort Ascend/Descend> as the function.
- In the Parameter 1 field, select the parameter list visualization object that should be displayed on the screen at runtime.
- Start the Galileo Simulator with function key F5 or
  by clicking on Home\ . Build.

When the button for the function key visualization object is clicked or tapped on, the order in which the parameter list entries are being sorted on the device screen at runtime will switch from ascending order (default) to descending order. Clicking or tapping on the button again will switch sorting back to ascending order. By default, sorting will be done by number.

→ Not all device types support this function

Number of the Function	3593 (E09h)
Executable from the PLC	No
Function with ON/OFF state	No
Parameter 1	Name of parameter list (X-axis coordinates, Y-axis coordinates)
Function available in standard screens	Yes
Function available with keyboards	No
Function available in script	Yes
Function available in application entry script	No

Since the *Sort Ascend/Descend* function has two states, i.e., *Off* and *On*, a specific icon can be shown as the foreground for each state.

In this case, the *Off* state corresponds to the default setting, i.e., to sorting in ascending order,

while the *On* state corresponds to sorting in descending order.

Example Sort Ascend/Descend

This example is shown in DemoApplication2.prj.

Create a function key visualization object

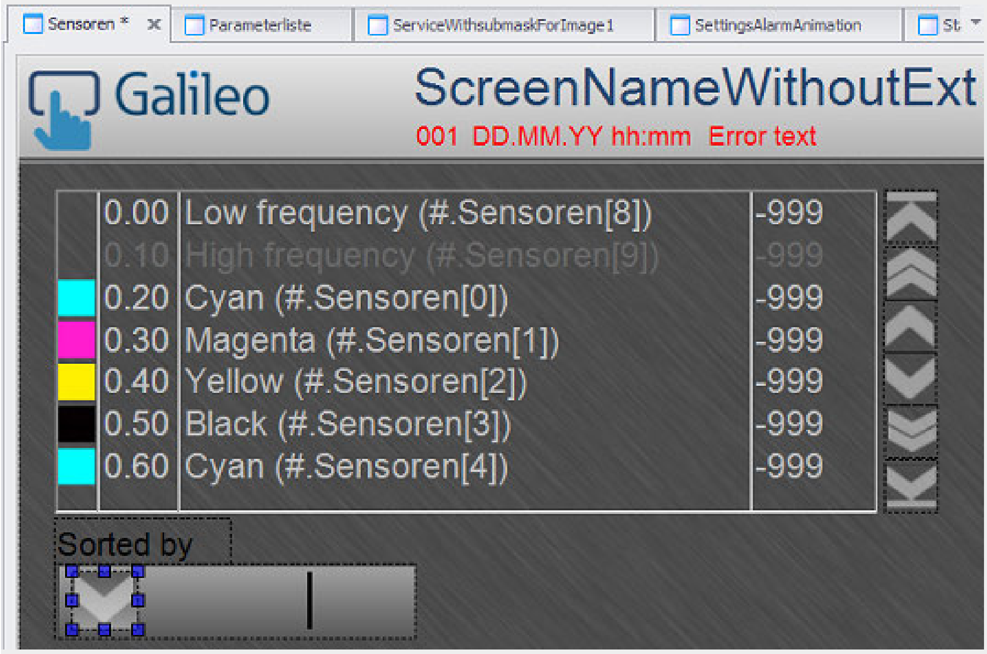
- Click on the parameter list visualization object.
- Move the cursor to the visualization object's top right corner.
- Move the cursor over the 💡 icon and let it stay there until the quick access menu appears.
- Select the <Sort Ascend/Descend> option in the quick access menu. A function key visualization object will be automatically created and linked to the corresponding function.

Configure the function key visually

To add an image to the button for the function key visualization object, follow the steps below:

- Go to the Position/Size section in the Properties 'Function Key' pane and enter <40> for the width and <40> for the height.
- Double-click on the function key to open the Extended properties "Function Key" dialog box.
- Open the States tab and click on the <Off> state. Then select the Icon as foreground option.
- In the field to the right, click on the drop-down menu arrow and select the image you want, e.g., *Project images\Icons\Navigation\<lineUp>*.
- Enable the Stretched checkbox.
- Select the Color as background option and then select <Transparent>

- Now click on the <On> state and select the Icon as foreground option.
- In the field to the right, click on the drop-down menu arrow and select the image you want, e.g., *Project images\Icons\Navigation\<lineDown>*.
- Enable the Stretched checkbox.
- In the States tab, select Color as background and then select <Transparent> as the color.
- Click on OK.



	0.00	Low frequency (#.Sensoren[8])	-999
	0.10	High frequency (#.Sensoren[9])	-999
🟦	0.20	Cyan (#.Sensoren[0])	-999
🟪	0.30	Magenta (#.Sensoren[1])	-999
🟨	0.40	Yellow (#.Sensoren[2])	-999
⬛	0.50	Black (#.Sensoren[3])	-999
🟦	0.60	Cyan (#.Sensoren[4])	-999

Figure 535: Section from the <Sensors> screen with parameter list visualization object and selected function key visualization object with <Sort Ascend/Descend> function in Off state.

9.4.9 Sort by Description

By default, the entries in a parameter list visualization object will be sorted in ascending order by number. You can call the Sort by Description function to change this so that the entries in the parameter list visualization object will be sorted alphabetically *by description*. This call can only be triggered by a function key visualization object.

→ The function key and parameter list visualization objects must be found on the same screen.

To do so, follow the steps below:

- Click on *Controls\Function Key*.
- Position the object
  Move the cursor to the screen and draw the area for the visualization object.
- In the *Properties 'Function Key'* pane, select <ParameterList> as the group and <Sort by Description> as the function.

- In the Parameter 1 field, select the parameter list visualization object that should be displayed on the screen at runtime.
- Start the Galileo Simulator with function key F5 or by clicking on Home\ . Build.

The sorting criterion for sorting the parameter list entries will be switched to by description. When the function key visualization object is clicked or tapped on, the parameter list entries will be shown in ascending or descending alphabetical order based on the sorting criterion. As for whether sorting is done in ascending or descending order, that will depend on whether or not the Sort Ascend/Descend function is activated beforehand.

→ Not all device types support this function

Number of the Function	3592 (E08h)
Executable from the PLC	No
Function with ON/OFF state	No
Parameter 1	Name of parameter list (X-axis coordinates, Y-axis coordinates)
Function available in standard screens	Yes
Function available with keyboards	No
Function available in script	Yes
Function available in application entry script	No

Example Sort by Description

This example is shown in DemoApplication2.prj.

Create a function key visualization object

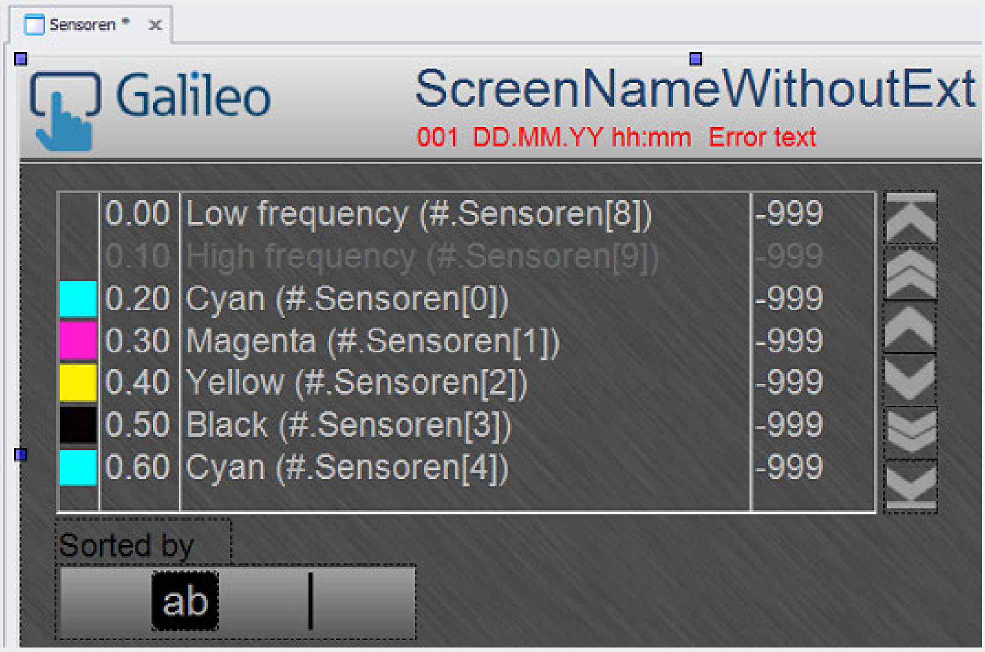
- Click on the parameter list visualization object.
- Move the cursor to the visualization object's top right corner.
- Move the cursor over the 💡 icon and let it stay there until the quick access menu appears.
- Select the <Sort by Description> option in the quick access menu. A function key visualization object will be automatically created and linked to the corresponding function.

Configure the function key visually

To add an image to the button for the function key visualization object, follow the steps below:

- Go to the Position/Size section in the Properties 'Function Key' pane and enter <40> for the width and <40> for the height.
- Double-click on the function key to open the Extended properties "Function Key" dialog box.
- Open the States tab and click on the <Off> state. Then select the Icon as foreground option.

- ▶ In the field to the right, click on the drop-down menu arrow and select the image you want, e.g., *Project images\Icons\Navigation\<abc_OFF>*.
- ▶ Enable the Stretched checkbox.
- ▶ Select the Color as background option and then select <Transparent>
- ▶ Now click on the <On> state and select the Icon as foreground option.
- ▶ In the field to the right, click on the drop-down menu arrow and select the image you want, e.g., *Project images\Icons\Navigation\<abc_ON>*.
- ▶ Enable the Stretched checkbox.
- ▶ In the States tab, select Color as background and then select <Transparent> as the color.
- ▶ Click on OK.



Sensoren *

Galileo

ScreenNameWithoutExt

001 DD.MM.YY hh:mm Error text

	0.00	Low frequency (#.Sensoren[8])	-999
	0.10	High frequency (#.Sensoren[9])	-999
🟦	0.20	Cyan (#.Sensoren[0])	-999
🟪	0.30	Magenta (#.Sensoren[1])	-999
🟨	0.40	Yellow (#.Sensoren[2])	-999
⬛	0.50	Black (#.Sensoren[3])	-999
🟦	0.60	Cyan (#.Sensoren[4])	-999

Sorted by

ab

Figure 536: Section from the <Sensors> screen with parameter list visualization object and selected function key visualization object with <Sort by Description> function in Off state

9.4.10 Sort by Number

You can call the Sort by Number function for a parameter list visualization object so that the entries in the *parameter list* visualization object will be sorted by number. This call can only be triggered by a function key visualization object.

→ The function key and parameter list visualization objects must be found on the same screen.

To do so, follow the steps below:

- ▶ Click on *Controls\Function Key*.
- ▶ Position the object
  Move the cursor to the screen and draw the area for the visualization object.

- In the *Properties 'Function Key'* pane, select <ParameterList> as the group and <Sort by Number> as the function.
- In the Parameter 1 field, select the parameter list visualization object that should be displayed on the screen at runtime.
- Start the Galileo Simulator with function key F5 or by clicking on Home\ . Build.

The sorting criterion for sorting the parameter list entries will be switched back to by number, i.e., the default setting. When the function key visualization object is clicked or tapped on, the parameter list entries will be shown in ascending or descending numerical order based on the sorting criterion. As for whether sorting is done in ascending or descending order, that will depend on whether or not the Sort Ascend/Descend function is activated beforehand.

→ Not all device types support this function

Number of the Function	3591 (E07h)
Executable from the PLC	No
Function with ON/OFF state	No
Parameter 1	Name of parameter list (X-axis coordinates, Y-axis coordinates)
Function available in standard screens	Yes
Function available with keyboards	No
Function available in script	Yes
Function available in application entry script	No

Example Sort by Number

This example is shown in DemoApplication2.prj.

Create a function key visualization object

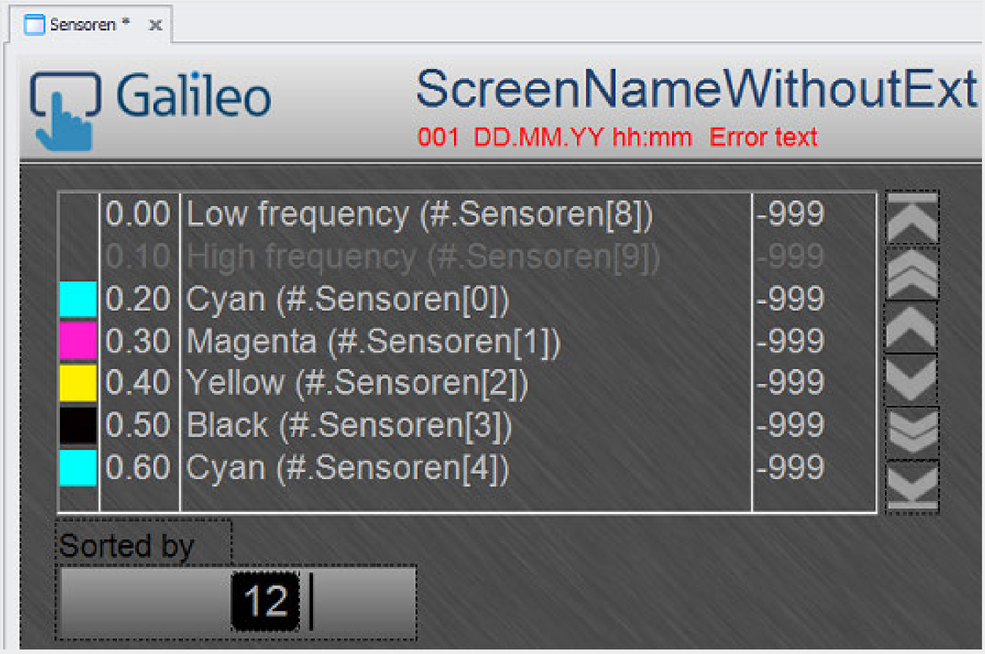
- Click on the parameter list visualization object.
- Move the cursor to the visualization object's top right corner.
- Move the cursor over the 💡 icon and let it stay there until the quick access menu appears.
- Select the <Sort by Number> option in the quick access menu. A function key visualization object will be automatically created and linked to the corresponding function.

Configure the function key visually

To add an image to the button for the function key visualization object, follow the steps below:

- Go to the Position/Size section in the Properties 'Function Key' pane and enter <40> for the width and <40> for the height.
- Double-click on the function key to open the Extended properties "Function Key" dialog box.

- Open the States tab and click on the <Off> state. Then select the Icon as foreground option.
- In the field to the right, click on the drop-down menu arrow and select the image you want, e.g., *Project images\Icons\Navigation\<123_OFF>*.
- Enable the Stretched checkbox.
- Select the Color as background option and then select <Transparent>
- Now click on the <On> state and select the Icon as foreground option.
- In the field to the right, click on the drop-down menu arrow and select the image you want, e.g., *Project images\Icons\Navigation\<123_ON>*.
- Enable the Stretched checkbox.
- In the States tab, select Color as background and then select <Transparent> as the color.
- Click on OK.



	0.00	Low frequency (#.Sensoren[8])	-999
	0.10	High frequency (#.Sensoren[9])	-999
🟦	0.20	Cyan (#.Sensoren[0])	-999
🟪	0.30	Magenta (#.Sensoren[1])	-999
🟨	0.40	Yellow (#.Sensoren[2])	-999
⬛	0.50	Black (#.Sensoren[3])	-999
🟦	0.60	Cyan (#.Sensoren[4])	-999

Figure 537: Section from the <Sensors> screen with parameter list visualization object and selected function key visualization object with <Sort by Number> function in Off state

10. Alarms

In real-life applications, it is necessary to immediately alert device operators to errors and faults in a system. In addition, in order to enable operators to quickly fix problems, it is important to provide accurate and detailed information on the source, allow for flexible options to deal with the problem, and ensure that operators have access to useful help.

To meet this need, GALILEO features an alarm management function that can be used to easily signal and archive errors during a system's operation.

The alarm management function is based on the use of error tags, i.e., tags of data type error or error array. The error data type is a bit tag, while the error array type is an array of bits. When an error tag has a state of 1, it is signaling an error or fault.

The properties for each error tag can be used to assign the tag an error text and a message type. If the error tag has a state of 1, the tag will be shown on the device screen, together with the corresponding error text, with the help of the alarm list visualization object. Moreover, the message type can be used to ensure that error texts will be displayed with different formatting in the alarm list visualization object depending on the message type configured for them.

Exactly one alarm list visualization object can be defined for each screen. Alarm lists can be placed on standard screens and user controls. Since the errors shown in alarm lists often have to be confirmed, multiple function keys are usually added to the corresponding screen as well.

Error texts can be shown on the device or printed on a printer. The following sections describe how error texts can be shown.

A timestamp is added to all errors, and a message type will be assigned to them as well so that this information can be used to generate an error history. Within this scope, a distinction must be drawn between system error messages and project error messages. System error messages are generated by GRS, and cannot be modified by GALILEO users. Meanwhile, project error messages are set up by users and are an integral part of the project. For purposes of clarity, project error messages will simply be referred to as error messages hereafter.

A placeholder can be added to error texts so that this placeholder will be replaced with a tag value or with the ID for a standard text when the corresponding error occurs. This makes it possible to make error texts behave dynamically.

Error texts are shown exclusively using alarm list visualization objects. These error texts are stored together with standard texts and help texts in the Text Editor. For a detailed description of how to work with the Text Editor, importing and exporting texts in general, and error texts in general, please refer to the → "Working with the Text Editor ", page 640 section.

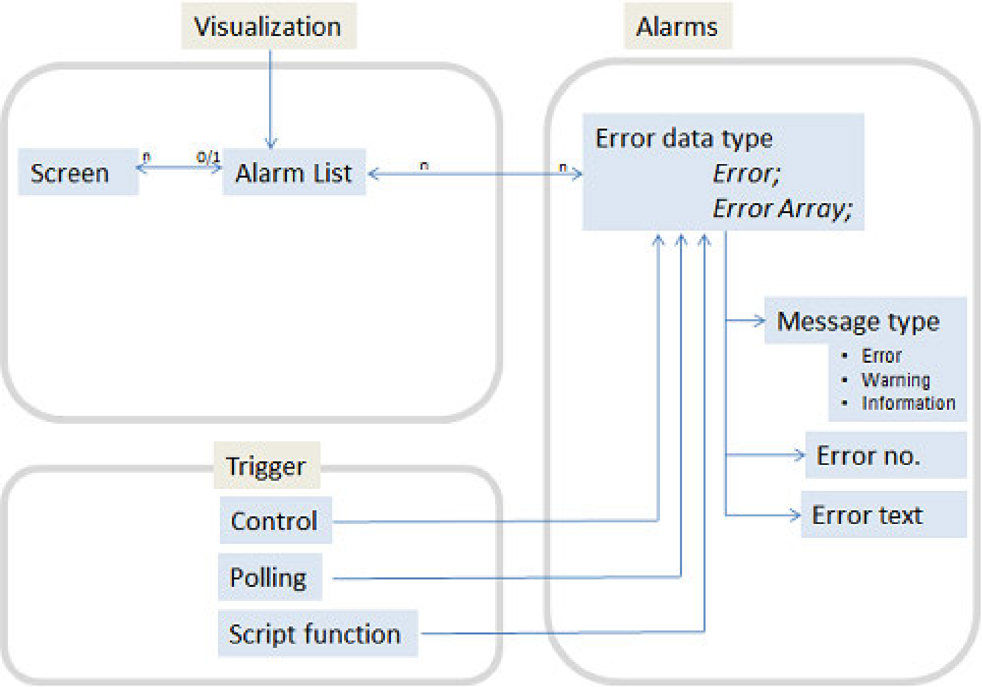
How are error messages triggered and canceled?

To understand how error messages are triggered, it is necessary to take a look at data communications. Generally speaking, an error message is triggered as soon as an error tag in an alarm list assumes a state of 1. The value for this error tag is set by the PLC. But when exactly is it updated?

The device reads the various error tag values from the PLC in the following cases and will automatically update the alarm list after this:

- Event-based error updates triggered by the Control system tag.
- Whenever the Error Refresh script function is run
- Polling (not recommended)

If an error occurs, assistance can be provided to device operators in the form of help information. For a description of the topic of help management, please refer to the → "Help management system", page 930 section



10.1 Displaying messages for device users and providing them with help

Setting up alarm management

In real-life applications, it is necessary to immediately alert device operators to malfunctions and faults in a system with the use of messages. In addition, in order to enable operators to quickly fix problems, it is important to provide accurate and detailed information on the cause, allow for flexible options to deal with the problem, and ensure that operators have access to useful help.

The use case that follows describes a typical scenario, as well as the results expected on the device screen.

Prerequisites

- A device must already be selected, e.g., XV-102-D6-70TWRC-10
- A communication system must already be selected, e.g., XSOFT-CODESYS-2
- A screen must have already been created, e.g., Start screen.
- You must have already imported the error tags you want to display and edit from the PLC.

To set up a small alarm management system, follow the steps below:

1. **Creating tags for errors**

   are normally imported from the PLC. For a description of how to import error tags, please refer to the → section "Importing tags", page 546 section.

   Tags can also be created in GALILEO. To do so, they need to be created in *Project Tree\Tags*. i.e., tags of data type ERROR or error array.

   To create error tags, follow the steps below:

   ▶ Click on *Project Tree\Tags\Internal tags* and open the context menu by right-clicking.

   ▶ Create a tag of data type structure by opening the context menu, clicking on Add, Struct, and then entering a tag name, e.g., <AlarmsX>.

   ▶ Click on the tag you just created and then create as many error tags as you want inside the structure by using the following command sequence: context menu\Add, Error.

   You have now created the error tags.

2. **Creating an alarm list visualization object**

   This section assumes that you only want to display errors in the event of an error, and not warnings or information messages.

   To create an alarm list visualization object, follow the steps below:

   ▶ Click on *Controls\Alarm List*.

- Position the object
  Move the cursor to the screen and draw the area for the visualization object.
- Go to the Properties 'Alarm List' pane and select everything on the same line in the <Representation> field.
- In the Warning and Message fields, use the drop-down menu to set the priority to <low>. Set it to high in the <Error> field.
- Enable the State, Error No.,Date/Time, and Error text checkboxes.

3. Linking error tags in alarms Alarme

Error tags can be easily and conveniently linked using the Alarms tag object. This object will already contain all error tags. To configure the properties for error tags in the Alarms dialog box, follow the steps below:

- Open the Alarms dialog box by clicking on *Home\Alarms*. Then open the Tags tab.

All the error tags in the project will be listed in the Error Tag column.

- Select a message type for each error tag. The error tags will later be prioritized in the alarm list display based on the message type set for them. In addition, the message type can be used to have error texts be output in different colors on the device screen.
- Enter an Error no. for the error tag. You can assign any Error No. you want. This error number will later be shown on the device screen. To make things faster, you can also use series to define multiple error numbers – please refer to → section "Error Tag", page 860.
- Select an Error text for each error tag. Clicking on the ... icon will open a selection of existing error texts.
- You can also place the cursor inside the appropriate Error text cell and simply type in a new text.
- Make sure that the *Error text* checkbox in Controls\Alarm List\Properties 'Alarm List' is enabled. If it is not, the error text will not be shown on the device screen at runtime.

New error texts can also be entered and edited by clicking on *Home\Texts\Error Texts tab*.

4. Using an error text to output tag values

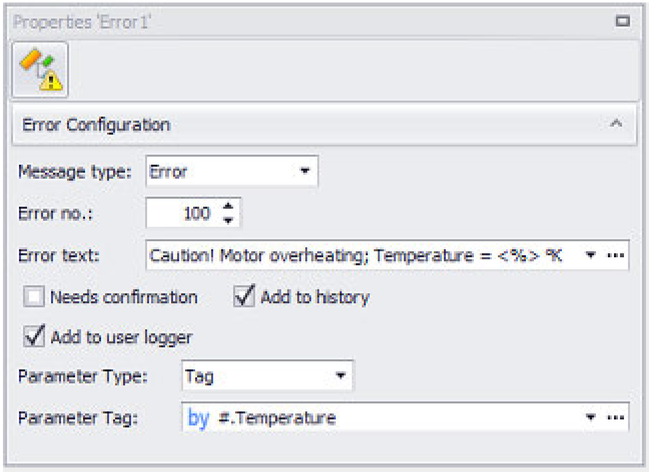
Say you want to assign an error text with a placeholder for a value to an error tag. This can come in handy when the description for an error includes a value that will only be known when the error actually occurs. For example, you could have a scenario in which a system overheats and you want the error message to include the system's current temperature. In this case, you could have a tag of data type WORD hold the temperature value and have this value replace a placeholder in the error text if the error occurs.

In GALILEO, placeholders are assigned addresses with <%>.

To use parameters in error texts, follow the steps below:

- Import the tag that should be used as a parameter or create the tag of data type WORD in *Home\Project Tree\Tags\<PLC tag>*.
- Open the properties for the error tag by clicking on *Home\Project Tree\Tags\<error>*.
- In the Error text field, enter the error text. Then add the following placeholder to this text where required: "<%>".
- Enter an Error no. for the error tag. You can enter any number of your choice.
- In the Parameter Type field, select the <Tag> option.
- In the Parameter Tag field, select the tag you imported or created previously, e.g., <Temperature>.

Example showing how to use an error text to output a tag value



Properties 'Error1'

Error Configuration

Message type: Error

Error no.: 100

Error text: Caution! Motor overheating; Temperature = <%> °K

□ Needs confirmation ☑ Add to history

☑ Add to user logger

Parameter Type: Tag

Parameter Tag: by #.Temperature

Figure 538: Properties for Error2 error tag; in the event of an error, the content of the Temperature tag will replace the <%> placeholder in the error text shown.

5. Using an error text to output a standard text

Say you want to assign an error text and a placeholder for an additional standard text to an error tag. This can come in handy when the additional standard text depends on factors that will only be known when the error actually occurs. For example, you could have a scenario in which a system overheats and you want the error message to offer a potential solution depending on the temperature in the system. If the temperature is greater than 100 °C, the system should be immediately shut down and cooled down. On the other hand, if the temperature is less than 100 °C, system operation should be restricted. In this case, you would create standard texts with IDs using the Text Editor. Then, in the event of an error, the standard text with the ID specified in Parameter Tag would be shown in addition to the error text. In order to achieve this, the standard text with the right ID would be replacing a placeholder in the error text.

In GALILEO, placeholders are assigned addresses with <%>.

To use additional standard texts in error texts, follow the steps below:

- ▶ Create a tag of data type WORD or BYTE with the following command sequence: *Home\Project Tree\Tags\<Internal tags>*, e.g., <IDinTEXTEDITOR>. This tag will be containing the ID for the standard text.
- ▶ Create the standard texts that should be output in addition to the error text. To do so, use the Standard Texts tab in the Text Editor after clicking on *Home\Texts\Texts*.
- ▶ Open the properties for the error tag by clicking on *Home\project Tree\Tags\<error>*.
- ▶ In the Error text field, enter the error text. Then add the following placeholder to this text where required: "<%>".
- ▶ Enter an Error no. for the error tag. You can enter any number of your choice.
- ▶ In the Parameter Type field, select the <Text> option.
- ▶ In the Parameter Tag field, select the tag you created previously, e.g., <IDinTEXTEDITOR>.

6. Simulating errors and Testing the alarm management system

To simulate the error and test the alarm management system, you will first need a button visualization object in order to set and clear a bit. In addition, you will need a loop script that transfers the bit's state to an error bit. The following description builds on the → section "Using an error text to output tag values", page 849 example.

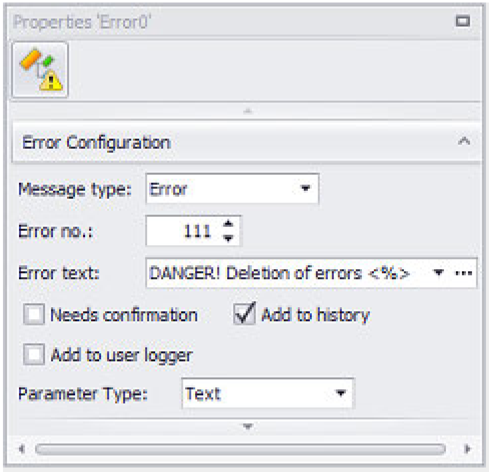
To simulate the error, follow the steps below:

- ▶ Click on *Project Tree\Tags\Internal tags*.
- ▶ Create a new error tag of data type ERROR with *Context menu\Add\Error*, e.g., <Error0>.
- ▶ Create a new tag of data type BIT with *Context menu\Add\Bit*, e.g., <Errorbit>.
- ▶ Click on *Controls\Button* .
- ▶ Position the object
  Move the cursor to the screen and draw the area for the visualization object.

- In the properties for the button visualization object, select Set Bit in the <Action> field and Errorbit in the <Tag> field.
- Click on *Controls\Value Display/Entry* .
- Position the object
  Move the cursor to the screen and draw the area for the visualization object.
- In the properties for the value display/entry visualization object, select IDinTEXTEDITOR in the <Tag> field and Numeric in the <Keyboard> field.
- Click on *Project Tree\Scripts\Loop script*.
- Create a new script with *Context menu\Add script*, e.g., ErrorScript.
- Enter the following line in the script: "Error0:=Errorbit;"
- Start the Galileo Simulator with function key F5 or
  by clicking on Home\ . Build.

This example will allow you to click on the value display/entry object at runtime in order to enter the ID for the standard text. The Errorbit bit tag will not be set until you click on the button. In the loop script, this bit tag is assigned to error tag Error0. Accordingly, the alarm list will show the error text for Error0 and add the standard text with ID 0 or 1 to it.

Example showing how to output an error text that displays a standard text



Properties 'Error0'

Error Configuration

Message type: Error

Error no.: 111

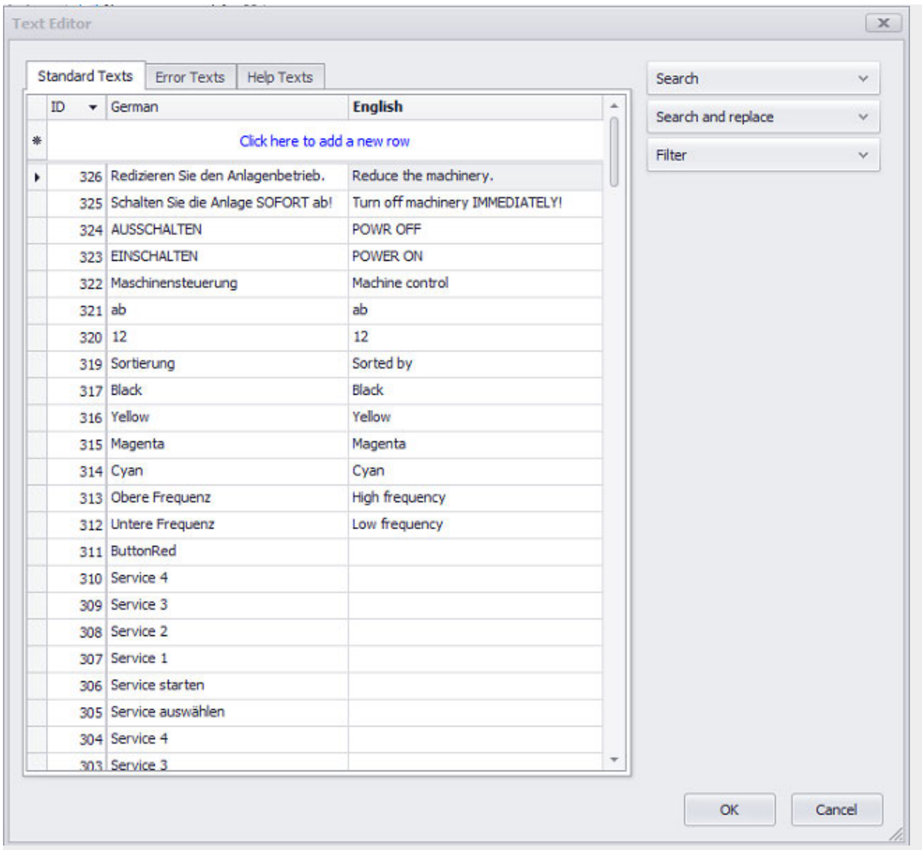
Error text: DANGER! Deletion of errors <%>

□ Needs confirmation ☑ Add to history

□ Add to user logger

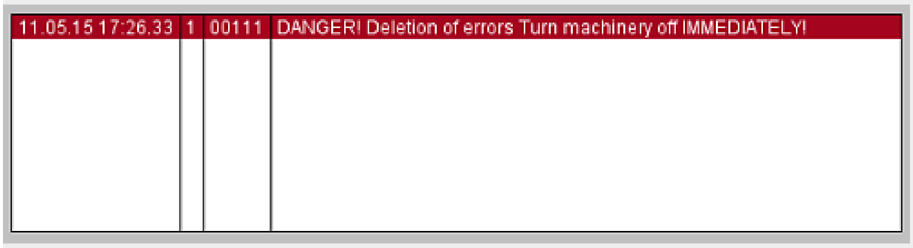
Parameter Type: Text

Figure 539: Properties for the Error0 error tag; the standard text with the ID specified by the tag in the Parameter Tag field will replace the <%> placeholder in the error text displayed when an error occurs.



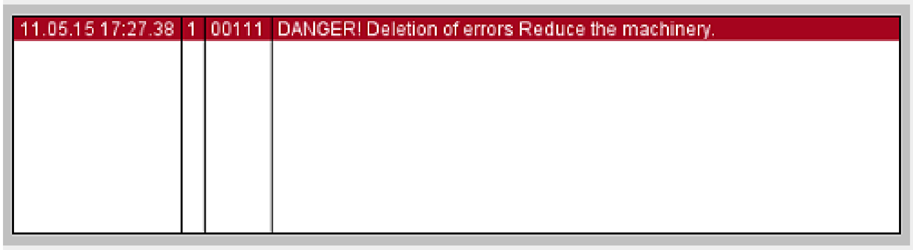
ID	German	English
326	Redizieren Sie den Anlagenbetrieb.	Reduce the machinery.
325	Schalten Sie die Anlage SOFORT ab!	Turn off machinery IMMEDIATELY!
324	AUSSCHALTEN	POWR OFF
323	EINSCHALTEN	POWER ON
322	Maschinensteuerung	Machine control
321	ab	ab
320	12	12
319	Sortierung	Sorted by
317	Black	Black
316	Yellow	Yellow
315	Magenta	Magenta
314	Cyan	Cyan
313	Obere Frequenz	High frequency
312	Untere Frequenz	Low frequency
311	ButtonRed	
310	Service 4	
309	Service 3	
308	Service 2	
307	Service 1	
306	Service starten	
305	Service auswählen	
304	Service 4	
303	Service 3	

Figure 540: Two standard texts in the Text Editor with IDs 27 and 26



11.05.15 17:26.33	1	00111	DANGER! Deletion of errors Turn machinery off IMMEDIATELY!

Figure 541: Error message displayed on device screen; Error0 = 1 and IDinTEXTEDITOR = 0



11.05.15 17:27.38	1	00111	DANGER! Deletion of errors Reduce the machinery.

Figure 542: Error message displayed on device screen; Error0 = 1 and IDinTEXTEDITOR = 1

7. Confirming errors

You can set up an error confirmation for each error tag for when an error event occurs and the error is being displayed. A button must be added so that device operators will be able to confirm the error(s). The → "Error confirm", page 883 / → "Error confirm (actual window)", page 884 / → "Error confirm all", page 884 functions are used to implement the error confirmation functionality. However, the following sections only cover the Error confirm function. Say that you want an error text to be shown in an orange color as soon as a device operator confirms the error. To implement the corresponding error confirmation functionality, follow the steps below:

Creating a confirmation button

- ▶ Open the screen with the Alarm List visualization object, e.g., <AlarmManagement>.
- ▶ Move the cursor over the 💡 icon and let it stay there until the quick access menu appears.
- ▶ Select the <Error confirm> option in the quick access menu. A function key visualization object will be automatically created and linked to the Error confirm function.
- ▶ Double-click on the function key to open the Extended properties "Function Key" dialog box and enter the following text: <Confirm error>.

Requesting a confirmation and setting a color for confirmed errors

- ▶ Open screen *Home\Alarms\Register Handling*.
- ▶ Enable the Needs confirmation checkbox for all error tags.
- ▶ Open the Handling tab and enable the <Confirmation always possible> option.
- ▶ Open screen *Home\Alarms\Register Color*.
- ▶ Select a blue color in the Confirmed errors field.
- ▶ Click on OK.
- ▶ Start the Galileo Simulator with function key F5 or by clicking on Home\ . Build.

At runtime, all errors will be shown with an error number, date/time, and error text in the alarm list. If the error is fixed and the error tag assumes a state of <0>, the error text will still be shown in the alarm list. It will not be removed from the list until the Confirm error button is tapped on.

If the Confirm error button is tapped on and the error is still present, the error tag's state will still be <1> and the error text will be highlighted in blue, although it will still be shown in the alarm list.

8. Providing help information in the event of an error

You can set up troubleshooting information so that device operators will be able to access it if an error event occurs and the error is displayed.

The Error help function is used for this purpose. The DemoApplication2.prj example is used below to show what this help information can look like.

Prerequisites
The <AlarmsX.ErrorBit9> error tag must have already been created. Error number *108* and error text <End position error E34> must have already been assigned in <Home\Alarms\Tags tab>.

To do so, follow the steps below:

By default, the <Help screen of current screen> will be used as the help screen. In other words, the help screen entered as such in the properties for the screen on which the tags are found. Accordingly, the <Help screen of current screen > is set later on.

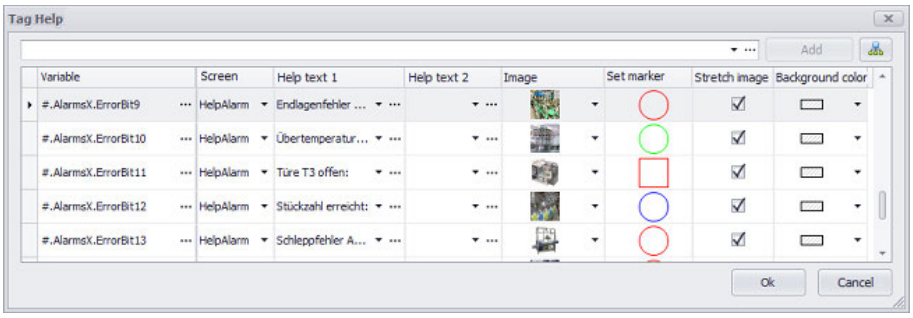
Assigning help information with the Tag Help dialog box

Two help texts and one help image can be defined for each tag. To assign help text 1, follow the steps below:

- Open the tag help by clicking on *Home\Tag Help*.
- Click on the drop-down menu arrow to select the error tag to which you want to assign the help texts, e.g., <AlarmsX.ErrorBit9>.
- In the Help text 1 cell for the error tag, enter the text you want, e.g., <End position error E34>.
- Click on the drop-down menu arrow in the Image column to select the image you want, e.g.,<mb6>. Make sure that the image is found in *Project Tree\Media\Project images*.
- Double-click on the cell to the right, in the Set marker column.

The Marker Editor will appear.

- Place the cursor on the image, press and hold down the left mouse button, and drag the mouse to draw a marking.
- Click on OK.
- Check the settings you have just configured in the Tag Help dialog box.
- Click on OK.

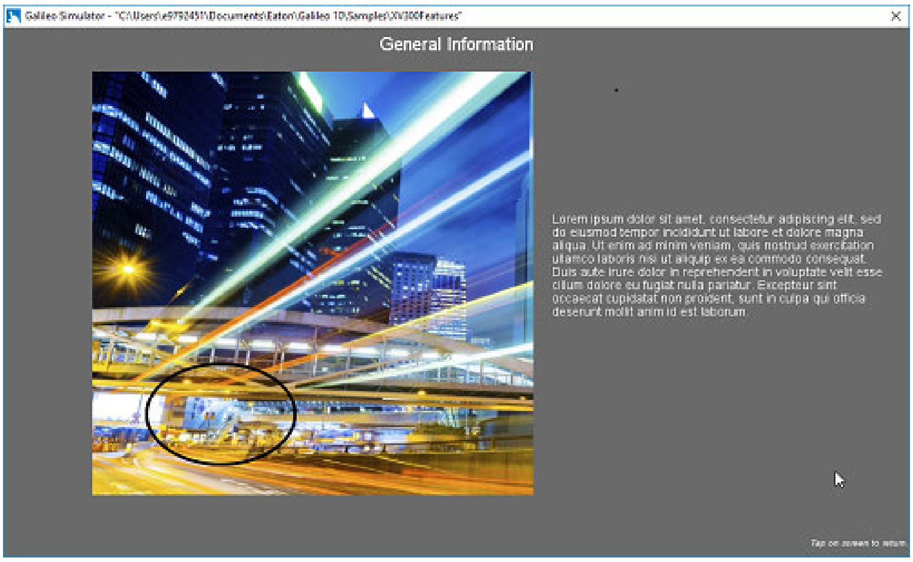


Variable	Screen	Help text 1	Help text 2	Image	Set marker	Stretch image	Background color
#.AlarmsX.ErrorBit9	HelpAlarm	Endlagenfehler ...			○	☑	▭
#.AlarmsX.ErrorBit10	HelpAlarm	Übertemperatur...			○	☑	▭
#.AlarmsX.ErrorBit11	HelpAlarm	Türe T3 offen:			□	☑	▭
#.AlarmsX.ErrorBit12	HelpAlarm	Stückzahl erreicht:			○	☑	▭
#.AlarmsX.ErrorBit13	HelpAlarm	Schleppfehler A...			○	☑	▭

Figure 543: Tag help assigning a help text from the HelpScreen_2 help screen to every tag; the contents for the three help information items are defined here: Help text 1, Help text 2, and Image.

Creating a help screen and configuring its layout

- Open the *Project Tree\Screens\Help Screens folder*.
- Create a new help screen with *Context menu\Add new screen*, e.g., HelpAlarm.
- In the *Controls tab* click on Help Information.
- Position the object
  Move the cursor to the screen and draw the area for the visualization object.
- In the Properties 'Help Information' pane, select the Image option in the <Information type> field.
- In the *Controls tab* click on Help Information.
- Position the object
  Move the cursor to the screen and draw the area for the visualization object.
- In the Properties 'Help Information' pane, select the Help text 1 option in the <Information type> field.
- Enable the Display scrollbar when needed checkbox.


Galileo Simulator - "C:\Users\e9792451\Documents\Eaton\Galileo 10\Samples\XV300Features"
General Information
Tap on screen to return.

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Figure 544: Help screen on device screen at runtime after the <AlarmsX.ErrotBit9> error tag is selected and the help button is tapped on.

Button for accessing the help function

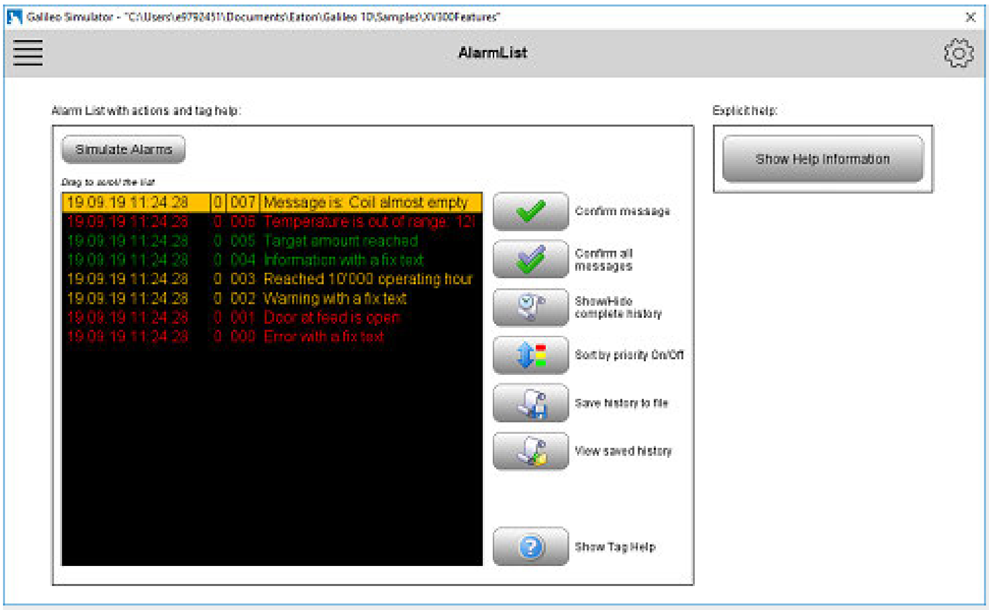
- ▶ Open the screen with the alarm list visualization object, e.g., <AlarmManagement>.
- ▶ Move the cursor over the 💡 icon and let it stay there until the quick access menu appears.
- ▶ Select the <Error help> option in the quick access menu. A function key visualization object will be automatically created and linked to the Error help function.
- ▶ Double-click on the function key to open the Extended properties "Function Key" dialog box and select the Icon as foreground option.
- ▶ Click on the drop-down menu arrow and select an image, e.g., help(0). Make sure that the image is found in *Project Tree\Media\Project images*.
- ▶ Click on OK.
- ▶ Start the Galileo Simulator with function key F5 or

  by clicking on Home\ . Build.

As soon as the device operator selects the <End position error E34> error in the alarm list at runtime and then taps on the ❓ button, the corresponding help screen will be shown with all the corresponding help information. Tapping on the device screen again will make the help screen disappear.

9. → "Refreshingerrors", page 616

Example showing a device screen with an alarm management system



```
19.09.19 11:24:28  0  007  Message is: Coil almost empty
19.09.19 11:24:28  0  006  Temperature is out of range: 12(
19.09.19 11:24:28  0  005  Target amount reached
19.09.19 11:24:28  0  004  Information with a fix text
19.09.19 11:24:28  0  003  Reached 10'000 operating hour
19.09.19 11:24:28  0  002  Warning with a fix text
19.09.19 11:24:28  0  001  Door at feed is open
19.09.19 11:24:28  0  000  Error with a fix text
```

Figure 545: Device screen with alarm list visualization object; the alarm list shows four lines, each with an error number, date/time for the error event, and error text

Follow the steps below to provide device operators with help information in the event of an error:

1. Link the error tag to help texts using the tag help object
2. Configure the settings for the behavior in the event of an error
3. Place the Error Help function on the screen

The Help window is displayed when an error is active. If this option is enabled, the corresponding help screen will be closed automatically as soon as the error becomes inactive. Otherwise the Help window will remain open until it is closed manually.

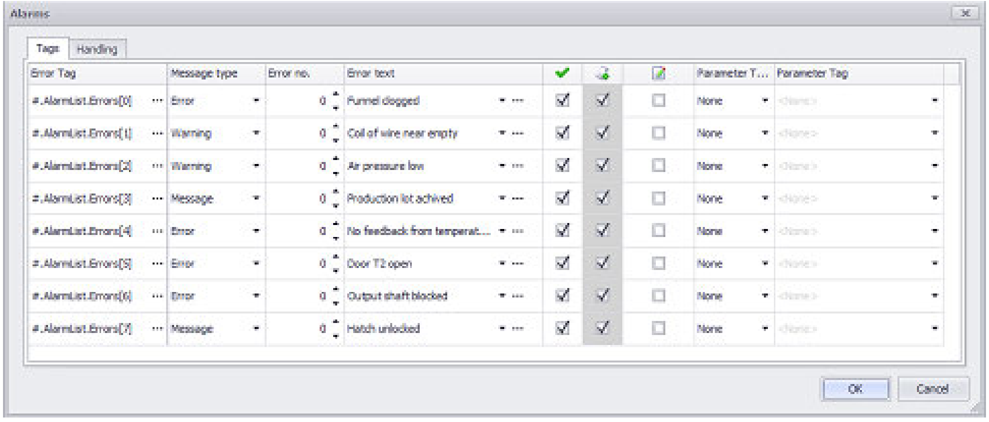
10.2 Alarms

the Alarms tag object can be used to manage error tags and their settings.

Alarms is not a visualization object. Instead, it is a table that contains all the error tags in a project. Alarms ensures that error tags can be configured in a manageable way and can be used to configure the alarm management settings that apply to the entire project.

In addition, the help management functionality can be used to define a help entry for each error tag.

Home\Project Configuration\ Alarms tab



Alarms

Tags | Handling

Error Tag	Message type	Error no.	Error text	✓			Parameter T...	Parameter Tag
#.AlarmList.Errors[0]	Error	0	Funnel clogged	☑	☑	☐	None	<None>
#.AlarmList.Errors[1]	Warning	0	Coil of wire near empty	☑	☑	☐	None	<None>
#.AlarmList.Errors[2]	Warning	0	Air pressure low	☑	☑	☐	None	<None>
#.AlarmList.Errors[3]	Message	0	Production lot achived	☑	☑	☐	None	<None>
#.AlarmList.Errors[4]	Error	0	No feedback from temperat...	☑	☑	☐	None	<None>
#.AlarmList.Errors[5]	Error	0	Door T2 open	☑	☑	☐	None	<None>
#.AlarmList.Errors[6]	Error	0	Output shaft blocked	☑	☑	☐	None	<None>
#.AlarmList.Errors[7]	Message	0	Hatch unlocked	☑	☑	☐	None	<None>

Figure 546: Properties pane Alarm manager

Variables

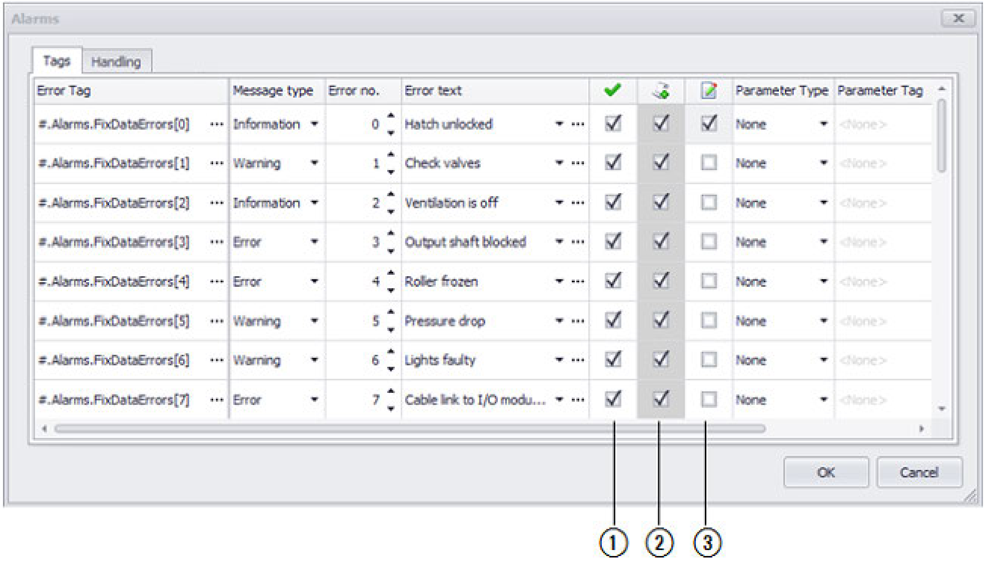
The Tags tab can be used to configure the properties for the various error tags. These settings can also be configured for each individual error tag by going to *Project Tree\Tags\<Error tag>\Properties*.

Error tags are used to display an error text in an alarm list. If anerror tag = 1, the corresponding error text will be displayed. This text can have any length. Error texts can be shown with up to three lines in an alarm list visualization object. If this is not enough to display the entire text, the Error line extend function can be used to show the error text, with scroll bars, in a help screen.

The error texts for error tags are displayed using the color defined for the corresponding message type. This color can be selected in the Color tab.

The context menu features a "Define series" option that can be used to define a series for error numbers. Both error texts and error numbers can be copied to multiple lines using the clipboard.

Home\Project Configuration\ Alarms tab



Error Tag	Message type	Error no.	Error text	✓			Parameter Type	Parameter Tag
#.Alarms.FixDataErrors[0]	Information	0	Hatch unlocked	☑	☑	☑	None	<None>
#.Alarms.FixDataErrors[1]	Warning	1	Check valves	☑	☑	□	None	<None>
#.Alarms.FixDataErrors[2]	Information	2	Ventilation is off	☑	☑	□	None	<None>
#.Alarms.FixDataErrors[3]	Error	3	Output shaft blocked	☑	☑	□	None	<None>
#.Alarms.FixDataErrors[4]	Error	4	Roller frozen	☑	☑	□	None	<None>
#.Alarms.FixDataErrors[5]	Warning	5	Pressure drop	☑	☑	□	None	<None>
#.Alarms.FixDataErrors[6]	Warning	6	Lights faulty	☑	☑	□	None	<None>
#.Alarms.FixDataErrors[7]	Error	7	Cable link to I/O modu...	☑	☑	□	None	<None>

Figure 547: Window Alarms; Register Tags

①	Figure 548: Needs confirmation
②	Figure 549: Add to History
③	Figure 550: Add to user logger

Error Tag

Error tags data types are error or error array. The error data type represents a bit tag, while the error array type represents a bit array in which a state of 1 signals an error.

Error Type

The system distinguishes between three types of messages:

- Errors
- Warning
- Information

The error will be shown with different priorities and colors in the alarm list based on the message type at hand. The colors for these three message types can be selected by going to *Home\Alarms\Color tab*. Meanwhile, the priority level can be defined in the Properties 'Alarm List' pane.

Error no.

Error numbers are assigned when errors need to be labeled and sorted using a numeric system. This makes it possible to set up an error management system that is based exclusively on error numbers, i.e., without error texts.

Error numbers can be freely assigned, and multiple error tags can have the same error number. The error number can also be assigned in the properties for tags of data type error – please refer to → " Error Tags", page 877.

The error number will be passed to the error tags when an error occurs.

When an error occurs, the error number in the alarm list will be shown if the Error No. checkbox is enabled in *Controls\Alarm List\Properties*.

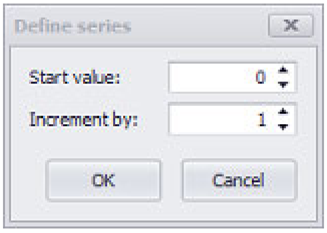
Either enter a number into every Error no. field or define series.

Defining series

To define series, follow the steps below:

- ▶ For the first list entry, click on the No. column.
- ▶ While holding down the <Shift> key, click on the No. column for the last list entry.
- ▶ Multiple list entries will be selected in the No. column.
- ▶ Right-click to open the context menu and click on the <Define series> option.

The Define series dialog box will appear.



Define series

Start value: 0

Increment by: 1

- ▶ Enter the value with which the first list entry should start into the "Start value" field.
- ▶ Enter the increments for the series into the Increment by field.
- ▶ Click on OK.

Error text

Clicking on the drop-down menu arrow will display a drop-down menu with error texts that can be used to select the error text for the corresponding error tag. You can also click directly on the cell to type in a new error text instead.

Error texts are not standard texts. However, they are subject to the same rules as standard texts:

All texts are language-dependent. They are Unicode-compatible and support characters for all languages. Moreover, the number of characters in a text has no limit. Just like standard texts, these texts will be grouped under a tab in the Text Editor, which can also be used to create new texts. To get to this tab, go to *Home\Texts\Error Texts tab*.
A different section explains how to work with texts, including error texts, in detail – please refer to → "Working with the Text Editor ", page 640.

As soon as the text is longer than the defined display area on the device screen, it will be truncated on the right.

✓ **Needs confirmation**

If this option is enabled, the corresponding error will have to be confirmed with one of the following special functions: Error confirm, Error confirm all, or Error confirm (actual window). If the option is not enabled, a confirmation will not be required. When a confirmation is required, there are two different options that can be selected in *Home\Alarms\Register Handling tab*. These two options are different from each other in terms of when the confirmation will be required:

- Confirmation always possible
- Confirmation only possible if an error is inactive

Confirmed errors will be marked with an "X" in the History state column.

An error confirmation example is provided in a different section – please refer to → section "Confirming errors", page 853. Moreover, the special confirmation functions are described in detail in a different section – please refer to → section "Error confirm", page 883, → section "Error confirm all", page 884, and → section "Error confirm (actual window)", page 884.

Add to History

If this option is enabled, the error tag will be added to the history. The history will contain one entry:

- Time of error
- Time when error disappeared
- Time of confirmation if the Needs confirmation option is enabled in *Home\Alarms\Tags tab*.

The history will be stored in RAM memory. Special functions can be used to save its contents in non-volatile memory. In addition, special functions can be used to switch the view in the alarm list between the current errors and the history.

The following special functions are especially relevant within this context:

→ "Show History On/Off", page 889
→ "Save Error History as File", page 888
→ "Save Error History as File Var", page 888
→ "Print History", page 1033
→ "Clear all Histories", page 882

Add to user logger

If this checkbox is enabled, an entry for the error message will be added to the user logger → section "User Logger", page 942.

→ Only the first 200 characters of an error text will be saved in the user logger.

Parameter Type / Parameter Tag

In order to be able to provide device users with detailed and precise information on errors, it is a good idea to be able to expand the corresponding static error texts with a dynamic portion. This can be done by using placeholders. Placeholders are represented using the <%> string. If the <%> placeholder is used in the Error text field, the placeholder will be replaced by the contents of the selected parameter in the event that the corresponding error occurs.

If there are multiple placeholders in a single error text, only the first placeholder will be replaced.

The drop-down menu in the Parameter Type column must first be used to specify whether a placeholder will be used.

Parameter Type

- None
  A placeholder will not be used.
- Tag
  The placeholder will represent a tag of data type BIT, BYTE, WORD, DWORD, QWORD or FLOAT. The corresponding tag must be specified in the Parameter Tag column.
- Text
  The placeholder will represent a standard text. The ID for this text must be specified in the Parameter Tag column.

Parameter Tag

This drop-down menu will only be enabled if either the <Tag> or <Text> option has been selected in the Parameter Type column.

The drop-down menu will show all the tags matching the selection in the Parameter Type column. The selected tag will replace the <%> placeholder at runtime.

In order to ensure data integrity, the polling rate for the parameter tag must always be set to "as fast as possible."

Example illustrating the problem of potential data inconsistency:

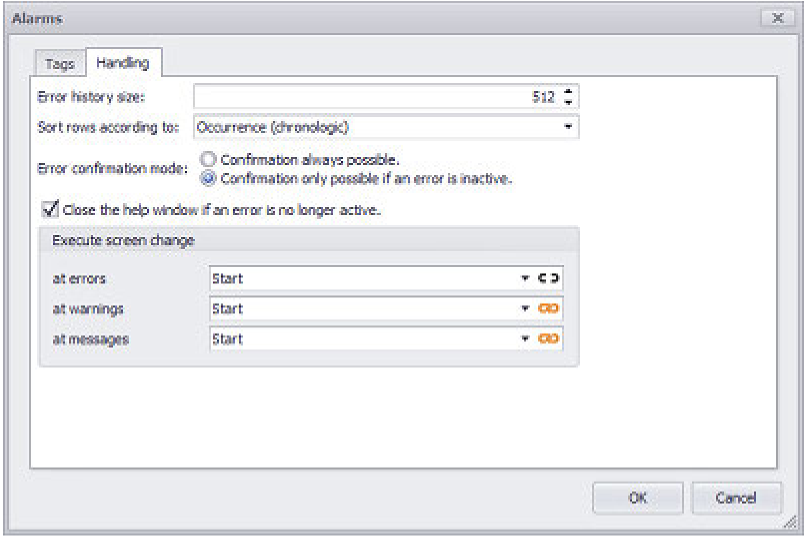
- The polling rate for an error tag is set to 1 sec.
- The polling rate for the corresponding parameter tag is set to 5 sec.
- If an error is triggered, it is possible for the memory to still contain an old parameter tag value. This is because the last parameter tag read operation may have only been 3 seconds ago, for example.

You should also make sure that the parameter tag value is already set correctly on the PLC when the error is triggered (and will not be set after the fact instead).

Handling

The Handling tab is used to define how errors that occur should be handled. Error texts will be shown in the corresponding alarm list until they become inactive or inactive and confirmed. Moreover, current and previous errors, as well as their confirmation, will be recorded in the error history with the corresponding date and time. This history will be stored in RAM, but there are functions that can be used to save the contents in non-volatile memory. In additions, there are functions that can be used to switch the view in an alarm list between the current errors and the error history.

Home\Project Configuration\ Alarms tab



Alarms

Tags | Handling

Error history size: 512

Sort rows according to: Occurrence (chronologic)

Error confirmation mode: ○ Confirmation always possible. ● Confirmation only possible if an error is inactive.

☑ Close the help window if an error is no longer active.

Execute screen change

at errors	Start
at warnings	Start
at messages	Start

Figure 551: Alarms dialog box, Handling tab

Error history size

The error history is used to keep a record of activities for error tags that have the *Add to history* checkbox enabled (*) in* Home\Alarms\☑ Tags tab. A maximum of 4096 activities can be saved with the corresponding date and time. Error confirmations will also be saved. The recorded times will be accurate to a second. Finally, the oldest entries will be overwritten once the history is full.

The Show History On/Off function can be used to show the error history in an alarm list.

The following functions are especially relevant within this context:

→ "Show History On/Off", page 889
→ "Save Error History as File", page 888
→ "Save Error History as File Var", page 888
→ "Print History", page 1033
→ "Clear all Histories", page 882

Sort rows according to

Entries can be sorted based on two different sorting criteria: error occurrence or priority level. This setting is used to define which of these two criteria will be the main one used to determine how entries will be sorted in alarm lists.

- Occurrence (chronologic): Depending on the setting selected in the Sort order field, the newest entries will be added either at the top or at the bottom of the list.

When this option is selected, the entries' priority level will not be taken into account when sorting them.

- Priority: The first sorting criterion will be the entries' priority level. Depending on the setting selected in the Sort order field, the entries with the highest priority level will be shown either at the top or at the bottom of the alarm list. If there are multiple entries with the same priority level, they will be shown as a block. The setting selected in the Sort order field will determine whether the newest event in each priority block will be shown at the top or at the bottom of the block.

The Sort order field is found in the properties for the alarm list visualization object.

The sorting order can be changed using a function key on the device, and can also be changed by the PLC. For a description of this function, please refer to the → section "Error Priority ON/OFF", page 887 section.

Error confirmation mode

Error texts will remain in the alarm list until the value for the corresponding error tag becomes 0 and the error is confirmed. There are two different available options when it comes to error confirmations:

- Confirmation always possible: It will be possible to confirm errors regardless of whether the corresponding error tag has a state of "0" or "1". The error tag will then be removed from the alarm list as soon as its state changes to "0".
- Confirmation only possible if an error is inactive: It will only be possible to confirm errors if the corresponding error tag has a state of "0".

Whether or not a confirmation is required to begin with can be defined in the properties for the corresponding error tags by using the Needs confirmation checkbox.

The color for confirmed errors can be selected in the Color tab.

Error confirmations can be implemented using the following functions:

→ "Error confirm", page 883
→ "Error confirm all", page 884
→ "Error confirm (actual window)", page 884

Close the help window if an error is no longer active

Defines how help screens associated with alarm lists will behave.

In the event of an error, the Error help function can be used to provide device operators with a button that, when tapped on, will show a help text for the selected error tag in the alarm list. If this option is enabled, the corresponding help screen will be closed automatically as soon as the error becomes inactive. Otherwise, the help screen will remain open until the device operator closes it manually. For an example, please refer to the → section "Refreshingerrors", page 616 section

Execute screen change

These settings can be used to configure a screen change to a standard screen of your choice in the project. This screen change can be defined separately for each message type.

The defined Screen Change will take place automatically if an error, a warning or an info message is generated.

There is a global setting for defining whether a visualization is 🔗 linked or ⊂ ⊃ unlinked. In order for the 🔗 and ⊂ ⊃ chain link icons to be displayed, there must be more than one visualization in a group.
For more information, please refer to: → section " Web visualization", page 1070

10.3 Alarm list

Alarm list visualization objects are used to display (on the device screen) the error texts defined for error tags when an error occurs.

Only one alarm list can be defined for each screen. This limit also takes into account whether a UserControl being used is already using an Alarm List visualization object. If multiple errors occur, multiple error texts will be displayed one after the other in the corresponding alarm list.

There are two ways to make it possible for operators to navigate within an alarm list: Either by providing them with function key visualization objects as navigation buttons or
by enabling touch navigation for devices belonging to the XP500 and XV300 product families.

When the former option is used, the function key visualization objects can be used to run functions such as Error first line.
For a description of how to navigate through alarm lists and set up error confirmations, please refer to the→ section "Alarm management functions", page 881 section

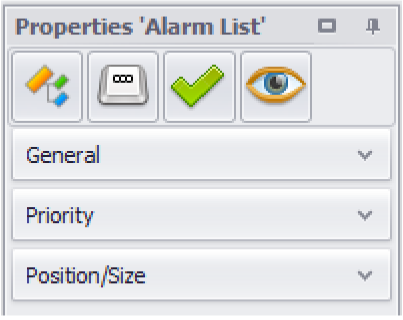
Error texts will be displayed in the Alarm List as long as the corresponding error is active or as long as a required confirmation is not provided for errors that have become inactive.

→ The error entries can be selected directly inside the window.

Alarm lists can be placed on all standard screens and UserControls.

Prerequisites
In order to be able to follow the descriptions below, you must first have clicked on an alarm list visualization object on the current screen.

Controls\ tab



Properties 'Alarm list'

→ section " Tags", page 874

→ " States", page 875

✓

→ section "General", page 868

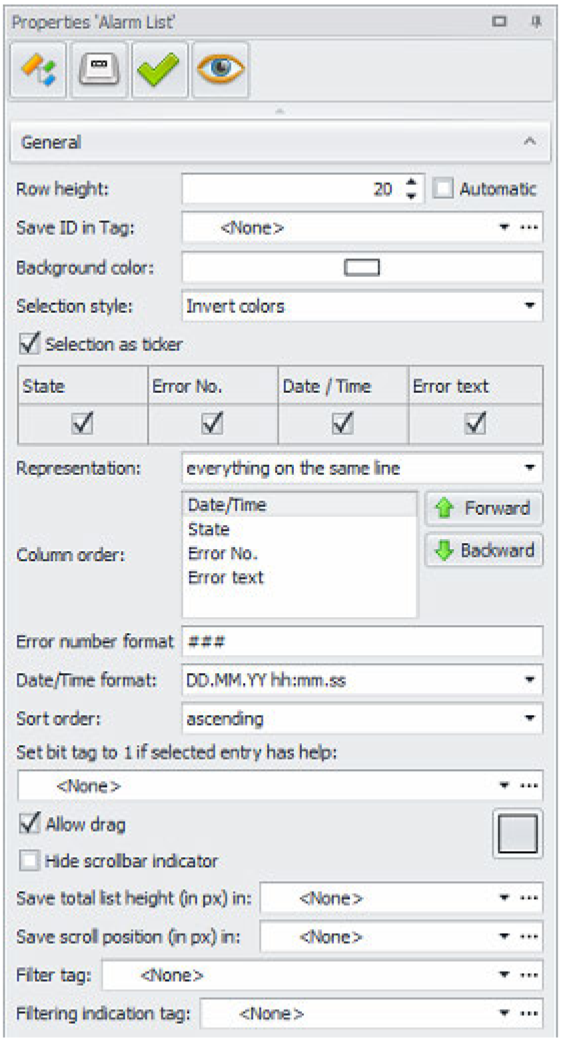
→ section "Priority", page 873

The ✓ accessibility button will only be shown in the Properties pane if a tag and a keyboard have been assigned to the visualization object, as it would not make sense to define accessibility conditions otherwise.

10.3.1 General

The General section can be used to configure all the properties that affect how the visualization object will be displayed, with the exception of its size and position.

Controls\ Alarm List tab



Properties 'Alarm List'

General

Row height: 20 □ Automatic

Save ID in Tag: <None>

Background color: □

Selection style: Invert colors

☑ Selection as ticker

State	Error No.	Date / Time	Error text
☑	☑	☑	☑

Representation: everything on the same line

Column order:
Date/Time
State
Error No.
Error text

Forward
Backward

Error number format ###

Date/Time format: DD.MM.YY hh:mm.ss

Sort order: ascending

Set bit tag to 1 if selected entry has help: <None>

☑ Allow drag

□ Hide scrollbar indicator

Save total list height (in px) in: <None>

Save scroll position (in px) in: <None>

Filter tag: <None>

Filtering indication tag: <None>

Figure 552: Alarm list\General properties

Row height

Used to define the row height in pixels independently of the selected font size. If the row height is smaller than the font size, the text will be cut at the top and bottom.

☑ Automatic adaptation

Save ID in Tag

This ID refers to the error number. The Error no. for the entry currently selected in the alarm list will be passed to the tag selected in this drop-down menu. This tag can be of data type BYTE, WORT, or DWORD. In addition, it can then be written to the PLC or passed to a script so additional actions can be carried out based on the selected error number.

If an alarm is not selected or there is no listed alarm anymore, a value of -1 will be written to the "Save ID in Tag" tag.

As soon as you select a tag under Save ID in Tag, you will be able to enable the Set -1 if no selection option.

Background color

A color can be selected as the background for the list.

You can change the color if you want. The following options are available for this purpose:

- Transparent
- Standard Colors
- More Colors...

The most recently used colors will be shown (user-specific).

For detailed information on the available colors, please refer to the following section: → section "Color choice - Object's Color Palette", page 234

Selection style

The appearance can be defined for each list.

Invert colors and frames are available.

Depending on the selection, the active line on the device is either inverted, i.e. differently colored text line background or a frame line at the top and bottom marks the active line.

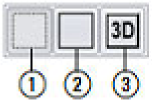
Selection as scrolling text

When enabled ☑, the selected alarm entry will be shown as animated scrolling text if the text length exceeds the object's width.

→ This option is not available for all device models.

Frame 3D

The visualization object can be displayed with any one of three different designs.



① no frame

② with frame

③ 3D frame

Automatic

If this checkbox is enabled, GALILEO will determine the line height automatically based on the font and font size being used.

→ The checkbox will be automatically enabled as soon as the Representation setting is set to <everything on the same line>.

State

If this checkbox is enabled, the column showing error states will be shown in the alarm list. This state can assume any of the following values:

0 = Error no longer present
1 = Error present
X = Error has been confirmed

Error no.

If this checkbox is enabled, the column showing error numbers will be shown in the alarm list.

Error numbers are assigned in the properties for tags of data type error – please refer to → section " Error Tags", page 877.

Time/Date

If this checkbox is enabled, the column showing the date and time will be shown in the alarm list. This date and time will correspond to the moment when the error event occurred.

Error text

If this checkbox is enabled, the column showing the error text as a column or additional line(s) will be shown in the alarm list. For a description of how to work with texts, including error texts, please refer to the → section "Working with the Text Editor ", page 640 section

Representation

The following display options are available for error texts:

everything on the same line	The error text will be shown on a single line with the other selected information. The order of columns can be customized. If the error text is longer than the space in a single line, you can increase the size of the alarm list, select a smaller font, or select a display option with additional lines.
an additional line for the error text	The error text will be shown on 1 to 3 lines.
two additional lines for the error text	
three additional lines for the error text	

Column order

This list can be used to customize the order in which the enabled columns will be shown. If one of the additional line(s) options is selected in the <Representation> field, it will not be possible to move the error text to a different position and, accordingly, it will not be shown here.

Error number format

You can use this format to define how the error numbers in the alarm list will be shown. Each "#" character in this field is a placeholder for one number. The following characters can be used as separators between the number placeholders:
; : . , ' / - |

If the error number has fewer decimal places than the defined format, the places are filled with leading zeros.

If the error number has fewer decimal places than the defined format, the places are filled with leading zeros.

Date/Time format

This display format field is used to define how the time and/or date will be shown.

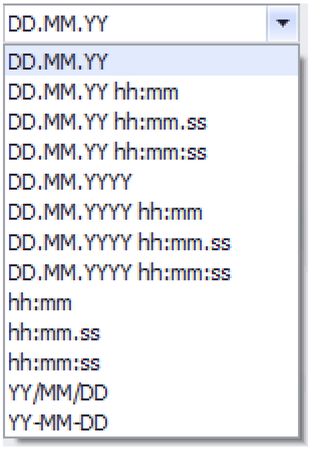


Figure 553: Date/Time format drop-down menu

DD	Day (1...31)
MM	Month (1...12)
YY	Year two digit (e.g. 15)
YYYY	Year four digit (e.g. 2015)
hh	Hour (0...23)
mm	Minute (00...59)
ss	Second (0...59)

Sorting

The sort order option selected will have different results based on whether the entries should be sorted by the time the error event occurred or by priority:

Sort rows according to: Occurrence (chronologic)

- <ascending>
  The most recent error event will be shown at the top of the alarm list.
- <descending>
  The most recent error event will be shown at the bottom of the alarm list.

Sort rows according to: Priority

- <ascending>
  The highest priority will be at the top of the alarm list and the most recent error event within each priority will be at the top of the block for that priority.
- <descending>
  The lowest priority will be at the top of the alarm list and the oldest error event within each priority will be at the top of the block for that priority.

The setting chosen in Home\Alarms\Handling tab\Sort rows according to determines whether the *alarm list* will be sorted by occurrence or by priority.

Set tag if there is help for the selected entry:

Used to select an available tag that will open a help text.

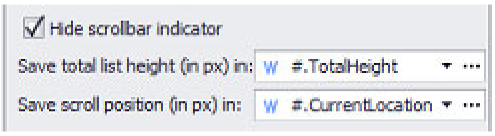
Allow drag

These touch gestures can only be used with PTC devices such as the XV300 and XP500.
This function is also supported by GALILEO Open applications.

Hide scrollbar indicator

In order to work with objects for which a list is stored (parameter lists, alarm lists, recipes, and containers) more efficiently, you can save the position in the list.

☑ If the checkbox is enabled, the following settings for the list object will be hidden.



☑ Hide scrollbar indicator

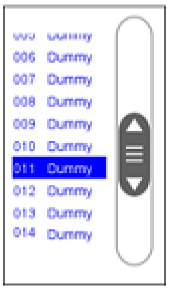
Save total list height (in px) in: W #.TotalHeight

Save scroll position (in px) in: W #.CurrentLocation

Save list height (in px) in: Saves the total list height (total height with all items added up) in the specified tag

Save scroll position (in px) in: Saves the current scroll position within the total list height in the specified tag.

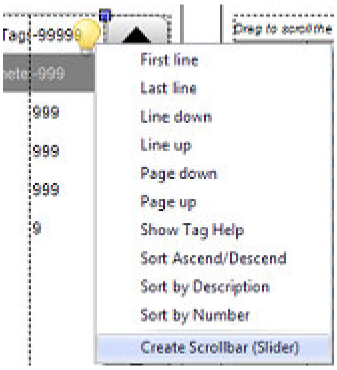
Together with Save list height (in px) in, this tag can be used, for example, in order to use the slider in a slider object as a scrollbar in the list.



005 Dummy
006 Dummy
007 Dummy
008 Dummy
009 Dummy
010 Dummy
011 Dummy
012 Dummy
013 Dummy
014 Dummy

To make it easier to create this slider object, you can follow the steps below:

1. Assign tags to the Save list height (in px) in and Save scroll position (in px) in settings
2. In the screen designer, move the cursor to the upper right corner of the object. The 💡 icon will appear
3. Click on the 💡 icon and select the Generate scrollbar (slider) option in the drop-down menu



4. A slider object will be created and the tags will already be assigned correctly.

Filter tag:

Specification of a tag of the type String. Together with the special function Filter the alarm list can then be filtered according to the content of the string variable. Only the Error text column is considered for the filter.

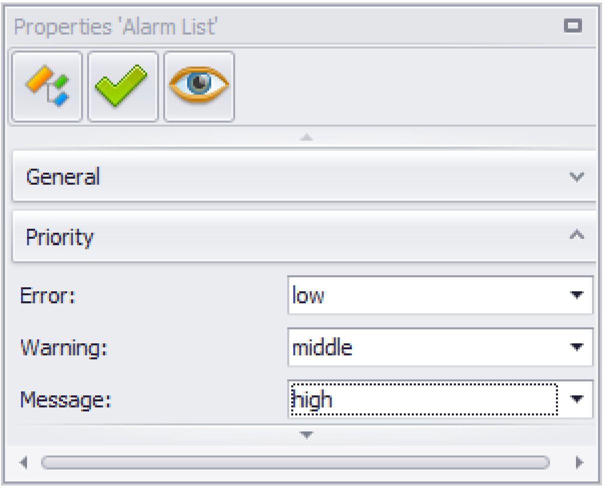
Filtering indication tag:

Specification of a bit tag. If filtering of the alarm list is active, the bit is set to 1. Otherwise it has the value 0.

10.3.2 Priority

In order for the entries in an alarm list to be sorted by priority, the various message types must first be assigned priority levels.

Controls\ Alarm List tab



Error: low

Warning: middle

Message: high

Figure 554: Properties 'Alarm List'\Priority

The following priorities can be set separately for each alarm list:

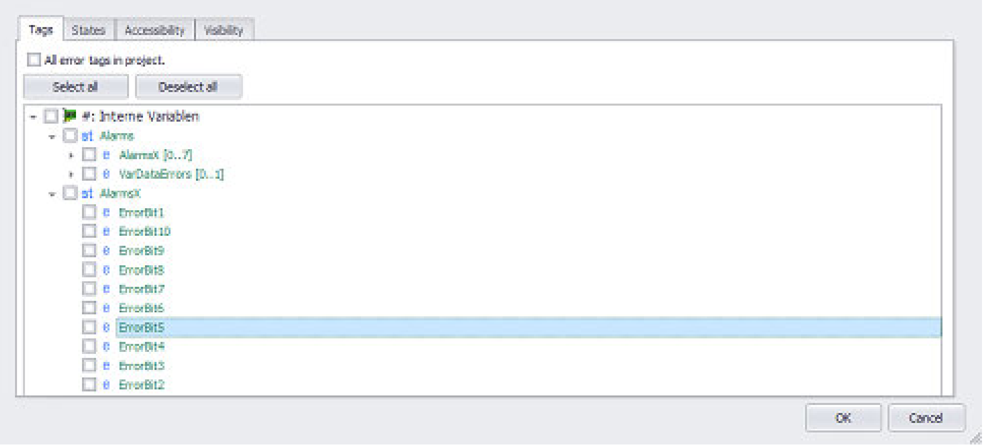
- High = Will be displayed as the first entry
- Middle = Will be displayed as the second entry
- Low = Will be displayed at the end of the entries

To set the sorting criterion for the entries in an alarm list to "priority," go to *Home\Alarms\Handling* – please refer to → section "Handling", page 863 as well.

10.3.3 Tags

The list in the Tags tab shows all the error tags in the project. To select the error tags that you want the alarm list visualization object to show, simply enable the corresponding checkboxes.

Controls\ Alarm List tab



□ All error tags in project.

- □ #: Interne Variablen
  - □ st Alarms
    - □ e AlarmsX [0..7]
    - □ e VarDataErrors [0..1]
  - □ st AlarmsX
    - □ e ErrorBit1
    - □ e ErrorBit10
    - □ e ErrorBit9
    - □ e ErrorBit8
    - □ e ErrorBit7
    - □ e ErrorBit6
    - □ e ErrorBit5
    - □ e ErrorBit4
    - □ e ErrorBit3
    - □ e ErrorBit2

Figure 555: Extended properties 'Alarm List' tag; Tags tab

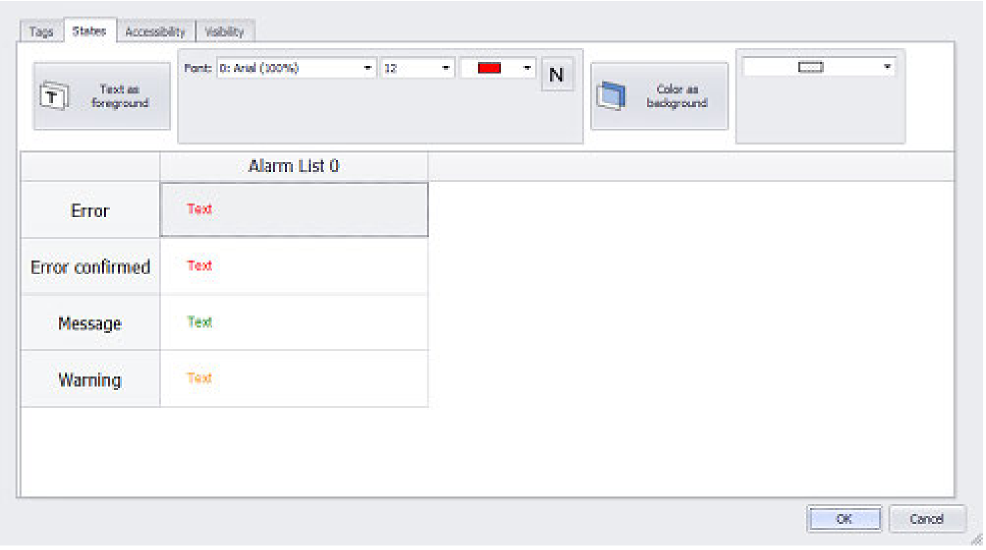
☑ All the error tags in the project

If this option is enabled, all the error tags in the project will be shown in the alarm list.

10.3.4 States

Each instance of an alarm list visualization object can have one state and one style.

Controls\ Alarm List tab



	Alarm List 0
Error	Text
Error confirmed	Text
Message	Text
Warning	Text

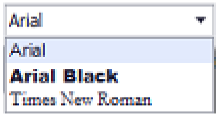
Figure 556: Extended properties "Alarm List" dialog box; Tags

Alignment — The buttons shown below can be used to configure the text to be left-aligned, centered.

Font Properties

Font

With the exception of the <Arial> default font, the fonts available will be those that have been added in *Home\Languages\Project Languages*.

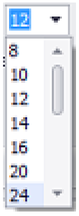


Arial

- Arial
- Arial Black
- Times New Roman

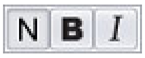
Font Size

The font size can be left with its default value or can be manually changed to a value within a range of 1 to 999 pt. The line height will be automatically adjusted.



Font Style N

The following formatting options are available:



- Default (**N**ormal)
- **Bold** (**B**old)
- *Italic (**I**talic)*

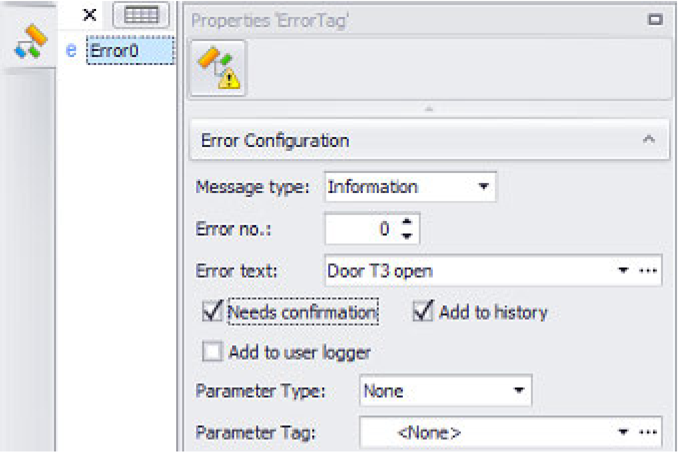
The basic configuration parameters for the visualization object's foreground and background are described in detail in the following section: Defining text series

10.4 e Error Tags

These properties are only available for error tags, i.e., they will not be shown in the properties for other data types. All the properties that can be configured here can also be configured in *Home\Alarms\Tags tab*. The alarm management function is based on the use of error tags, i.e., tags of data type error or error array. The error data type is a bit tag, while the error array type is an array of bits. When an error tag has a state of 1, it is signaling an error or fault.

For a description of how errors are communicated, please refer to the → section "Refreshingerrors", page 616 section.

Project Organization\Tags\ e



Error0

Properties 'ErrorTag'

Error Configuration

Message type: Information

Error no.: 0

Error text: Door T3 open

☑ Needs confirmation ☑ Add to history

□ Add to user logger

Parameter Type: None

Parameter Tag: <None>

Figure 557: Properties pane for the created tag of type error'[0]'

Error configuration

Error Type

The system distinguishes between three types of messages:

- Errors
- Warning
- Information

The error will be shown with different priorities and colors in the alarm list based on the message type at hand. The colors for these three message types can be selected by going to *Home\Alarms\Color tab*. Meanwhile, the priority level can be defined in the Properties 'Alarm List' pane.

Error no.

Error numbers are assigned when errors need to be labeled and sorted using a numeric system. This makes it possible to set up an error management system that is based exclusively on error numbers, i.e., without error texts.

Error numbers can be freely assigned, and multiple error tags can have the same error number. The error number can also be assigned in the properties for tags of data type error – please refer to → " Error Tags", page 877.

The error number will be passed to the error tags when an error occurs.

When an error occurs, the error number in the alarm list will be shown if the Error No. checkbox is enabled in *Controls\Alarm List\Properties*.

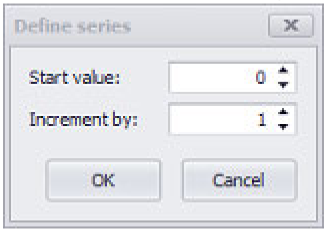
Either enter a number into every Error no. field or define series.

Defining series

To define series, follow the steps below:

- ▶ For the first list entry, click on the No. column.
- ▶ While holding down the <Shift> key, click on the No. column for the last list entry.
- ▶ Multiple list entries will be selected in the No. column.
- ▶ Right-click to open the context menu and click on the <Define series> option.

The Define series dialog box will appear.


Define series

Start value: 0

Increment by: 1

- ▶ Enter the value with which the first list entry should start into the "Start value" field.
- ▶ Enter the increments for the series into the Increment by field.
- ▶ Click on OK.

Error Text

Error texts are not standard texts. However, they are subject to the same rules as standard texts:

All texts are language-dependent. They are Unicode-compatible and support characters for all languages. Moreover, the number of characters in a text has no limit. Just like standard texts, these texts will be grouped under a tab in the Text Editor, which can also be used to create new texts. To get to this tab, go to *Home\Texts\Error Texts tab*.
A different section explains how to work with texts, including error texts, in detail – please refer to → "Working with the Text Editor ", page 640.

As soon as the text is longer than the defined display area on the device screen, it will be truncated on the right.

✔ **Needs confirmation**

If this option is enabled, the corresponding error will have to be confirmed with one of the following special functions: Error confirm, Error confirm all, or Error confirm (actual window). If the option is not enabled, a confirmation will not be required. When a confirmation is required, there are two different options that can be selected in *Home\Alarms\Register Handling tab*. These two options are different from each other in terms of when the confirmation will be required:

- Confirmation always possible
- Confirmation only possible if an error is inactive

Confirmed errors will be marked with an "X" in the History state column.

An error confirmation example is provided in a different section – please refer to → section "Confirming errors", page 853. Moreover, the special confirmation functions are described in detail in a different section – please refer to → section "Error confirm", page 883, → section "Error confirm all", page 884, and → section "Error confirm (actual window)", page 884.

Add to History

If this option is enabled, the error tag will be added to the history. The history will contain one entry:

- Time of error
- Time when error disappeared
- Time of confirmation if the Needs confirmation option is enabled in *Home\Alarms\Tags tab*.

The history will be stored in RAM memory. Special functions can be used to save its contents in non-volatile memory. In addition, special functions can be used to switch the view in the alarm list between the current errors and the history.

The following special functions are especially relevant within this context:

→ "Show History On/Off", page 889
→ "Save Error History as File", page 888
→ "Save Error History as File Var", page 888
→ "Print History", page 1033
→ "Clear all Histories", page 882

Add to user logger

If this checkbox is enabled, an entry for the error message will be added to the user logger → section "User Logger", page 942.

→ Only the first 200 characters of an error text will be saved in the user logger.

Parameter Type / Parameter Tag

In order to be able to provide device users with detailed and precise information on errors, it is a good idea to be able to expand the corresponding static error texts with a dynamic portion. This can be done by using placeholders. Placeholders are represented using the <%> string. If the <%> placeholder is used in the Error text field, the placeholder will be replaced by the contents of the selected parameter in the event that the corresponding error occurs.

If there are multiple placeholders in a single error text, only the first placeholder will be replaced.

The drop-down menu in the Parameter Type column must first be used to specify whether a placeholder will be used.

Parameter Type

- None
  A placeholder will not be used.
- Tag
  The placeholder will represent a tag of data type BIT, BYTE, WORD, DWORD, QWORD or FLOAT. The corresponding tag must be specified in the Parameter Tag column.
- Text
  The placeholder will represent a standard text. The ID for this text must be specified in the Parameter Tag column.

Parameter Tag

This drop-down menu will only be enabled if either the <Tag> or <Text> option has been selected in the Parameter Type column.

The drop-down menu will show all the tags matching the selection in the Parameter Type column. The selected tag will replace the <%> placeholder at runtime.

In order to ensure data integrity, the polling rate for the parameter tag must always be set to "as fast as possible."

Example illustrating the problem of potential data inconsistency:

- The polling rate for an error tag is set to 1 sec.
- The polling rate for the corresponding parameter tag is set to 5 sec.
- If an error is triggered, it is possible for the memory to still contain an old parameter tag value. This is because the last parameter tag read operation may have only been 3 seconds ago, for example.

You should also make sure that the parameter tag value is already set correctly on the PLC when the error is triggered (and will not be set after the fact instead).

10.5 Alarm management functions

Alarm management functions include functions for updating, navigating through, and sorting alarm list visualization objects, as well as functions for confirming errors and showing help information in the event of an error. In addition, the Show Tag Help function can be used to provide device operators with a help text for the selected entry in an alarm list. To run alarm list functions you will need either a script or a function key visualization object. You can design the button for this special function according to your specific needs, e.g., by adding a background color or image to it.

Display, navigation, and sorting functions for alarm lists:

→ "Error first line", page 884
→ "Error last line", page 885
→ "Error line down", page 885
→ "Error line up", page 886
→ "Error line extend", page 885
→ "Error page down", page 886
→ "Error page up", page 886
→ "Error Priority ON/OFF", page 887
→ "Error Autoscroll On/Off", page 883
→ "Filter", page 887

Error help functions:

→ "Error help", page 885

Error confirmation functions:

→ "Error last line", page 885
→ "Error confirm", page 883
→ "Error confirm (actual window)", page 884
→ "Error confirm all", page 884

Functions used to display and save the error history:

→ "Clear System History", page 882
→ "Clear History in Screen", page 882
→ "Clear all Histories", page 882
→ "Save Error History as File", page 888
→ "Save Error History as File Var", page 888
→ "Save System History as File", page 888
→ "Save System History as File Var", page 889
→ "Show History On/Off", page 889
→ "Show Sys. History On/Off", page 889
→ "Save History to CSV", page 994
→ "Save History to CSV Var", page 994

→ "Save History of previous day to CSV", page 993

→ "Save History of previous day to CSV Var", page 994

Refresh functions

→ "Error Refresh", page 887

10.5.1 Clear all Histories

The history entries of the entire project are deleted. The history of the system error messages is preserved.

Number of the Function	22 (16h)
Executable from the PLC	Yes
Function with ON/OFF state	No
Function available in standard screens	Yes
Function available with keyboards	No
Function available in script	Yes
Function available in application entry script	Yes

10.5.2 Clear History in Screen

Clears the history for the alarm list visualization object on the current screen. This function will not do anything in screens without an alarm list visualization object.

→ If other alarm list visualization objects use the same tags, the history entries will also be cleared for those alarm lists!

Number of the Function	21 (15h)
Executable from the PLC	Yes
Function with ON/OFF state	No
Function available in standard screens	Yes
Function available with keyboards	No
Function available in script	Yes
Function available in application entry script	No

10.5.3 Clear System History

Clears the history for the system error messages. System error messages are messages that are not generated in the project, but that are instead generated by the sequence program (GRS). It is possible to freely define which system error messages will be logged. To do so, the corresponding checkbox needs to be enabled in *Home\Settings\System Messages* tab.

If the system error message history is cleared, the project error message history will be retained.

Number of the Function	24 (18h)
Executable from the PLC	Yes
Function with ON/OFF state	No

Function available in standard screens	Yes
Function available with keyboards	No
Function available in script	Yes
Function available in application entry script	Yes

10.5.4 Error Autoscroll On/Off

This function is used together with alarm list visualization objects. By default, an alarm list will automatically scroll together with the most recent error. This automatic scrolling can be enabled and disabled with this function.

After a screen Change, automatic scrolling is reactivated.

→ If a script performs a screen change, the last setting will be retained.

Number of the Function	13 (0dh)
Executable from the PLC	No
Function with ON/OFF state	Yes
Function available in standard screens	Yes
Function available with keyboards	No
Function available in script	Yes
Function available in application entry script	No

10.5.5 Error confirm

This function can be used to confirm a selected entry in an alarm list.

If it is necessary to confirm errors for the selected error tag and the error tag has a value of <0>, the error entry will be removed from the alarm list visualization object.

To do this, it is necessary to navigate through the alarm list visualization object and select the relevant line. After this, the function key that runs the Error confirm function needs to be tapped on in order to confirm the error.

Error message statuses are indicated as follows:

- 1 = Error present
- 0 = Error no longer present
- X = Error has been confirmed

If it is not necessary to confirm errors for the selected error tag, or if the entry cannot be currently confirmed because the error is still active, this function will not do anything – please refer to → section "Needs confirmation", page 861 and → section "Handling", page 863

Whether or not an error message is confirmed will not have any effect on the corresponding error tag's state.

For an example showing how to implement error confirmations, please refer to the → section "Confirming errors", page 853 section

Number of the Function	10 (0Ah)
Executable from the PLC	No
Function with ON/OFF state	No
Function available in standard screens	Yes
Function available with keyboards	No
Function available in script	No
Function available in application entry script	No

10.5.6 Error confirm (actual window)

Confirms the error messages in the alarm list visualization object on the current screen.

Number of the Function	15 (0Fh)
Executable from the PLC	No
Function with ON/OFF state	No
Function available in standard screens	Yes
Function available with keyboards	No
Function available in script	Yes
Function available in application entry script	No

10.5.7 Error confirm all

This function confirms all existing error messages with a state of <0>. This function does not apply to errors within a single alarm list only, but to the entire project instead.

Number of the Function	11 (0Bh)
Executable from the PLC	Yes
Function with ON/OFF state	No
Function available in standard screens	Yes
Function available with keyboards	No
Function available in script	Yes
Function available in application entry script	Yes

10.5.8 Error first line

This function selects the first entry in an alarm list visualization object.

Number of the Function	4 (04h)
Executable from the PLC	No
Function with ON/OFF state	No
Function available in standard screens	Yes
Function available with keyboards	No
Function available in script	No
Function available in application entry script	No

10.5.9 Error help

This function shows the help text for the error tag that is currently selected in an alarm list. In order for this help text to be displayed, it is necessary for the error tag to be set up in the tag help object and for a help text and help screen to be assigned to it. If there is no help screen defined in the current screen, or if a help text has not been assigned to the selected error message, this function will not do anything. Help texts can be entered in *Home\Tag Help*.

Number of the Function	12 (0ch)
Executable from the PLC	No
Function with ON/OFF state	No
Function available in standard screens	Yes
Function available with keyboards	No
Function available in script	No
Function available in application entry script	No

10.5.10 Error last line

This function selects the last entry in an alarm list visualization object.

Number of the Function	5 (05h)
Executable from the PLC	No
Function with ON/OFF state	No
Function available in standard screens	Yes
Function available with keyboards	No
Function available in script	No
Function available in application entry script	No

10.5.11 Error line down

This functions selects the next entry in an alarm list visualization object. If the entry currently selected is the bottommost entry in the alarm list visualization object, the list will be moved one line down.

Number of the Function	1 (01h)
Executable from the PLC	No
Function with ON/OFF state	No
Function available in standard screens	Yes
Function available with keyboards	No
Function available in script	No
Function available in application entry script	No

10.5.12 Error line extend

This function is intended for long error texts that cannot be fully displayed in a few lines in an alarm list object. When the function is run, the error text for the selected entry in the alarm list will be shown on a help screen with the help of a help information visualization object. For this purpose, a help screen with a help information visualization object must be created. In the Properties 'Help Information' pane, the <Error text> option must be selected in the Information type field.

→ Other information types, such as Help text or Image, are not allowed.

If a help screen has not been assigned to the current screen, this function will not do anything.

Number of the Function	16 (10h)
Executable from the PLC	No
Function with ON/OFF state	No
Parameter 1	The help screen that should show the error text
Function available in standard screens	Yes
Function available with keyboards	No
Function available in script	No
Function available in application entry script	No

10.5.13 Error line up

This functions selects the previous entry in an alarm list visualization object. If the entry currently selected is the topmost entry in the alarm list visualization object, the list will be moved one line up.

Number of the Function	0 (00h)
Executable from the PLC	No
Function with ON/OFF state	No
Function available in standard screens	Yes
Function available with keyboards	No
Function available in script	No
Function available in application entry script	No

10.5.14 Error page down

This function selects the entry that is one page (window height) down in the alarm list visualization object. The selected entry will then be the bottommost line shown or the last entry (if the object is not full) in the alarm list visualization object.

Number of the Function	3 (03h)
Executable from the PLC	No
Function with ON/OFF state	No
Function available in standard screens	Yes
Function available with keyboards	No
Function available in script	No
Function available in application entry script	No

10.5.15 Error page up

This function selects the entry that is one page (window height) up in the alarm list visualization object. The selected entry will then be the topmost line shown.

Number of the Function	2 (02h)
Executable from the PLC	No
Function with ON/OFF state	No
Function available in standard screens	Yes
Function available with keyboards	No
Function available in script	No
Function available in application entry script	No

10.5.16 Error Priority ON/OFF

Defines the order for the entries in an alarm list. The corresponding function key will be in the OFF state by default.

ON: Entries will be sorted by priority

OFF: Entries will be sorted chronologically

Number of the Function	14 (0Eh)
Executable from the PLC	Yes
Function with ON/OFF state	Yes
Function available in standard screens	Yes
Function available with keyboards	No
Function available in script	Yes
Function available in application entry script	No

10.5.17 Error Refresh

If error tags are used in scripts, the Error refresh function must be called beforehand so that the values will be refreshed by the PLC.

→ The Error Refresh command should not be run cyclically, as doing so will overload communications.

Function available in script	Yes
Function available in application entry script	Yes

10.5.18 Filter

Filters the entries of the alarm list based on the content of the specified tag under Filter variable in the properties of the visualization object alarm list.

The filtering remains active until the function is executed again. Thus, live filtering is possible by changing the string variable.

Number of the Function	220 (00DCh)
Executable from the PLC	No
Function with ON/OFF state	Yes
Function available in standard screens	Yes
Function available with keyboards	No
Function available in script	Yes
Function available in application entry script	No

10.5.19 Save Error History as File

Saves the error history in a file. The filename (including the path) is passed as the first argument for the function.

Number of the Function	206 (CEh)
Executable from the PLC	No
Function with ON/OFF state	No
Parameter 1	Name of the file (including the path) where the data should be saved.
Function available in standard screens	Yes
Function available with keyboards	No
Function available in script	Yes
Function available in application entry script	No

10.5.20 Save Error History as File Var

Saves the error history in a file. The filename must be specified together with the corresponding path and extension. The filename is passed as the first argument for the function.

→ Important:
The file name must be entered with the path, the file name and the suffix.

Number of the Function	210 (D2h)
Executable from the PLC	No
Function with ON/OFF state	No
Parameter 1	String tag with path, filename, and extension where the data should be stored.
Function available in standard screens	Yes
Function available with keyboards	No
Function available in script	Yes
Function available in application entry script	Yes

10.5.21 Save System History as File

Save the system history in a file. The filename (including the path) is passed as the first argument for the function.

Number of the Function	207 (CFh)
Executable from the PLC	No
Function with ON/OFF state	No
Parameter 1	Name of the file (including the path) where the data should be saved.
Function available in standard screens	Yes
Function available with keyboards	No
Function available in script	Yes
Function available in application entry script	No

10.5.22 Save System History as File Var

Save the system history in a file. The file name (incl. path, file name and suffix) is transferred as the first parameter of the function.

→ Important:
The file name must be entered with the path, the file name and the suffix.

Number of the Function	211 (D3h)
Executable from the PLC	No
Function with ON/OFF state	No
Parameter 1	String tag with path, filename, and extension where the data should be stored.
Function available in standard screens	Yes
Function available with keyboards	No
Function available in script	Yes
Function available in application entry script	Yes

10.5.23 Show History On/Off

This function switches the alarm list visualization object to the history display. The error history is used to log error activities.

The functions for scrolling the list are the same as for the error display (Error line up, Error line down, etc.).

This setting is always deactivated after a screen Change.

→ If a script performs a screen change, the last setting will be retained.

Number of the Function	20 (14h)
Executable from the PLC	Yes
Function with ON/OFF state	Yes
Function available in standard screens	Yes
Function available with keyboards	No
Function available in script	Yes
Function available in application entry script	No

10.5.24 Show Sys. History On/Off

This function switches the alarm list visualization object to the history display for the system error messages.

This setting is always deactivated after a screen Change.

→ If a script performs a screen change, the last setting will be retained.

Number of the Function	137 (89h)
Executable from the PLC	Yes

Function with ON/OFF state	Yes
Function available in standard screens	Yes
Function available with keyboards	No
Function available in script	Yes
Function available in application entry script	No

For more information on which error messages are recorded in the history, please refer to → section "System Messages", page 1147.

11. User Management

Setting up user management

GALILEO features comprehensive user management functions.
This means, for example, that device operators can create and delete users and group users into previously defined groups at runtime. In addition, there are a variety of security options, including minimum password length requirements, a password history function, password aging, starting passwords, an automatic logout function, and the ability to monitor the system for invalid login attempts.

These user management features are a powerful tool for preventing unauthorized access to devices.

Users and their permissions can be easily and conveniently managed and configured, ensuring that security can be achieved by configuring just a few settings. The management concept revolves around the use of users and groups. A maximum of 200 groups and 500 users can be set up.

All rights are defined in the groups.
The Group membership must be defined for each user so that the rights can be assigned to the user.

→ Each user can only belong to ONE group!
However, groups can have more than one user!

These groups are not hierarchical. All groups must be defined in GALILEO – groups cannot be deleted or created on the device. Users can be deleted and created on the device.

The user management functionality comes with multiple security features that provide additional security when working with a device:

- Minimum length of the password
- Password history
- Start Password
- Password Aging
- Auto Logout
- Monitoring the system for invalid login attempts

User management is of crucial importance to projects. Because of this, it is a good idea to think in advance about what kinds of things you want to restrict in your project and how. In most cases, assigning permissions in a hierarchical manner will do the trick.

Here are some ideas and questions:

- What should be restricted in the project?
- Are the rights hierarchical?
- Who is the administrator?

- Are users created or deleted on the device?
- Do several users have the same rights?

11.1 Managing users

To set up an efficient user management system in the project, follow the steps below:

1. Create a group for each group of users that has the exact same permissions. Start with the group that has the most restrictions and the fewest permissions.
2. Create an administrator group, which has the most rights.
3. Define the rights of the individual groups.
4. Remember, the administrator must also be able to manage himself.
5. Create the already known users.
6. Assign each user to a group.

11.2 Properties User management

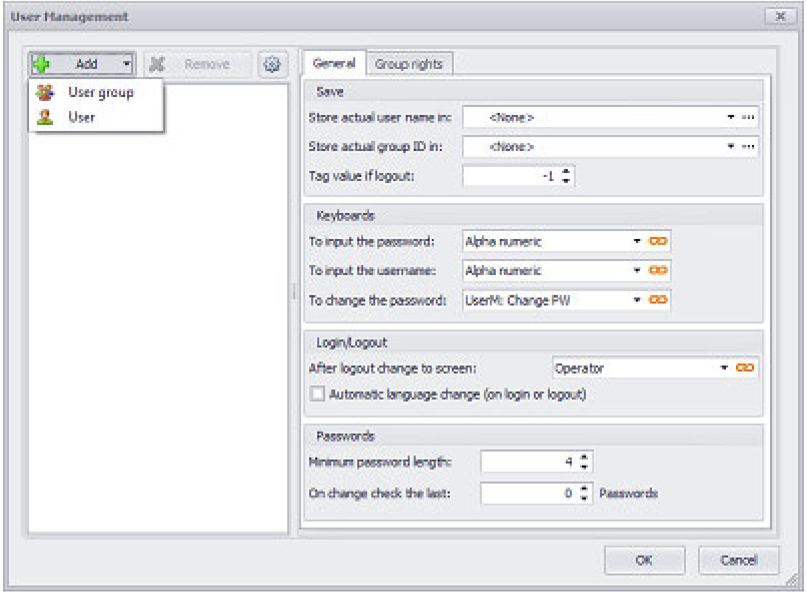
Adding users and user groups

Click on the Add button to create a user group for the project. Users can then be assigned to the user groups you create.

→ Usernames are not case-sensitive.

Clicking on ⚙ will show the General and Group rights tabs. These tabs can be used to configure the system's global settings for access permissions.

Home\Project Configuration\ User management



Save

Store actual user name in: <None>

Store actual group ID in: <None>

Tag value if logout: -1

Keyboards

To input the password: Alpha numeric

To input the username: Alpha numeric

To change the password: UserM: Change PW

Login/Logout

After logout change to screen: Operator

□ Automatic language change (on login or logout)

Passwords

Minimum password length: 4

On change check the last: 0 Passwords

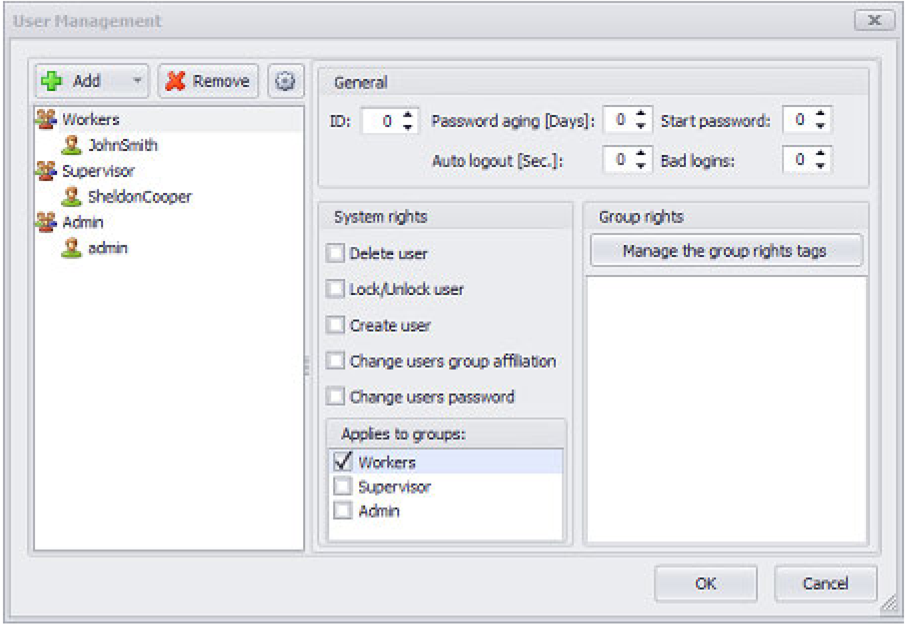
Figure 558: User Management dialog box showing how additional users can be added

User group

Groups are used to define security options and permissions that apply to multiple users. These groups are not hierarchical. All groups must be defined in GALILEO – groups cannot be deleted or created on the device.

→ Each user can only belong to ONE group!
However, groups can have more than one user!

Home\Project Configuration\ User management



ID: 0
Password aging [Days]: 0
Start password: 0
Auto logout [Sec.]: 0
Bad logins: 0

System rights

□ Delete user
□ Lock/Unlock user
□ Create user
□ Change users group affiliation
□ Change users password

Applies to groups:

☑ Workers
□ Supervisor
□ Admin

Figure 559: Settings for the group rights for <Workers>

Group

The group name. These names are case-sensitive. To edit a user name, double-click on it or click on it once and start typing.

General section

ID

An ID of your choice. This ID is used together with Store actual group ID in to store the current group ID in a tag.

Password aging [Days]

Used to define the time after which a password will become invalid. If a user tries to log in without changing the password before this period elapses, they will be asked to set a new password. This setting can be defined as a fixed constant or dynamically as a tag.

→ Note for dynamic values: Password aging will be disabled if the dynamic tag's value is 0.

Star password

If the value in the Start password field is set to <1>, this security option will be enabled. This means that a newly created user will be asked to change their password the first time they log in. This can come in handy when a supervisor creates a new user but should not know that user's password. This setting can be defined as a fixed constant or dynamically as a tag.

→ Note for dynamic values:
The request for setting a password the first time a user logs in will be disabled if the dynamic tag has a value of <0>.

Auto logout [Sec.]

If this option is enabled, users will be automatically logged out after x seconds of inactivity, i.e., the group rights tags will be reset as per the setting selected for the Set the Bit-Tag of the activated groups.... This setting can be defined as a fixed constant or dynamically as a tag.

→ Note for dynamic values:
The automatic logout function will be disabled if the tag has a value of 0.

Bad logins

If this option is enabled, the system will monitor the number of times that the Login User function is run and is followed by an invalid password. If the number of invalid login attempts set here is exceeded, the user will no longer be able to log in and their account will be locked. Locked users can be unlocked with the Lock/Unlock User function. This setting can be defined as a fixed constant or dynamically as a tag.

→ Note for dynamic values: The monitoring of the bad logins is inactive if the value of the tag is 0.

System rights section

The system rights section is used to configure administrative rights. These rights will apply to the user that is currently logged in and refer to the members of the corresponding target groups.

All system rights refer to members of this target groups.

Example:

Say you have a group called "Supervisor". You could assign users belonging to this group the right to create, delete, and lock/unlock any user in the "Supervisor," "Instructor," Operator," and "Service," groups,
as well as to change their password and the group to which they belong.

→ Defining a user's own group as a target group is only advisable for administrators!

Delete user

Defines the right of a user to delete a user that is member of the target group.

Lock/Unlock user

Defines the rights of a user to lock or unlock a user that is member of the target group.

Create user

Defines the right to create a user for the target groups.

Change users group affiliation

Defines the rights of a user to change the group affiliation of a user that is member of the target group.

Change users password

Defines the rights of a user to change the password of a user that is a member of the target group.

Applies to groups

The options enabled in the System rights section will apply to the members of the groups for which the corresponding checkbox is enabled.

Group rights

Group rights are bit tags. When a user logs in, the bit tags for the group to which the user belongs will be set. Likewise, these tags will be reset after the user logs out.

Manage the group rights tags

Variables

Bit tags

Set the Bit-Tag of the enabled groups

to Off (=0)

Negative logic: All selected tags will be set to 0. All other tags will be set to 1.

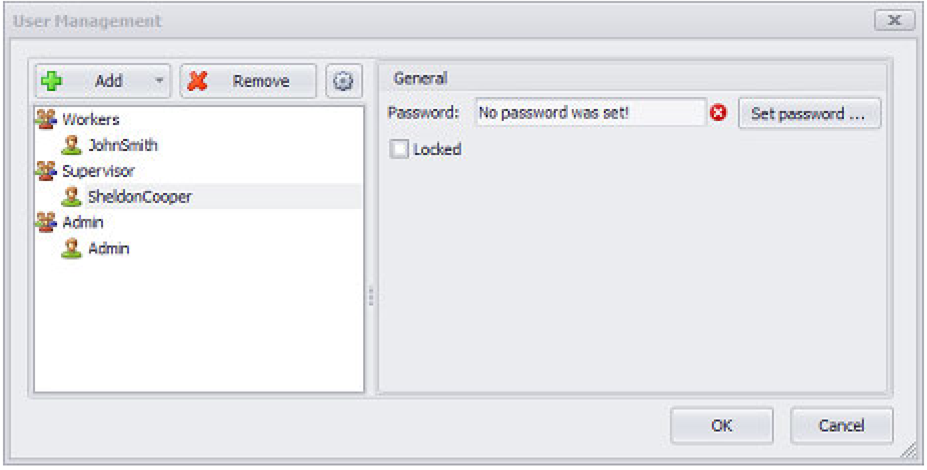
to On (=1)

Positive logic: All selected tags will be set to 1. All other tags will be set to 0.

User

Users can be deleted and created on the device. In order to set a password, a user must first be selected.

Home\Project Configuration\ User management



Password: No password was set!

□ Locked

Figure 560: User permission settings after selecting <SheldonCooper>

User

The username. These passwords are case-sensitive. To edit a user name, double-click on it or click on it once and start typing.

Password

Used to set the user's password. These passwords are case-sensitive. This setting can be defined as a fixed constant or dynamically as a tag.

→ Already defined passwords are only shown with "*" can cannot be displayed in plain text.

→ Users with dynamic passwords cannot be created on the device.

Locked

If this checkbox is enabled, the user will be locked. The Lock/Unlock User function can be used to unlock a locked user. In order to be able to do this, however, the user that is logged in must have the required "Lock/Unlock user" system right – please refer to → section "Lock/Unlock User", page 909.

Set Password

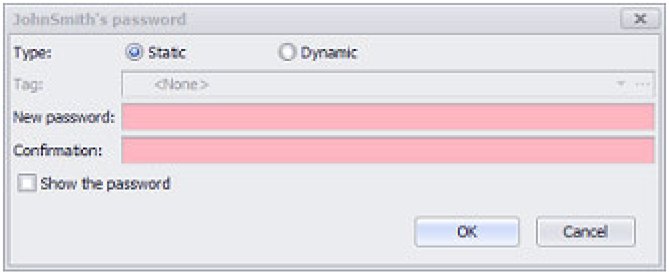
There are two ways to assign a password:

Static

A static password is a set password that doesn't change. This password needs to be entered both into the New password and Confirmation fields. After this password is set, the corresponding user will need to use it whenever logging in. If the Start password option is set to <1> for the user's group, the assigned static password will only be valid the first time the user logs in, after which they will be asked to enter a new password.

Dynamic

In this case, the password is defined using a tag, meaning that the password will be dynamic. To do this, you will need to select the Dynamic option for the Type setting. After this, select a tag of data type STRING using the drop-down menu. The user will have to log in with the password matching the value in the string tag.


JohnSmith's password

Type: ● Static ○ Dynamic

Tag: <None>

New password:

Confirmation:

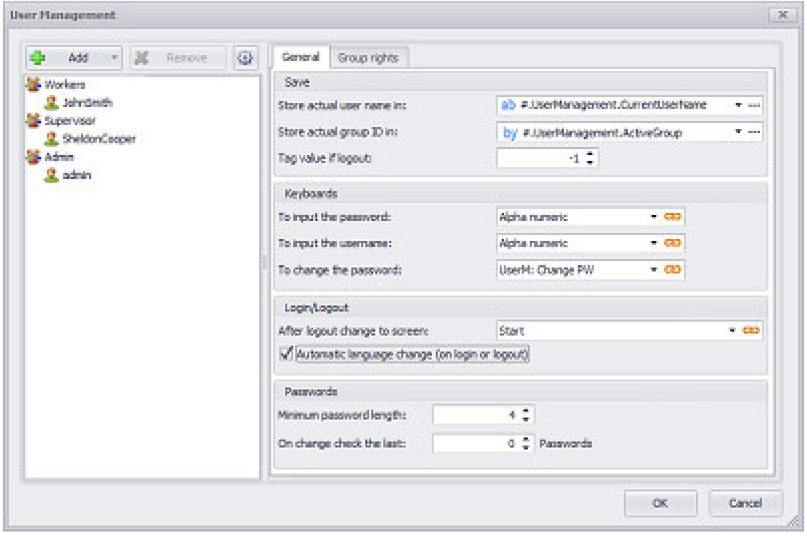
□ Show the password

OK Cancel

Figure 561: Dialog box for entering a user password

General

Home\Project Configuration\ User management



Save

Store actual user name in: #.UserManagement.CurrentUserName

Store actual group ID in: #.UserManagement.ActiveGroup

Tag value if logout: -1

Keyboards

To input the password: Alpha numeric

To input the username: Alpha numeric

To change the password: UserM: Change PW

Login/Logout

After logout change to screen: Start

☑ Automatic language change (on login or logout)

Passwords

Minimum password length: 4

On change check the last: 0 Passwords

Figure 562: User Management dialog box\General

Save section

Save current user name in

The current username can be stored in a tag of data type STRING so that it can be sent to the PLC.

Store actual group ID in

The current group ID can be stored in a tag so that it can be sent to the PLC. For example, the tag will have a value of 3 if group ID 3 is active.

Tag value if logout

If a tag is specified in the Store actual group ID in field, the value set here will be assigned to the tag as soon as a user logs out.

Keyboardssection

When a user management function is triggered with a function key, a set system keyboard will appear. Depending on the specific function that has been triggered, this system keyboard will have buttons that can be used to access additional keyboards, e.g., in order to enter a username or password. These keyboards can be configured as necessary. For example, when the Login User function is run, the following keyboard will appear:

The keyboard that will appear when the User or Password button is tapped on can be set in the fields found in the Keyboards section.

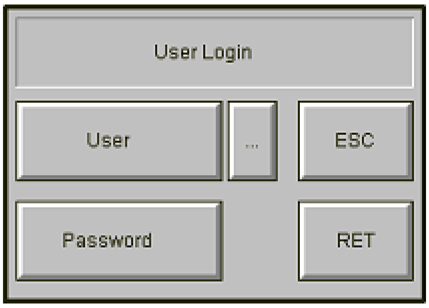


Figure 563: Keyboard for Login User function

The keyboards entered into the fields will appear on the device at runtime when the following actions are carried out:

- To input the password: These settings get applied when tapping on the Password , Password new or Password confirm button.
- To input the username: These settings get applied when tapping on the User button.
- To change the password: These settings get only applied if one of the following two cases happen:
  - The password has not been changed within the required period (Password aging > 0 days).
  - The password must be changed after the first time the user logs in (Start password = 1).

Both the Password aging and Start password options can be configured in *Home\User Management\<group>*.

Login/Logout section

After logout change to screen

With this option active the Touch Panel will switch automatically to the selected Screen after a logout.

There is a global setting for defining whether a visualization is 🔗 linked or C Ↄ unlinked. In order for the 🔗 and C Ↄ chain link icons to be displayed, there must be more than one visualization in a group.
For more information, please refer to: → section " Web visualization", page 1070

Automatic language change (on login or logout)

If this option is enabled and the Login User or Logout User function is run, the language will be changed. In the case of Login User, the language will change to the language that was last used by the user logging in. In the case of Logout User, the default language will be loaded. This default language can be set in *Home\Texts\Languages*.

If this option is not selected, no automatic language change will be executed.

Passwords section

Minimum password length

The minimum password length required when a password is changed or when a new user is created.

On change check the last x Passwords

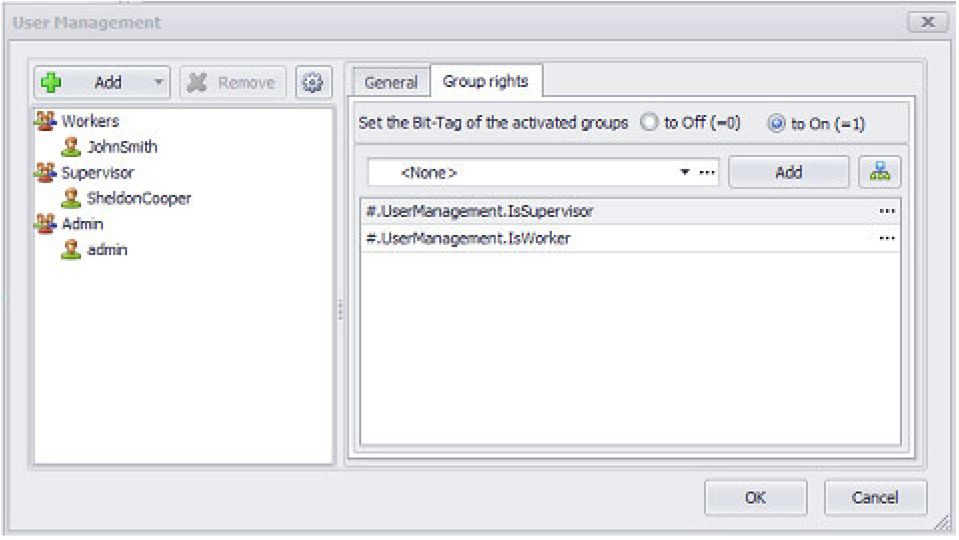
The password history stores the last x passwords. This prevents the reuse of an already existing password.

Group rights

The settings in this tab can be used to add tags of data type BIT that can later on be set as necessary for the groups. The same set logic applies to all the bit tags listed here: On or Off.

For the individual groups, you can later define which of the bit tags should be used by each group.

Home\Project Configuration\ User management



- Workers
  - JohnSmith
- Supervisor
  - SheldonCooper
- Admin
  - admin

Set the Bit-Tag of the activated groups ○ to Off (=0) ● to On (=1)

<None>

#.UserManagement.IsSupervisor

#.UserManagement.IsWorker

Figure 564: User Management dialog box\Group rights

11.3 User management functions

Almost all user management functions offer a keyboard that can be used to enter a username or password or to select a group. In this case, the keyboard will either be a preset GALILEO keyboard or a custom keyboard. If this option exists, the description for the function will indicate this for the corresponding button.

11.3.1 Change Group

Changes the group to which a user belongs. If this function is run by tapping on a function key, a system keyboard will appear.

Please note that only users for whom the corresponding system right has been enabled will be able to switch to a different group at runtime. For other users, the following system message will appear: <58> No permission to change group affiliations. For a description of system rights, please refer to → section "System rights section", page 896.

Number of the Function	1284 (504h)
Executable from the PLC	No
Function with ON/OFF state	No
Function available in Standard screens	Yes
Function available with keyboards	No
Function available in script	No
Function available in application entry script	No

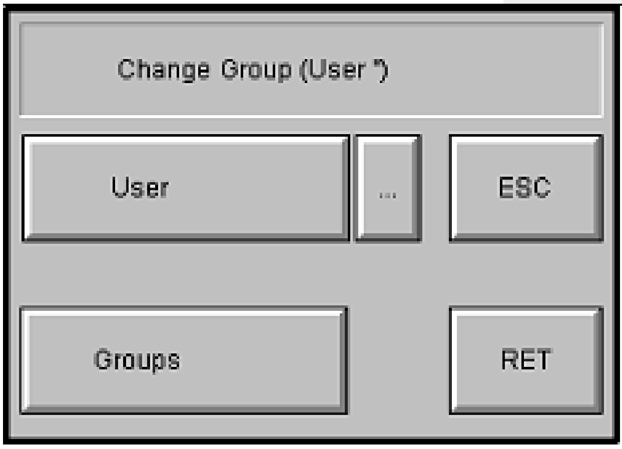


Figure 565: System keyboard appearing after the function key is tapped on

User — User Used to enter the username with the user input keyboard configured for this purpose.

... — Selection of an existing user.

Groups — Selection of a group in which the user is to become a member.

ESC — Exit the dialog without defining a new user.

RET — The new user is added into the list.

The keyboard can be project-programmed in *Home\User Management*⚙ *\General tab*→ section "Keyboardssection", page 899

11.3.2 Change Group -userdef.

Call of a user -defined keyboard to modify the group membership of a user.

Number of the Function	1316 (524h)
Executable from the PLC	No
Function with ON/OFF state	No
Parameter 1	Selected User Defined Keyboards
Function available in Standard screens	Yes
Function available with keyboards	No
Function available in script	No
Function available in application entry script	No

11.3.3 Change PW (User x)

The password of a specific user can be changed with this function.

Number of the Function	1290 (50Ah)
Executable from the PLC	No
Function with ON/OFF state	No
Parameter 1	User, whose password is to be changed
Function available in Standard screens	Yes
Function available with keyboards	No
Function available in script	No
Function available in application entry script	No

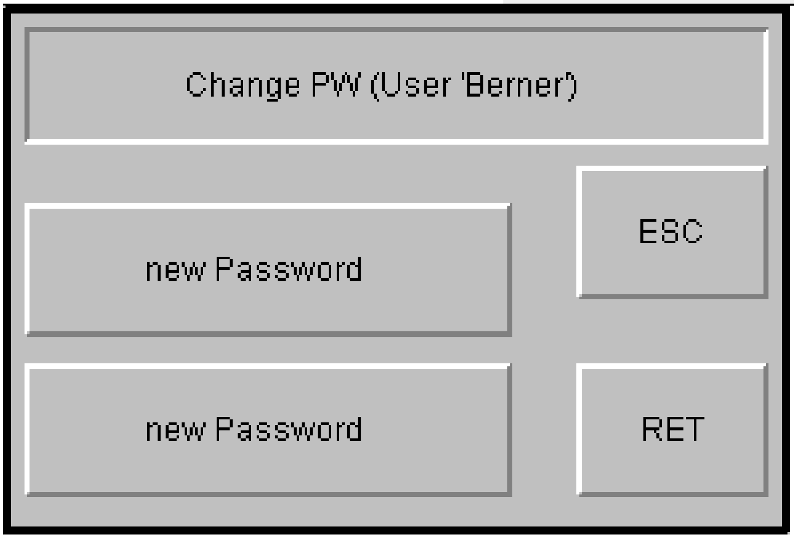


Figure 566: System keyboard appearing after the function key is tapped on

new Password Used to enter a new password with the password input keyboard.

new Password Confirm password. Used to check that the new password has been entered correctly.

If the passwords entered into the two new Password fields do not match, the old password will be kept.

ESC	Exits the dialog without changing the password.
RET	If the passwords entered into the two new Password match, the new password will be assigned to the user.

The keyboard can be project-programmed in *Home\User Management*⚙ *\General tab*→ section "Keyboardssection", page 899

11.3.4 Change PW (User x) -userdef.

Call of a user -defined keyboard to change a user password.

Number of the Function	1322 (52Ah)
Executable from the PLC	No
Function with ON/OFF state	No
Parameter 1	User, whose password is to be changed
Parameter 2	Selected User Defined Keyboards
Function available in Standard screens	Yes
Function available with keyboards	No
Function available in script	No
Function available in application entry script	No

11.3.5 Change PW (other User) -userdef.

Call of a user -defined keyboard to change a user password.

Number of the Function	1315 (523h)
Executable from the PLC	No
Function with ON/OFF state	No
Parameter 1	Selected User Defined Keyboards
Function available in Standard screens	Yes
Function available with keyboards	No
Function available in script	No
Function available in application entry script	No

11.3.6 Change PW (Own) -userdef.

Call of a user -defined keyboard for changing user's personal password.

Number of the Function	1314 (522h)
Executable from the PLC	No
Function with ON/OFF state	No
Parameter 1	Selected User Defined Keyboards
Function available in Standard screens	Yes
Function available with keyboards	No
Function available in script	No
Function available in application entry script	No

11.3.7 Change PW (other User)/ ChangePWOther

Changes the password of a user .

This function can only be used by logged-in users who are members of a group with the "Change users password" system right. In this case, the logged-in user will only be able to change the passwords of users who belong to a member of a group to which this system right applies.

Once the password has been changed successfully, the password tags' contents will be cleared due to security reasons.

Number of the Function	1283 (503h)
Executable from the PLC	No
Function with ON/OFF state	No
Function available in standard screens	Yes
Function available with keyboards	No
Function available in script	No
Function available in application entry script	Yes

Call via function key visualization object

The following dialog will appear if this function is activated:

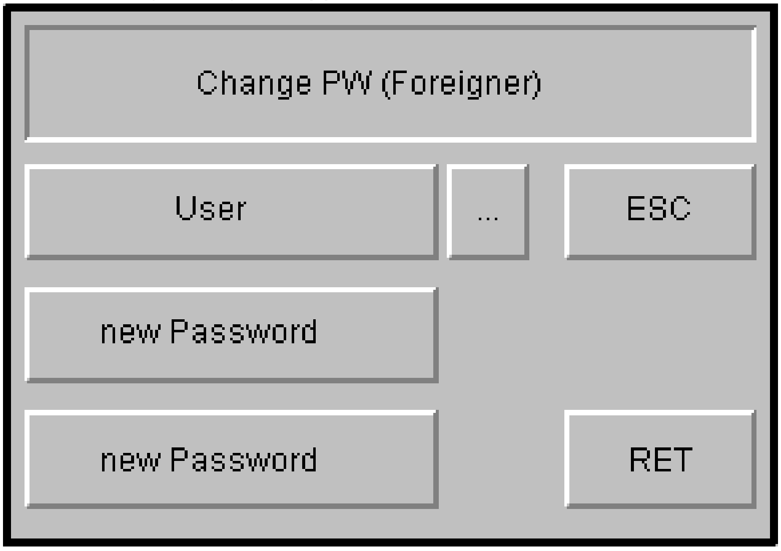


Figure 567: System keyboard appearing after the function key is tapped on

User
User Used to enter the username with the user input keyboard configured for this purpose.

...
Selection of an existing user.

new Password
new Password Used to enter a new password with the password input keyboard configured for this purpose.

new Password Used to enter a new password with the
Used to acknowledge the password with the password input keyboard configured for this purpose. Used to check that the new password has been entered correctly.

password input
keyboard
configured for
this purpose.

If the passwords entered into the two new Password fields do not match, the old password will be kept.

ESC	Exits the dialog without changing the password.
RET	If the passwords entered into the two new Password match, the new password will be assigned to the user.

The keyboard can be project-programmed in *Home\User Management\⚙ \General tab*→ section "Keyboardssection", page 899

Call via script

Parameter 1	User Name
Parameter 2	New Password
Parameter 3	Password Confirmation

11.3.8 Change PW (own)

Users can use this function to change their own password.

Prerequisites
A user must be logged in in order to be able to run this function.

Once the password has been changed successfully, the password tags' contents will be cleared due to security reasons.

Number of the Function	1282 (502h)
Executable from the PLC	No
Function with ON/OFF state	No
Function available in standard screens	Yes
Function available with keyboards	No
Function available in script	Yes
Function available in application entry script	Yes

Call via function key visualization object

The corresponding keyboard is defined in *Home\User Management\⚙ \General*, in the To change the password field.

The following dialog will appear if this function is activated:

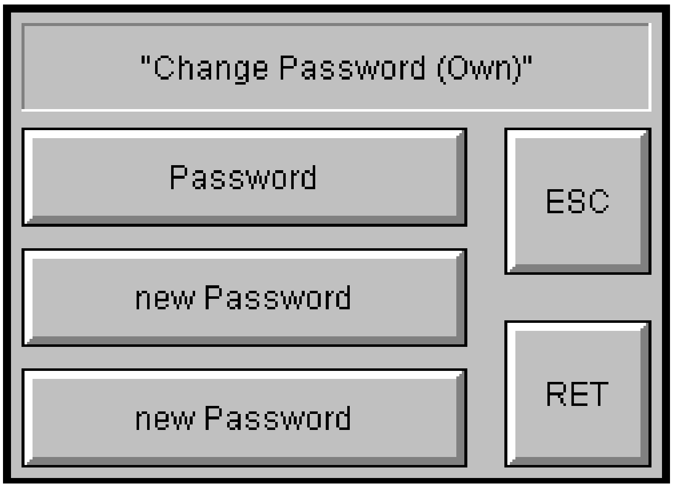


Figure 568: System keyboard appearing after the function key is tapped on

Password Used to enter the old password with the user input keyboard configured for this purpose

new Password new Password Used to enter a new password with the password input keyboard configured for this purpose.

new Password Used to acknowledge the password with the password input keyboard configured for this purpose. Used to check that the new password has been entered correctly.

If the passwords entered into the two new Password fields do not match, the old password will be kept.

ESC Exits the dialog without changing the password.

RET If the passwords entered into the two new Password match, the new password will be assigned to the user.

The keyboard can be project-programmed in *Home\User Management\ \General tab*→ section "Keyboardssection", page 899

Call via script

Parameter1	old password
Parameter2	new Password
Parameter3	Password Confirmation

11.3.9 Delete User -userdef.

Call of a user -defined keyboard for removing a user.

Function number	1318 (526h)
Executable from the PLC	No
Function with ON/OFF state	No
Parameter 1	Selected User Defined Keyboards
Function available in Standard screens	Yes

Function available with keyboards	No
Function available in script	No
Function available in application entry script	No

11.3.10 Delete User

Use this function to delete a user .

Number of the Function	1286 (506h)
Executable from the PLC	No
Function with ON/OFF state	No
Function available in Standard screens	Yes
Function available with keyboards	No
Function available in script	No
Function available in application entry script	No

The following dialog will appear if this function is activated:

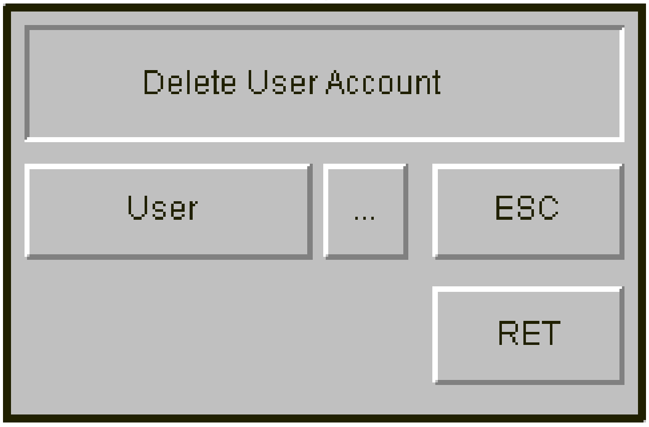


Figure 569: System keyboard appearing after the function key is tapped on

User User Used to enter the username with the user input keyboard configured for this purpose.

... Selection of an existing user.

ESC Exit dialog without logging in.

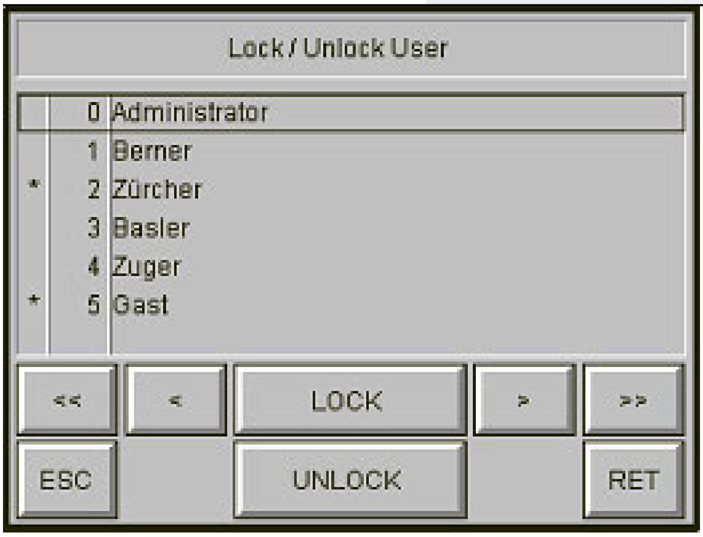
RET The selected user is deleted.

The keyboard can be project-programmed in *Home\User Management\⚙ \General tab*→ section "Keyboardssection", page 899

11.3.11 Lock/Unlock User

This function shows a list of all users. The asterisk (*) in the first column indicates whether a user is locked. Users will be automatically locked if they try to log in with the wrong password too many times and the option for monitoring the system for invalid login attempts is enabled. The maximum number of invalid login attempts can be configured in *Home\User Management*⚙ *\General*, in the Bad logins field.

Number of the Function	1328 (530h)
Executable from the PLC	No
Function with ON/OFF state	No
Function available in Standard screens	Yes
Function available with keyboards	No
Function available in script	No
Function available in application entry script	No


Lock / Unlock User

	0	Administrator
	1	Berner
*	2	Zürcher
	3	Basler
	4	Zuger
*	5	Gast

Figure 570: System keyboard appearing after the function key is tapped on

LOCK

Locks the currently selected user.

UNLOCK

Unlocks a locked user.

Button : <<
Moves inside the selection list one page up.

Button : <
Moves inside the selection list one Item up.

Button : >
Moves inside the selection list one Item down.

Button : >>
Moves inside the selection list one page bottom.

ESC

Exits the dialog without changing the password.

RET

If both password entries match, the current user is assigned the new password.

11.3.12 Lock/Unlock User -userdef.

Call of a user -defined keyboard to lock or unlock a user.

Number of the Function	1329 (531h)
Executable from the PLC	No
Function with ON/OFF state	No
Parameter 1	Selected User Defined Keyboards
Function available in Standard screens	Yes
Function available with keyboards	No
Function available in script	No
Function available in application entry script	No

11.3.13 Login User

Logging in with a user and the corresponding password. After this function is called successfully, the language that the active user last used will be loaded. Language 0 is loaded if the user was not yet logged in.

Number of the Function	1280 (500h)
Executable from the PLC	No
Function with ON/OFF state	No
Function available in Standard screens	Yes
Function available with keyboards	No
Function available in script	Yes
Function available in application entry script	Yes

The following dialog will appear if this function is activated:

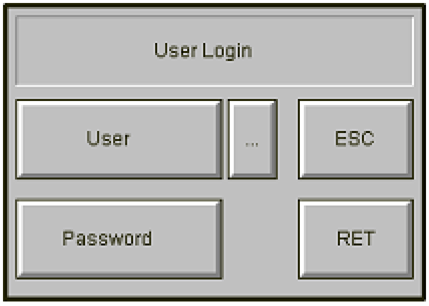


Figure 571: System keyboard appearing after the function key is tapped on

User	User Used to enter the username with the user input keyboard configured for this purpose.
...	Selection of an existing user.
Password	Used to enter the password with the password input keyboard configured for this purpose.

ESC	Exit dialog without logging in.
RET	If the correct user and password are entered, the corresponding bit tags are set or cleared due to the group membership.

The keyboard can be project-programmed in *Home\User Management* ⚙ *\General tab*→ section "Keyboardssection", page 899

11.3.14 Login User -userdef.

Call of a user -defined keyboard for logging in a user and the corresponding password.

Number of the Function	1312 (520h)
Executable from the PLC	No
Function with ON/OFF state	No
Parameter 1	Selected User Defined Keyboards
Function available in Standard screens	Yes
Function available with keyboards	No
Function available in script	Yes
Function available in application entry script	Yes

11.3.15 Login User x

Logging in a specific user and the corresponding password.

Number of the Function	1287 (507h)
Executable from the PLC	No
Function with ON/OFF state	No
Parameter 1	User to be logged in
Function available in Standard screens	Yes
Function available with keyboards	No
Function available in script	Yes
Function available in application entry script	Yes

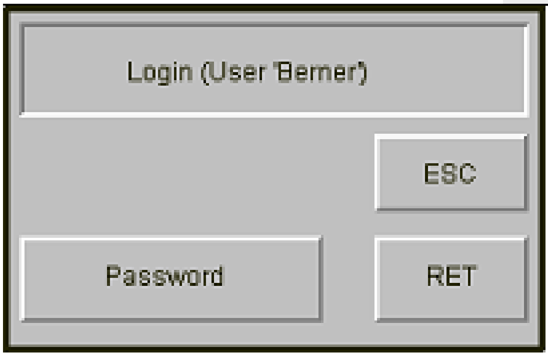


Figure 572: System keyboard appearing after the function key is tapped on

Password

Used to enter the password with the password input keyboard configured for this purpose.

The keyboard can be project-programmed in *Home\User Management*⚙ *\General tab*→ section "Keyboardssection", page 899

ESC

Exit dialog without logging in.

RET

If the correct user and password are entered, the corresponding bit tags are set or cleared due to the group membership.

11.3.16 Login User x -userdef.

Opens a custom keyboard for logging in a specific login user x - user-defined user.

Number of the Function	1319 (527h)
Executable from the PLC	No
Function with ON/OFF state	No
Parameter 1	User to be logged in
Parameter 2	Selected User Defined Keyboards
Function available in Standard screens	Yes
Function available with keyboards	No
Function available in script	Yes
Function available in application entry script	Yes

11.3.17 Logout User

Use this function to log out the currently active user . Calling this function deletes all bit Tags of the corresponding group.

If the Automatic language change (on login or logout) option is enabled, the user interface will switch to the default language when the Logout User function is run. The Automatic language change (on login or logout) option can be configured in *Home\User Management*⚙ *\General*, in the Login/Logout section.

Number of the Function	1281 (501h)
Executable from the PLC	Yes
Function with ON/OFF state	No
Function available in Standard screens	Yes
Function available with keyboards	No
Function available in script	Yes
Function available in application entry script	No

11.3.18 New User -userdef.

Call of a user-defined keyboard for defining a new user.

Number of the Function	1317 (525h)

Executable from the PLC	No
Function with ON/OFF state	No
Parameter 1	Selected User Defined Keyboards
Function available in Standard screens	Yes
Function available in keyboards	No
Function available in script	No
Function available in application entry script	No

11.3.19 New User

Define a new user. In the currently active group, the system right Group Change group association must be activated. The group of the new user must also be contained in the target groups. If 500 users are already defined, no additional user can be defined.

Number of the Function	1285 (505h)
Executable from the PLC	No
Function with ON/OFF state	No
Function available in standard screens	Yes
Function available with keyboards	No
Function available in script	Yes
Function available in application entry script	Yes

Call via function key visualization object

The following dialog will appear if this function is activated:

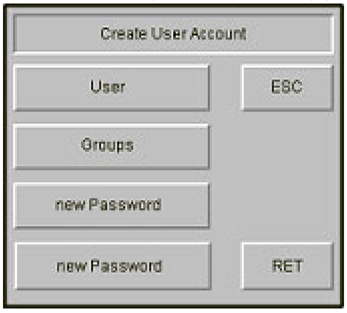


Figure 573: System keyboard appearing after the function key is tapped on

User — Used to enter the username with the user input keyboard configured for this purpose.

Groups — Selection of a group for membership.

new Password — Used to enter a password with the password input keyboard.

new Password — Used to acknowledge the password with the password input keyboard. Used to check that the new password has been entered correctly.

If the passwords entered into the two new Password fields do not match, the old password will be kept.

ESC	Exit the dialog without defining a new user.
RET	The new user is added into the list.

The keyboard can be project-programmed in *Home\User Management\⚙ \General tab*→ section "Keyboardssection", page 899

Call via script

Parameter entered	Name of new user
Parameter entered	Group of the new user
Parameter entered	Password of new user
Parameter entered	Password Confirmation

11.3.20 RefreshAutoLogoutTime

If the Auto Logout Time of the defined User group is dependent on a tag, the time will be refreshed with the current value of the tag.

Numbers of the Functions	1293 (50dh)
Executable from the PLC	Yes
Function with ON/OFF state	No
Parameter 1	The user group for which the auto logout time should be updated.
Function available in Standard screens	Yes
Function available in keyboards	No
Function available in script	Yes
Function available in application entry script	Yes
Return parameter	Time in milliseconds

11.3.21 Select Group

This function can be used to select a group, and will save the group name in the string tag.

Number of the Function	1338 (53Ah)
Executable from the PLC	No
Function with ON/OFF state	No
Parameter 1	String tag for saving the selected tag name.
Function available in Standard screens	Yes
Function available in keyboards	No
Function available in script	No
Function available in application entry script	No

11.3.22 Select Group -userdef.

This function calls a custom keyboard for selecting a group.

Number of the Function	1339 (53Bh)
Executable from the PLC	No
Function with ON/OFF state	No

Parameter 1	String tag for saving the selected tag name.
Parameter 2	Selected User Defined Keyboards
Function available in Standard screens	Yes
Function available in keyboards	No
Function available in script	No
Function available in application entry script	No

11.3.23 Select User

This function can be used to select a user and saves the username in the string tag.

Number of the Function	1292 (50ch)
Executable from the PLC	No
Function with ON/OFF state	No
Parameter 1	String tag for saving the selected username.
Function available in Standard screens	Yes
Function available in keyboards	No
Function available in script	No
Function available in application entry script	No

11.3.24 Select User -userdef.

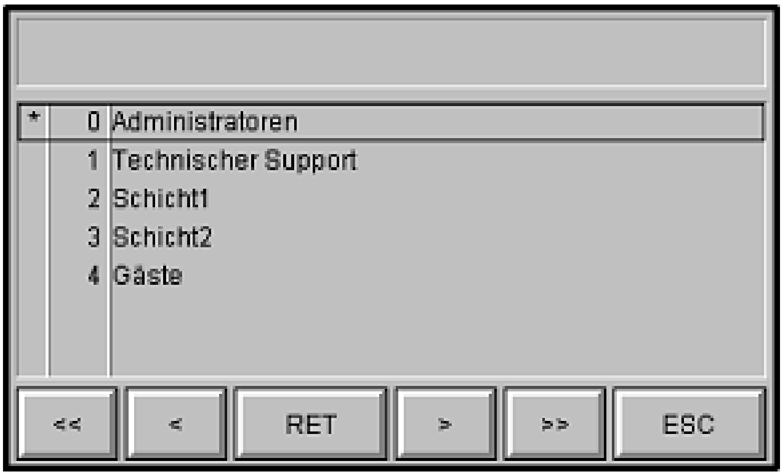
This function calls a custom keyboard for selecting a user.

Number of the Function	1324 (52ch)
Executable from the PLC	No
Function with ON/OFF state	No
Parameter 1	String tag for saving the selected username.
Parameter 2	Selected User Defined Keyboards
Function available in Standard screens	Yes
Function available in keyboards	No
Function available in script	No
Function available in application entry script	No

11.3.25 Show_all_Groups

Shows all groups in a list.

Number of the Function	1332 (534h)
Executable from the PLC	No
Function with ON/OFF state	No
Function available in Standard screens	Yes
Function available in keyboards	No
Function available in script	No
Function available in application entry script	No



*	0	Administratoren
	1	Technischer Support
	2	Schicht1
	3	Schicht2
	4	Gäste

Figure 574: System keyboard appearing after the function key is tapped on

11.3.26 Show all Groups -userdef.

Call of a user-defined keyboard to display all groups.

Number of the Function	1333 (535h)
Executable from the PLC	No
Function with ON/OFF state	No
Parameter 1	Selected User Defined Keyboards
Function available in Standard screens	Yes
Function available in keyboards	No
Function available in script	No
Function available in application entry script	No

11.3.27 Show all Users of Group

Shows all the users that are members of the group specified with parameter 1.

Number of the Function	1336 (538h)
Executable from the PLC	No
Function with ON/OFF state	No
Parameter 1	The group for which the members should be displayed.
Function available in Standard screens	Yes
Function available in keyboards	No
Function available in script	No
Function available in application entry script	No

11.3.28 Show all Users of Group -userdef.

Call of a user-defined keyboard to display all members of a group.

Number of the Function	1337 (539h)
Executable from the PLC	No
Function with ON/OFF state	No
Parameter 1	The group for which the members should be displayed.

Parameter 2	Selected User Defined Keyboards
Function available in Standard screens	Yes
Function available in keyboards	No
Function available in script	No
Function available in application entry script	No

11.3.29 Show All Users -userdef.

Call of a user-defined keyboard to show all users.

Number of the Function	1323 (52Bh)
Executable from the PLC	No
Function with ON/OFF state	No
Parameter 1	Selected User Defined Keyboards
Function available in Standard screens	Yes
Function available in keyboards	No
Function available in script	No
Function available in application entry script	No

11.3.30 Show Groups of User

Shows the group membership of a user-definable User.

Number of the Function	1334 (536h)
Executable from the PLC	No
Function with ON/OFF state	No
Function available in Standard screens	Yes
Function available in keyboards	No
Function available in script	No
Function available in application entry script	No

11.3.31 Show Groups of User -userdef.

Opens a custom keyboard for showing all the groups to which the user belongs.

Number of the Function	1335 (537h)
Executable from the PLC	No
Function with ON/OFF state	No
Parameter 1	Selected User Defined Keyboards
Function available in Standard screens	Yes
Function available in keyboards	No
Function available in script	No
Function available in application entry script	No

11.3.32 Show all Users

Shows all the users defined in the project in a list.

Number of the Function	1291 (50Bh)
Executable from the PLC	No

Function with ON/OFF state	No
Function available in Standard screens	Yes
Function available in keyboards	No
Function available in script	No
Function available in application entry script	No

The following dialog will appear if this special function is activated:

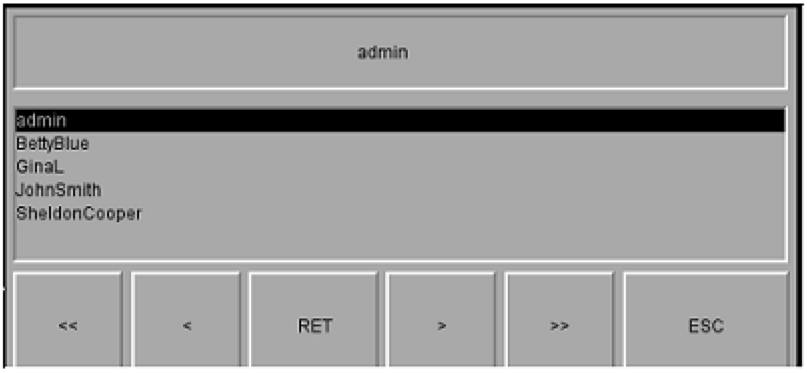


Figure 575: System keyboard appearing after the function key is tapped on

<< Used to move one page up within the selection list

< Moves inside the selection list one Item up.

> Moves inside the selection list one Item down.

>> Moves inside the selection list one page bottom.

ESC Exit the dialog again.

11.3.33 Unlock User

This function will unlock a locked user.

Number of the Function	1288 (508h)
Executable from the PLC	No
Function with ON/OFF state	No
Function available in Standard screens	Yes
Function available in keyboards	No
Function available in script	No
Function available in application entry script	No

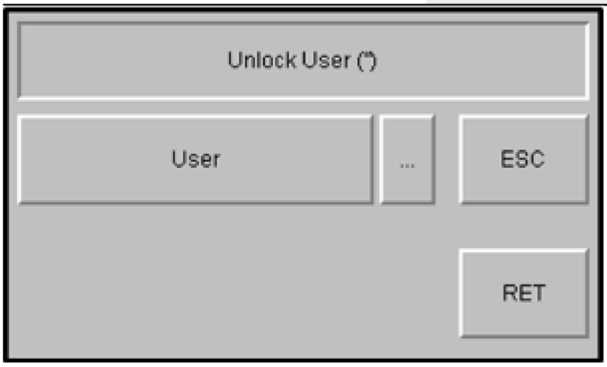


Figure 576: System keyboard appearing after the function key is tapped on

User User Used to enter the username with the user input keyboard configured for this purpose.

ESC Used to exit the dialog box without unlocking a user.

RET Unlocks the selected user.

The keyboard can be project-programmed in *Home\User Management\⚙ \General tab*→ section "Keyboardssection", page 899

11.3.34 Unlock User -userdef.

Opens a custom keyboard for unlocking a locked user.

Number of the Function	1320 (528h)
Executable from the PLC	No
Function with ON/OFF state	No
Parameter 1	Selected User Defined Keyboards
Function available in Standard screens	Yes
Function available in keyboards	No
Function available in script	No
Function available in application entry script	No

11.3.35 Unlock user x

Unlocks a locked user. The user is passed as parameter 1.

Number of the Function	1289 (509h)
Executable from the PLC	No
Function with ON/OFF state	No
Parameter 1	The user that should be unlocked.
Function available in Standard screens	Yes
Function available in keyboards	No
Function available in script	No
Function available in application entry script	No

11.3.36 UserLogin

Logging in with a user and the corresponding password.

After this function is called successfully, the language that the active user last used will be loaded. If the user has never logged in before, the language with ID <0> will be loaded.

→ After the login has been completed successfully, the content of the password tag will be cleared due to security reasons.

Parameter entered	User name

Parameter entered	User Password
Function available in script	Yes
Function available in application entry script	Yes

12. Conversions

Conversion of units

These values do not have to be displayed or entered in the same way as they are required by the PLC or stored in the PLC. Units and conversion factors can be defined.

The Value Conversion Manager can be used to convert tag values and assign a variety of units to them before they are shown by visualization objects. Moreover, it can be used to convert input values from visualization objects before these values are stored in tags.

A linear conversion factor with an offset can be specified. "MMI=PLC" will always be set by default. Tag conversions will always be performed in both directions between the PLC and MMI.

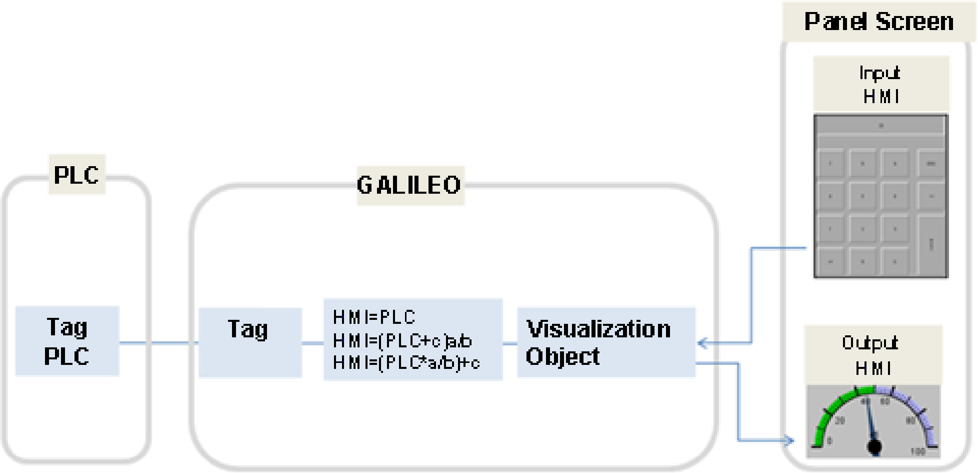


Figure 577: How value conversions work

The Conversions dialog box can be used to create multiple conversion groups, which can then be used by visualization objects. If a visualization object uses a value conversion for its input, the input value will be converted before it is stored in a tag. If the visualization object is instead used for output purposes, the tag value will be converted before the visualization object displays it.

Example

Say that the following settings have been configured for a conversion: MMI=PLC*a/b+c; a = 1; b = 2; c = 0. The tag has a default value of = 40. A value display/entry object uses this conversion and accordingly shows the tag with a value of <20>. If the keyboard is used to enter a value of <20> into the value display/entry object, the visualization object will show a value of <20>, but the tag will have a value of <40>.

The following visualization objects can use conversions:

- Value Display/Entry
- Bargraph
- Gauge
- Parameter list

The conversion group you want can be selected in the properties for the corresponding visualization object. If there are multiple conversions defined for that group, only the one at the top of the list will be performed.

The Conversion of Group and Next Conversion of Group functions can be used to move from one conversion to the next within a conversion group. All the tags belonging to the group will be converted again accordingly and displayed.

Multiple tags can belong to the same conversion group. Every single tag that belongs to the conversion group will be converted again accordingly and displayed.

An example: There can be a conversion group called <Temperature>. All tags with °C or °F temperatures would belong to this group. In order for values to be converted to <Celsius> in °C and <Fahrenheit> in °F, the following settings would be required:

Conversion	Catalog Number	a	b	c	Decimals	Unit
Celcius	MMI = PLC	1	1	0		°C
Fahrenheit	MMI = PLC* a / b + c	9	5	32		°F

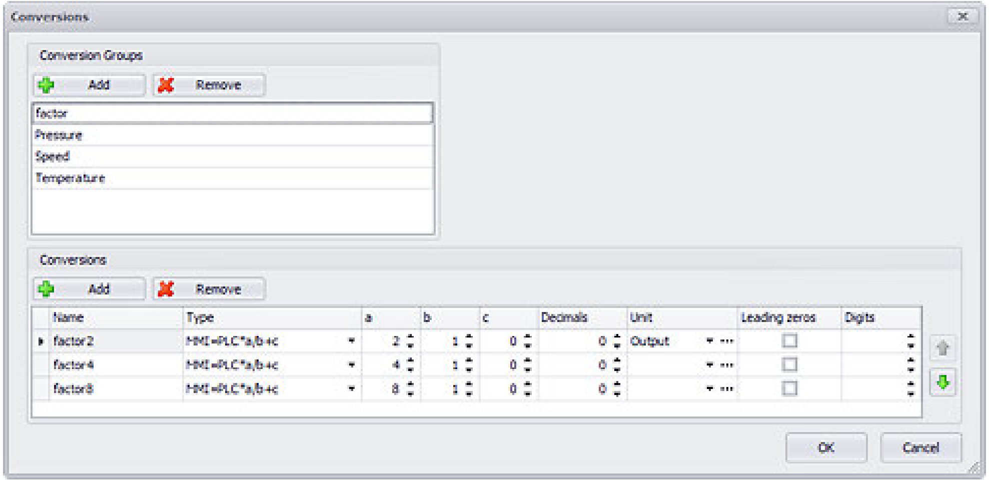
Another example would be a conversion group that converts between millimeters and inches or between kilometers and miles. Yet another one would be a conversion group called <US - Europe> to which all tags with millimeters and inches, kilometers and miles, °C and °F would be assigned, making it possible to switch between all conversions with one single function.

As a general rule, it is a good idea to assign all the tags that you want to convert at the same time to the same conversion group and implement aConversion of Group function and a Next Conversion of Group function in the project.

12.1 Value Conversion

The conversion tag object is used to manage the conversions of values into various units for the project.

Home\Project Configuration\ Conversions tab



factor
Pressure
Speed
Temperature

	Name	Type	a	b	c	Decimals	Unit	Leading zeros	Digits
▸	factor2	MMI=PLC*a/b+c	2	1	0	0	Output	☐	
	factor4	MMI=PLC*a/b+c	4	1	0	0		☐	
	factor8	MMI=PLC*a/b+c	8	1	0	0		☐	

Figure 578: Conversions dialog box with a Temperature conversion group

Conversion Groups

You can create a maximum of 65,000 conversion groups. The conversion group you want can be selected in the properties for the corresponding visualization object. Each conversion group can contain multiple conversions. Normally, several visualization objects use a single conversion group. All the tags in these visualization objects will be converted before being input or output using the conversion at the top of the list for the conversion group, and the values will be shown with the units defined for the list. If ⇧ or ⇩ is used to move a different conversion to the top of the list, this new conversion will be the one applies to all output and input using the corresponding conversion group. This means that you do not have to change the conversion for all visualization objects individually, and can instead make the change for all of them by making it in the conversion group.

Conversions

A maximum of 65,000 conversions can be defined for each conversion group. However, only the conversion at the top of the list will be applied..

Name

Name of conversion within the conversion group.

Type

Between the following three conversion modes can be selected:

MMI = PLC	In this conversion mode the values between the PLC and the device are not converted. The Fields a, b, c are not accessible and have no meaning.
MMI = PLC* a / b + c	In this conversion mode the values between the PLC and the device are converted

	using the equation specified. Both MMI values (Min. and Max. fields) are calculated according to the defined equation and shown according to the specified settings in → "Decimals", page 924 and → "Digits", page 925.
MMI = (PLC + c) * a / b	In this conversion mode the values between the PLC and the device are converted using the equation specified. Both MMI values (Min. and Max. fields) are calculated according to the defined equation and shown according to the specified settings in → "Decimals", page 924 and → "Digits", page 925.

Fields a, b, c

These fields contain the constants for the conversion between the device value and the PLC value.

Field	Description	Number range
a	Multiplier	Only positive numbers
b	Divisor	Only positive numbers greater than 0
c	Offset	Positive and negative numbers

If you select the MMI = PLC conversion type, these values will not be relevant.

Decimals

Used to specify the number of decimal places that should be shown for the value. The number of decimal places cannot be changed at runtime without changing the conversion with the Conversion of Group or Next Conversion of Group function.

The following applies to tags of data type byte, word, double word and quad word:

Since data the BYTE, WORD, DWORD and QWORD data types are integer data types, the decimal point in the value will be placed the number of decimal places to the left. This means that the value will be reduced by the corresponding power of 10.

Example:

Value	25636
Decimals	2
Display	256.36

The following applies to tags of data type FLOAT:

The value will be displayed with the configured number of decimal places.

Example:

Value	123,254
Decimals	2
Display	123.25

Unit

This cell can be used to enter a unit or select an existing one by clicking on the drop-down menu arrow. Units entered will be interpreted as standard texts, meaning that clicking on ... will open the Text Editor so that a standard text can be selected and/or edited. The conventions that apply to standard texts apply here as well.

In order for the unit to be shown with a visualization object, it must be enabled in the properties for that object.

Leading zeros

12. Conversions

12.1 Value Conversion

If this property is enabled, it will only have an effect if a number is entered into the Digits cell. If the value has fewer digits than the number defined in Digits, the remaining whole number places will be filled with zeros. If the value has more digits than the number defined in Digits, the field will be ignored.

Digits

This setting can only be used with tags of data type BYTE, WORD, DWORD, QWORD and FLOAT.

Used to specify how many digits should be used when displaying the value. This value must not be smaller than the number of digits for the minimum or maximum MMI value (Min. and Max. display fields). In addition, the number of digits must be at least one digit more than the number of decimal places (not including the decimal mark).

The number of digits cannot be changed at runtime without changing the conversion with the Conversion of Group or Next Conversion of Group function.

If the value has fewer digits than the number defined in Digits, the remaining whole number places will be filled with zeros.

12.1.1 Conversion of Group

Calling the Conversion of Group function will switch conversions within the conversion group specified with the 1st parameter. The switch will be from the current conversion (the one at the top of the Conversions list by default) to the conversion specified with the 2nd parameter.

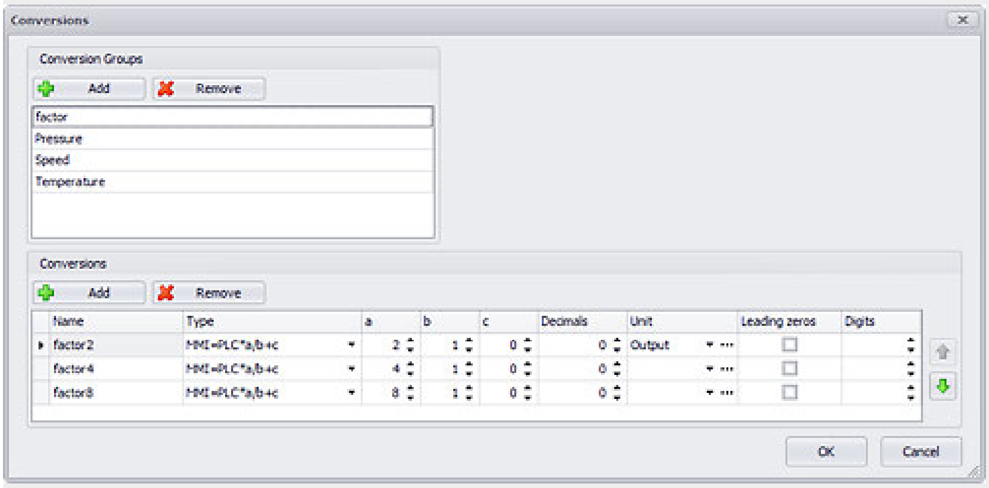
This setting will be retained even after the device is switched off.

Number of the Function	1792 (700h)
Executable from the PLC	No
Function with ON/OFF state	Yes
Parameter 1	Conversion Group
Parameter 2	Conversion within the conversion group that should be activated
Function available in standard screens	Yes
Function available with keyboards	No
Function available in script	Yes
Function available in application entry script	Yes

Example of a conversion with the Conversion of Group function

Say that all visualization objects (e.g., value display/entry objects) that are linked to the <factor> conversion group convert the value of the tag they are displaying with the conversion at the very top (<factor2) by default.> Now say that when the device operator clicks on a button for the Conversion of Group function (from the Conversion group), the conversion used should change to <factor8>.

Home\Conversions tab



Conversions

Conversion Groups

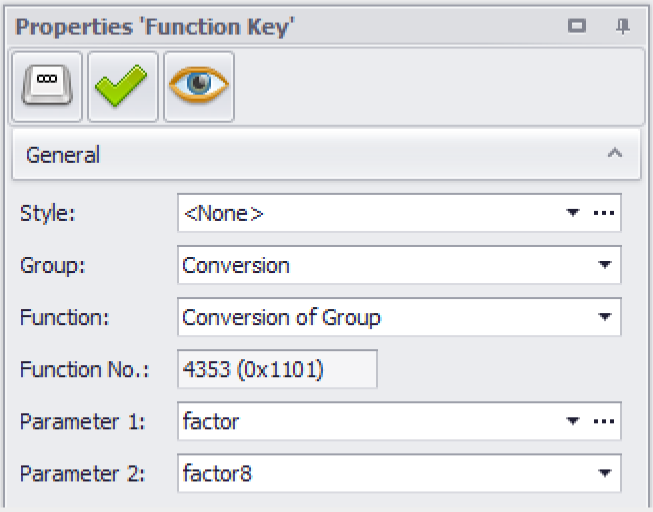
factor
Pressure
Speed
Temperature

Conversions

	Name	Type	a	b	c	Decimals	Unit	Leading zeros	Digits
▸	factor2	HMI=PLC*a/b+c	2	1	0	0	Output	□	
	factor4	HMI=PLC*a/b+c	4	1	0	0		□	
	factor8	HMI=PLC*a/b+c	8	1	0	0		□	

Figure 579: Conversions dialog box

The properties for the function key will look as follows:



Properties 'Function Key'

General

Style: <None>

Group: Conversion

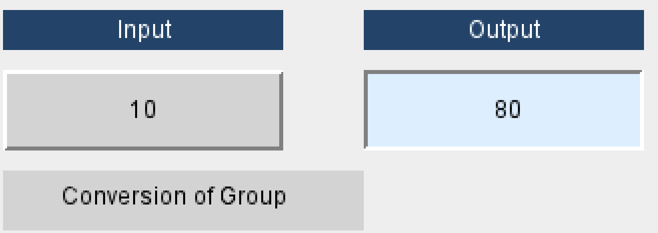
Function: Conversion of Group

Function No.: 4353 (0x1101)

Parameter 1: factor

Parameter 2: factor8

Device screen



12.1.2 Next Conversion of Group

Calling the Next Conversion of Group function causes the conversion to switch to the next conversion within the specified conversion group. If there are no additional conversions after the one currently selected within the conversion group, the conversion will be changed to the topmost conversion in the list. If the conversion group only has one conversion, this function will not do anything.

This setting will be retained even after the device is switched off.

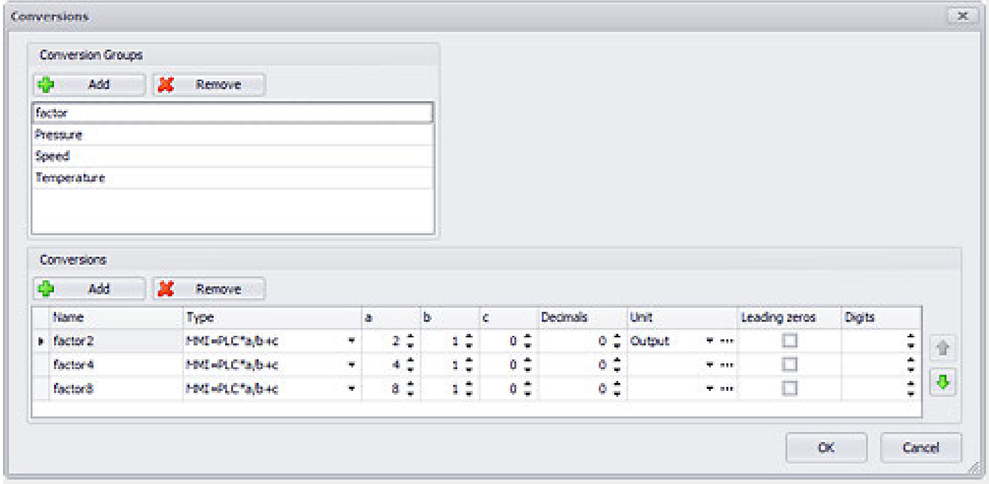
Number of the Function	1792 (700h)
Executable from the PLC	No
Function with ON/OFF state	Yes
Parameter 1	Conversion Group
Parameter 2	Conversion group conversion
Function available in standard screens	Yes
Function available with keyboards	No
Function available in script	Yes
Function available in application entry script	Yes

Example Conversion

In order to make the following example easy to test on a computer, we will create a tag called <inputUser>. The device operator (i.e., you) will be able to change the value on the <Input> value display/entry object at runtime. A value display/entry object called <Output> will output the tag's value after it has been converted using the <factor> conversion group.

The <factor> conversion group in this example has three conversions. Conversion <factor2> multiplies the tag by two, conversion <factor4> multiplies it by four, and conversion <factor8> multiplies it by 8. Accordingly, the conversion group looks as shown in the screenshot below:

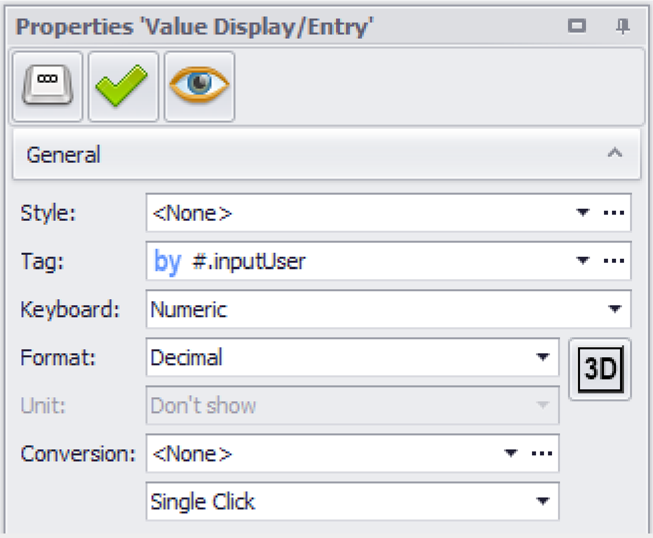
Home\Conversions tab



factor
Pressure
Speed
Temperature

Name	Type	a	b	c	Decimals	Unit	Leading zeros	Digits
factor2	HMI=PLC*a/b+c	2	1	0	0	Output	□	
factor4	HMI=PLC*a/b+c	4	1	0	0		□	
factor8	HMI=PLC*a/b+c	8	1	0	0		□	

The <Input> value display/entry object is linked to a numeric keyboard. The properties for the <Input> value display/entry object will look as follows:



Properties 'Value Display/Entry'

General

Style: <None>

Tag: by #.inputUser

Keyboard: Numeric

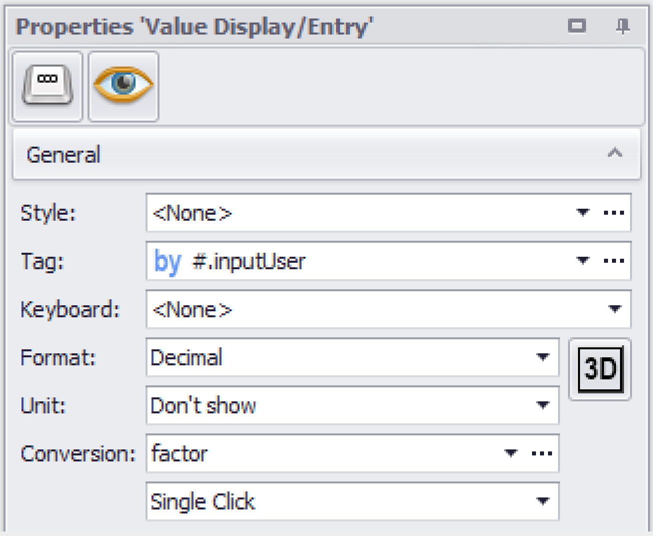
Format: Decimal

Unit: Don't show

Conversion: <None>

Single Click

The value shown in the <Output> object should be converted using the <factor> conversion group before it is displayed. The properties for the <Output> value display/entry object will look as follows:



Properties 'Value Display/Entry'

General

Style: <None>

Tag: by #.inputUser

Keyboard: <None>

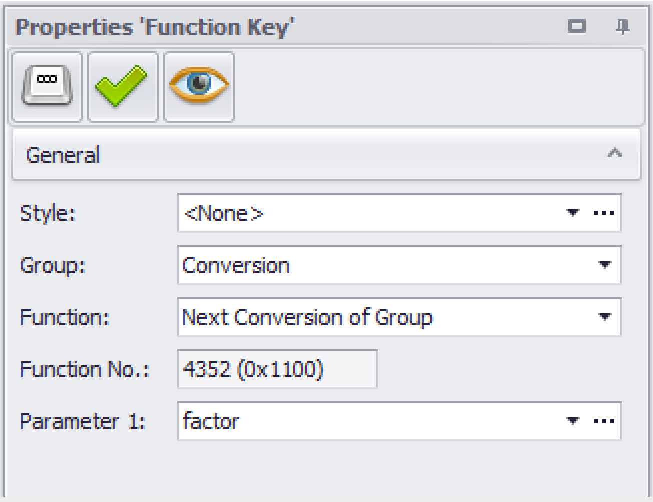
Format: Decimal

Unit: Don't show

Conversion: factor

Single Click

The properties for the function key will look as follows:



Properties 'Function Key'

General

Style: <None>

Group: Conversion

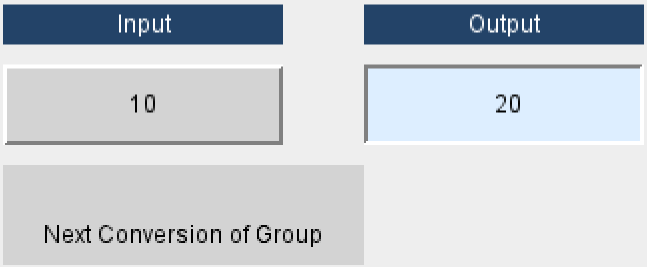
Function: Next Conversion of Group

Function No.: 4352 (0x1100)

Parameter 1: factor

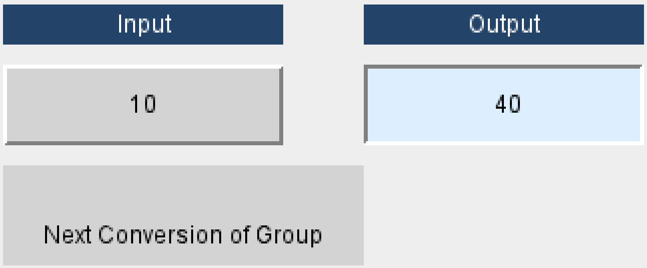
The example can then be compiled and started with the Inspector. The result will be that when the device operator (i.e., you) clicks on the button under <Input> after the application starts, a numeric keyboard will appear.

If you enter a value of <10>, the <Output> value display/entry object will show a value of <20>. At this point, the <factor2> conversion in the <factor> conversion group is being applied.



Input	Output
10	20

If you now click on the button for the Next Conversion of Group function (from the Conversion function group), the <factor4> conversion will be applied instead, and the result on the device screen will be as follows:



Input	Output
10	40

13. Help management system

for providing users with assistance

GALILEO users can use the visualization software program to provide device operators with quickly accessible help in order to provide additional information in specific situations.

Help-specific screens, visualization objects, and texts are used when setting up a help management system. More specifically, this type of system is based on help screens, help buttons, and help texts.

There are two help strategies that can be used with a help management system:

1. Providing a help button that device operators can use at runtime. When a device operator taps on the button, a help text will appear please refer to → section "Providing device users with help", page 932.
2. Providing a list with tags that can be selected by device operators at runtime. In addition, this list needs to be combined with a button with the Show Tag Help function. When a device operator taps on the button, a help text for the selected tag will appear please refer to → section "Showing the help for tags in a list", page 796.

The help screen that will appear is set in the properties for the corresponding help button or tag list. The corresponding help text, which is also defined using the properties for the help button or tag list, will be passed to the help screen.

In turn, the properties for the help screen can be used to define what the help texts should look like when displayed. In fact, it is also possible to show an image instead of a help text, or to show both. A total of three display areas can be defined on a help screen: two for help texts and a third for a complementary image.

The help texts themselves are managed in the Text Editor dialog box. For detailed information on the Text Editor dialog box and how to manage help texts, please refer to → section "Working with the Text Editor ", page 640.

The following sections go over the properties in help buttons and help screens. In addition, an example is used to illustrate how these two interact. Please note that the same rules that apply to standard texts apply to help texts. Moreover, all texts are language-dependent, Unicode-compatible, and support characters for all languages.

13. Help management system

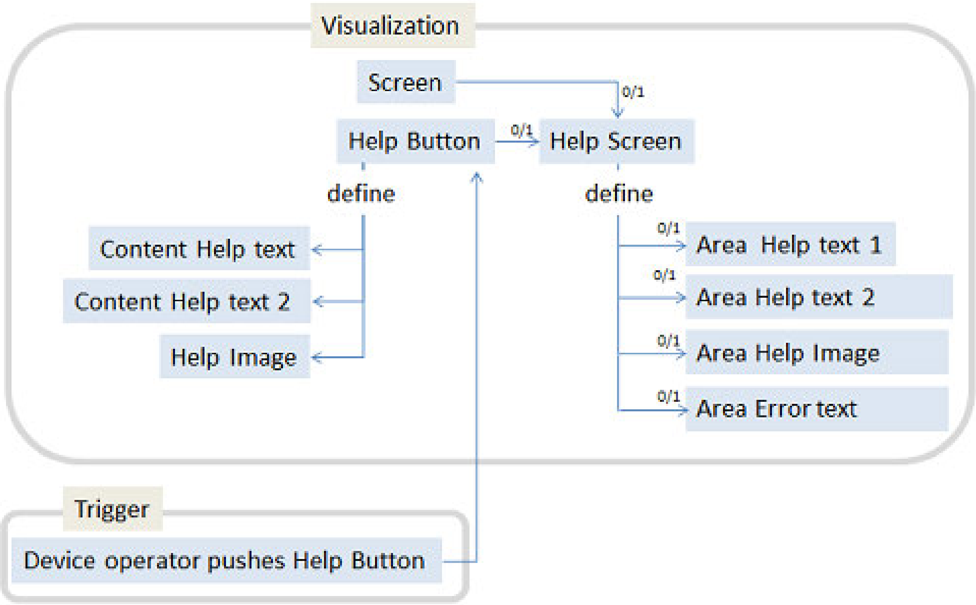


Figure 580: Help management system with a help button

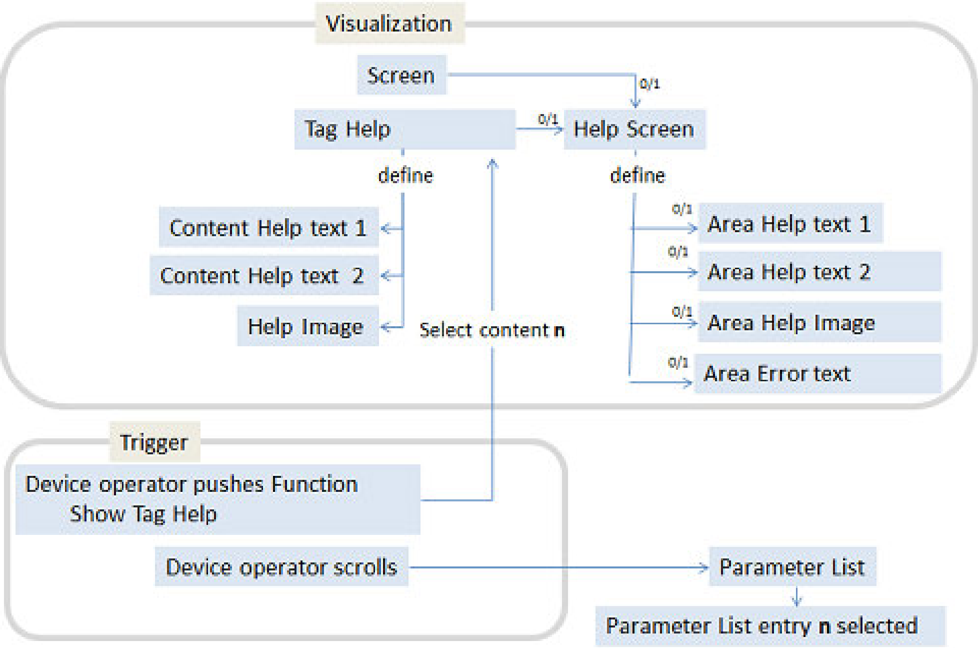


Figure 581: Help management system with tag help

13.1 Providing device users with help

In real-life applications, it is usually advisable to provide device operators with help information for when errors occur, there are multiple options available for selection on the screen, etc.
This functionality can also be used when operators need to enter values that are not self-explanatory, for example.

The following help strategies can be used within this context:

1. Providing a help button that device operators can use at runtime. When a device operator taps the button, a help text will appear – please refer to → "Creating help buttons and assigning help information", page 932
2. Providing a list with tags that can be selected by device operators at runtime. In addition, this list needs to be combined with a button with the Show Tag Help function. When a device operator taps on the button, a help text for the selected tag will appear. For detailed information on how to implement this type of help, please refer to the sections covering the parameter list object – → "Showing the help for tags in a list", page 796 .
3. Displaying an error message for the device operator as soon as an error event occurs. The Error help function can be used to provide the operator with additional troubleshooting information. For detailed information on how to implement this type of help, please refer to the sections covering the alarm list object – → section "Providing help information in the event of an error", page 854 .
4. If the alarm list is too small for showing the entire error text, the Error line extend function from the Internals group can be used to open an additional help window that can display the entire error text – please refer to → section "Error line extend", page 885.

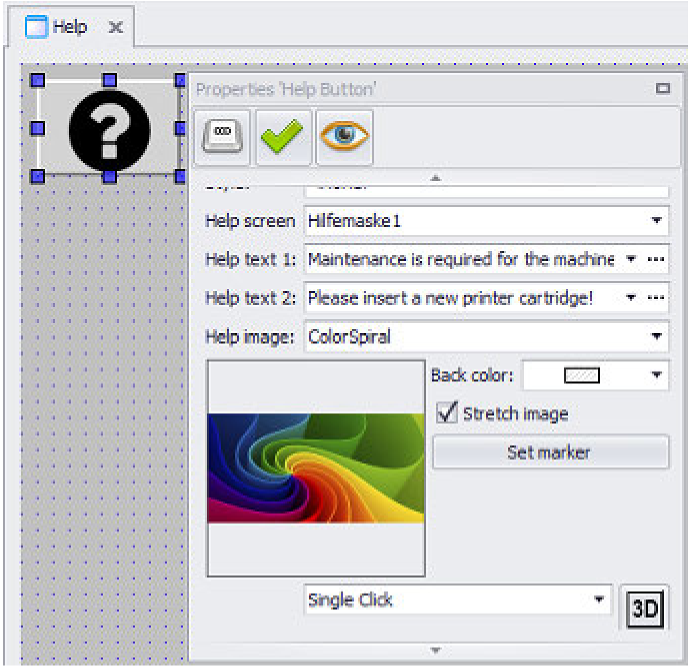
Following are a series of examples that show how to use these strategies.

Creating help buttons and assigning help information

Follow the steps below to provide operators with a help text when they tap on a button:

- ▶ Create a new screen in *Project Tree\Screens\Standard screens*, e.g., <Help>.
- ▶ In the *Controls* tab, click on Help Button and draw a help button visualization object on the screen.
- ▶ Double-click on the button to access its Extended properties dialog box and assign an image to it.
- ▶ In the States tab, select the Icon as foreground option.
- ▶ Click on the drop-down menu arrow and select an image from the Project images folder, e.g., <help(0)>.
- ▶ Click on OK.

- Make sure that the help button is still selected on the screen.
- In the Properties 'Help Button' pane, enter the texts you want into the Help text 1 and Help text 2 fields, i.e., the texts that should be shown on the device screen when the button is tapped on.
- Click on the drop-down menu arrow for the Help image field and select an image from the Project images folder. This image will also be shown on the device screen when the button is tapped on.
- Enable a Stretch option.

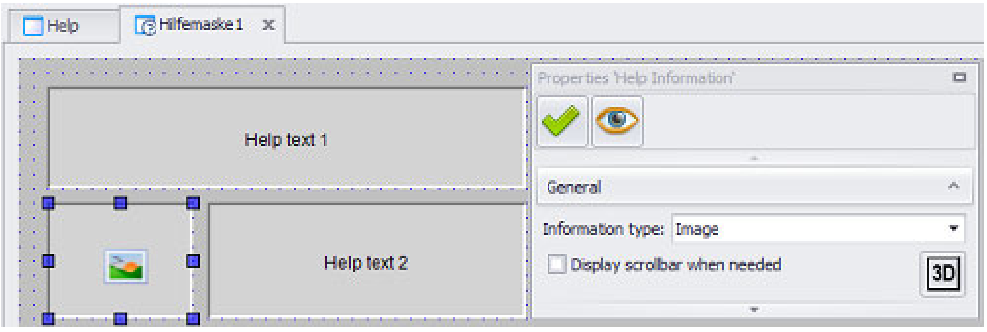


Help screen: Hilfemaske1
Help text 1: Maintenance is required for the machine
Help text 2: Please insert a new printer cartridge!
Help image: ColorSpiral
Back color:
☑ Stretch image
Set marker
Single Click
3D

Figure 582: Help screen with help button; Properties 'Help Button' pane on the right

The Help screen of current screen< option will be selected by default in the >Help screen field.

- Right-click on *Project Tree\Screens\Help Screens* and use the context menu to add a new help screen, e.g., <HelpScreen_1>.
- Click on Controls\Help Information and draw a help information visualization object on the help screen. Repeat this step twice so that there are a total of three help information objects on the screen.
- Click on the first help information visualization object. Now click on the drop-down menu arrow for the Information type field in the Properties 'Help Information' and select the Help text 1 option. Do the same for the other two objects, except selecting Help text 2 for one and Image for the other.



Properties 'Help Information'

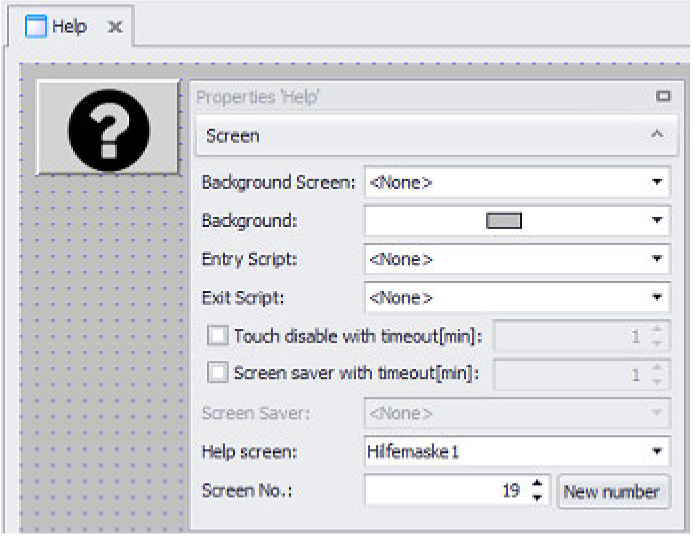
General

Information type: Image

□ Display scrollbar when needed

Figure 583: HelpScreen_1 help screen with three help information visualization objects; Properties 'Help Information' pane on the right

- ▶ Go to *Project Tree\Screens\Standard screens*, select the screen with the help button, and open the properties for the screen using the context menu.
- ▶ In the Help screen field, select the help screen you just created, e.g., <HelpScreen_1>.



Properties 'Help'

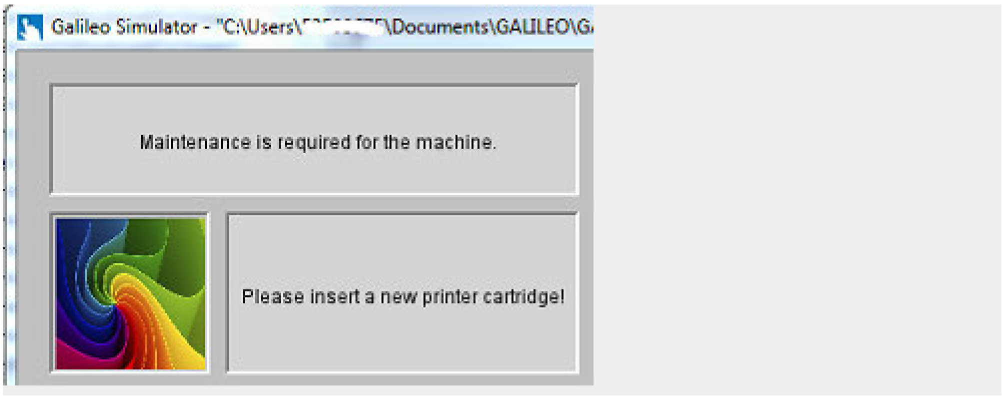
Screen

Background Screen:	<None>
Background:	
Entry Script:	<None>
Exit Script:	<None>
□ Touch disable with timeout[min]:	1
□ Screen saver with timeout[min]:	1
Screen Saver:	<None>
Help screen:	Hilfemaske1
Screen No.:	19 New number

Figure 584: Help screen with help button; Properties 'Help' pane on the right

- ▶ Start the Inspector by pressing the F5 key.

Example on the Device screen



Galileo Simulator - "C:\Users\[redacted]\Documents\GALILEO\G

Maintenance is required for the machine.

Please insert a new printer cartridge!

13.2 Tag Help

The tag help object is used when you want to show help texts for tags in a list at runtime. As of this writing, only the parameter list object can be used for this purpose. Please note that the tag help object is not a visualization object, but rather an object that is used to link help information to tags.

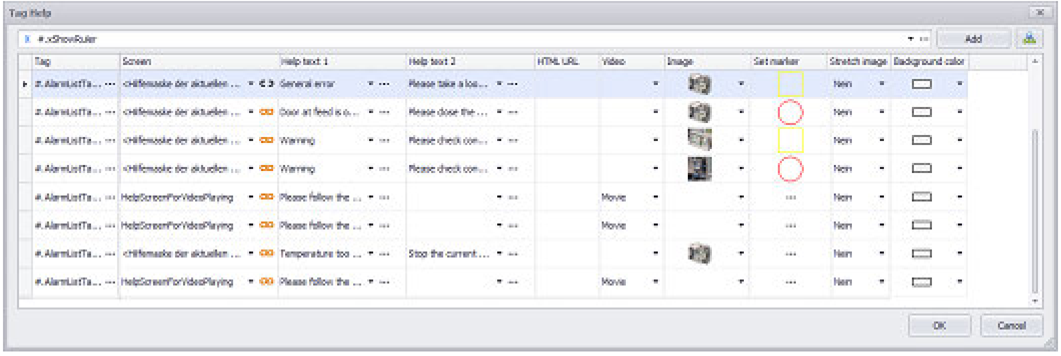
The tag help object can be used to link each tag to two help texts and one help image. Likewise, the tag needs to be linked to a help screen that will show the help texts and help image as soon as the Show Tag Help function from the ParameterList group is run. This function will show the help texts for the tag that is currently selected in the list when the function is called. In order to trigger the function, there must be a function key displayed as a button on the device screen so that device operator can tap on the button to run the function.

The help screen must first be created like any other screen.

Prerequisites

In order to be able to follow the descriptions below, you must first have clicked on a help button on the current screen.

Home\Project Configuration\ Tag Help



Tag Help

#.xShowRuler

Tag	Screen	Help text 1	Help text 2	HTML URL	Video	Image	Set marker	Stretch image	Background color
#.AlarmListTa...	<Hilfemaske der aktuellen ...	General error	Please take a loo...				□	Nein	□
#.AlarmListTa...	<Hilfemaske der aktuellen ...	Door at feed is o...	Please close the ...				○	Nein	□
#.AlarmListTa...	<Hilfemaske der aktuellen ...	Warning	Please check con...				□	Nein	□
#.AlarmListTa...	<Hilfemaske der aktuellen ...	Warning	Please check con...				○	Nein	□
#.AlarmListTa...	HelpScreenForVideoPlaying	Please follow the ...			Movie		...	Nein	□
#.AlarmListTa...	HelpScreenForVideoPlaying	Please follow the ...			Movie		...	Nein	□
#.AlarmListTa...	<Hilfemaske der aktuellen ...	Temperature too ...	Stop the current ...				...	Nein	□
#.AlarmListTa...	HelpScreenForVideoPlaying	Please follow the ...			Movie		...	Nein	□

Figure 585: Tag Help dialog box

Tag

A tag can be entered for each entry in the Tag Help dialog box. Clicking on the drop-down menu arrow will show all the tags in the project so that you can select the one you want. If no tags are shown, this means that no tags have been created in or imported into the project. For a description of how to work with tags, please refer to the → section "Working with tags", page 539 section.

Screen

Used to specify the help screen that will be used to show the help text 1, help text 2, and image contents. For a description of the various configuration options for help screens, please refer to the → section " Help screens", page 939 section

There is a global setting for defining whether a visualization is linked or unlinked. In order for the and chain link icons to be displayed, there must be

more than one visualization in a group.
For more information, please refer to: → section " Web visualization", page 1070

Help text 1 / Help text 2

Used to enter the help texts that should be shown for the selected tag when the function key with the Show Tag Help function is tapped on.

→ If the help text does not have enough space in the help information visualization object at runtime, a scrollbar will automatically appear on the right side of the help text field if the Display scrollbar when needed option is enabled for the help information visualization object.
If a line is too wide for the help information visualization object, the line will automatically wrap on the device at the last space in the line if the Word Wrap option is enabled for the help information visualization object. Please refer to → section " Help information", page 433

...

If the help text you want does not exist yet, clicking on this button will open the Text Editor, which can be used to define and select help texts as necessary. For detailed information on how to work with the Text Editor, please refer to → section "Working with the Text Editor ", page 640

Image

The selected image is associated with the relevant Tag.

Selected images can be removed by clicking on the drop-down menu and selecting <None> or by selecting the field and pressing the Delete key.

Clicking on the drop-down menu arrow in the Image field will open a drop-down menu with all the images found in the Project images folder. This means that, before selecting an image here, you will need to add it to Project Tree\Images\Project images by importing it or by dragging & dropping it there. Images will be shown in the preview window.

Copying an image to the Media Project images folder

To copy an image from Windows Explorer to Project Tree\Media tab\Project images, follow the steps below:

- ▶ Open Windows Explorer and locate the image you want
- ▶ In the project tree, open the Media\Project images folder.
- ▶ In Windows Explorer, click on the image you want and drag it specifically to the Project images folder.

After you release the left mouse button, the image will be added to the Project images folder.

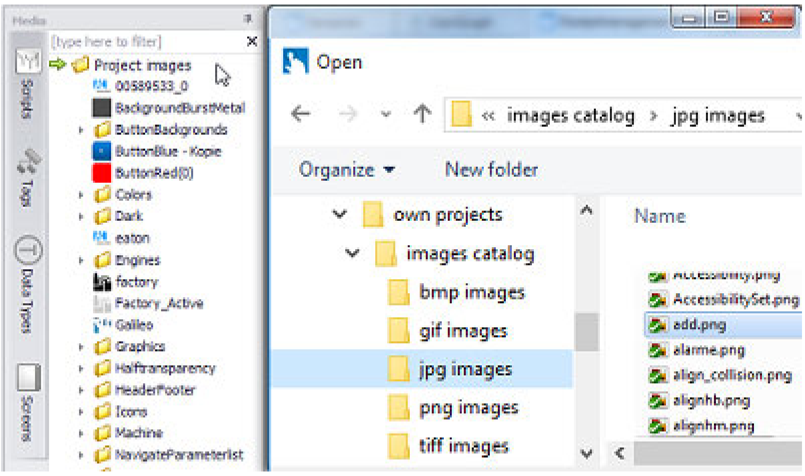
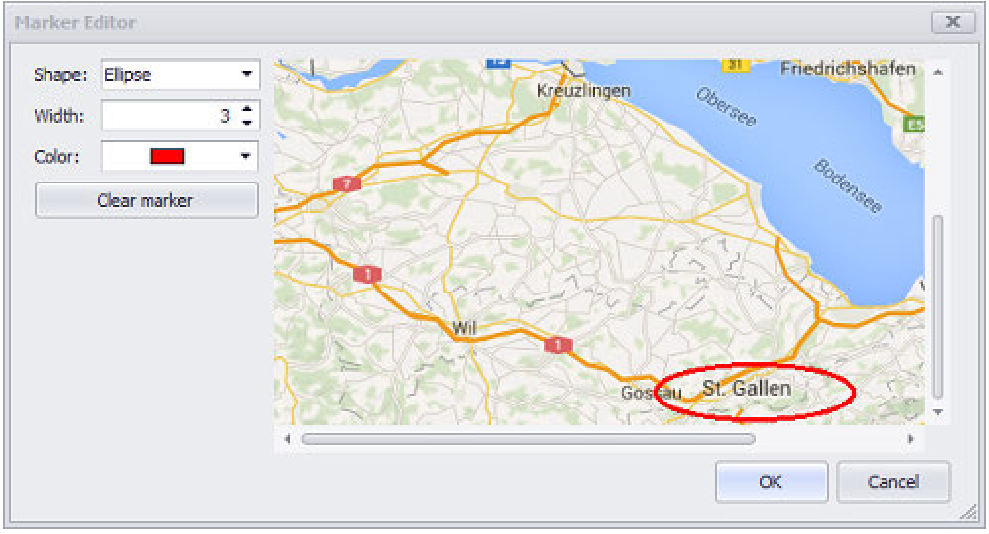


Figure 586: The add.png image is copied from Windows Explorer to the Project images folder by dragging and dropping it

Set marker

This option will only be enabled if an image is selected in the Image field. Clicking inside the field will open the Marker Editor dialog box.

Home\ Tag Help\Set marker


Marker Editor

Shape: Ellipse
Width: 3
Color: 🟥

Figure 587: Windows

Shapes

The "Ellipse" and "Rectangle" shapes are available in the drop-down menu for the type of marker.

Width

The line width for the shape is defined here, in pixels.

Color

This field can be used to select a color for the marker over the button.

You can change the color if you want. The following options are available for this purpose:

- Transparent
- Standard Colors
- More Colors...

The most recently used colors will be shown (user-specific).

For detailed information on the available colors, please refer to the following section:
→ section "Color choice - Object's Color Palette", page 234

Clear marker

Clicking on this button will remove the marker from the image.

Stretch image

If the checkbox is enabled, the following will happen:

- If it is smaller than the selected display area, the image will be stretched to fit the visualization object Help Information.
- If it is larger than the selected display area, the image will be shrunk to fit the visualization object Help Information.

If the checkbox is disabled, only a section of the image will be shown in the help information visualization object if the image is larger. If it is smaller, it will be shown centered in the visualization object. In the latter case, the display area will be filled out with the color that was selected in the Background property.

Background color

This cell can be used to set a background color. This color will be visible if the image is not stretched and does not fill the entire display area.

You can change the color if you want. The following options are available for this purpose:

- Transparent
- Standard Colors
- More Colors...

The most recently used colors will be shown (user-specific).

For detailed information on the available colors, please refer to the following section:
→ section "Color choice - Object's Color Palette", page 234

13.3 Help screens

Help screens provide assistance to device operators in specific situations by providing additional information. In contrast to standard screens, help screens can contain help information visualization objects. In addition, all static visualization objects can be used in help screens, as can containers.

If a device operator taps on a help button at runtime, the corresponding help screen will be shown on the device screen. The help screen that will appear is set in the properties for the help button. Meanwhile, the corresponding help text, which is also defined using the properties for the help button, will be passed to the help screen.

In turn, the properties for the help screen can be used to define what the help texts should look like when displayed. In fact, it is also possible to show an image instead of a help text, or to show both. A total of three display areas can be defined on a help screen: two for help texts and a third for a complementary image.

Prerequisites

In order to be able to follow the descriptions below, you must first have clicked on a help screen in *Project Tree\Screens\Help Screens*.

Project Tree\Screens\Help Screens

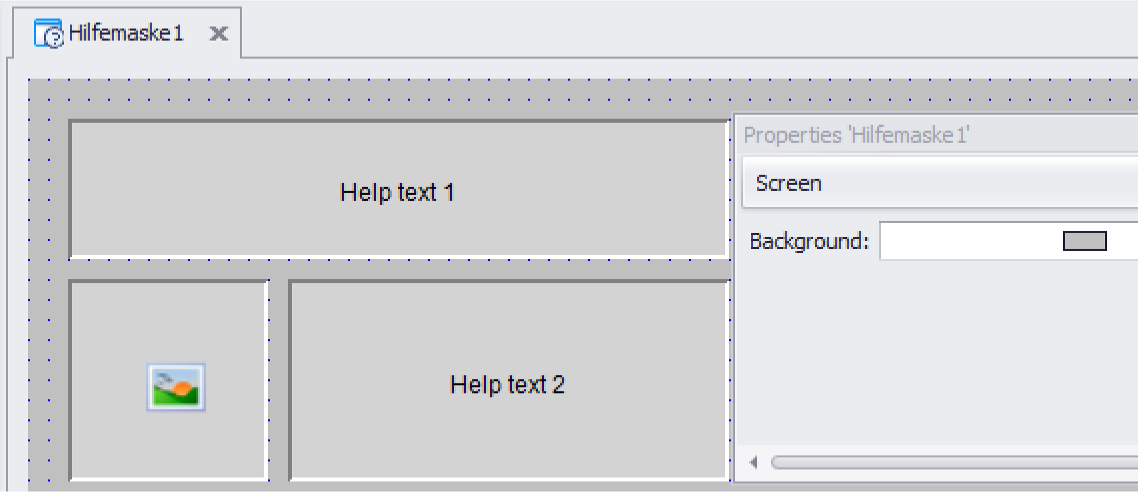


Figure 588: Properties pane for the HelpScreen_1 help screen

Background color

You can change the color if you want. The following options are available for this purpose:

- Transparent
- Standard Colors
- More Colors...

The most recently used colors will be shown (user-specific).

For detailed information on the available colors, please refer to the following section:
→ section "Color choice - Object's Color Palette", page 234

Transparent

The transparent background color is only available for user controls, help screens, and keyboards. The <transparent> background color is not available for standard screens.

Activating this check box provides a screen with a transparent background so that a color setting cannot be made for it. On the device a screen change to a screen with a transparent background gives the appearance of overlaying the new screen on the previous one.

→ The zones in the previous screen will remain visible, but will no longer be functional.
Only the objects in the current screen will be usable and updated.

13.4 Help management functions

13.4.1 Show Tag Help

Used to display a help screen for the parameter list entry that is currently selected. The entry's index will be automatically passed to the object. The help screen shown will be the one assigned to the tag for the currently selected entry in the tag help object.

The help text that will appear for the selected parameter list entry is stored in the tag help object. Meanwhile, the help screen area in which the help texts will be displayed is defined in the corresponding help screen.

The function will not do anything if a help screen has not been assigned to the tag in the tag help object or if no areas have been defined in the help screen. Since a screen can contain multiple parameter lists, parameter 1 needs to be used to select the one you want.

Number of the Function	3590 (E06h)
Executable from the PLC	No
Function with ON/OFF state	No
Parameter 1	Used to select the parameter list for which a help text should be displayed.
Function available in standard screens	Yes
Function available with keyboards	No
Function available in script	Yes
Function available in application entry script	No

For an example showing how to use this special function, please refer to the following section: → section "Providing device users with help", page 932.

14. User Logger

The user logger can be used to log user input on the device. When a value change is logged, a timestamp and the user management user that is currently logged in will be logged with it. The user logger will log all value changes (e.g., made with the value display/entry visualization object), as well as instances of certain functions being run.

→ Value changes made by the PLC will not be logged.

Value changes and actions will be written to a buffer. The contents of this buffer will then be written to the log file at regular intervals.

The log file will be saved as a .csv file.

Where is the log file saved?

The log file will be saved in the folder entered as an absolute path in the *Directory* field in Home\Settings\User Logger tab, e.g., <`C:\Temp\log`>.

If a relative path is entered, it will be created as a subfolder in the project path and the file will be saved there. The project directory is defined in the *Home\Settings\Runtime tab\General data directory* field.

If the buffer reaches a definable maximum number of entries, its contents will be written from the data storage unit to a log file.

> *NOTICE*
>
> All log data in the buffer will be lost if you disconnect the touch panelpanel with the enabled user logger from the power supply

→ The user logger will automatically start again when the device is started again.

Applications

- Quality assurance
- Statistical data analysis
- Troubleshooting
- Logged production

Basic setup procedure

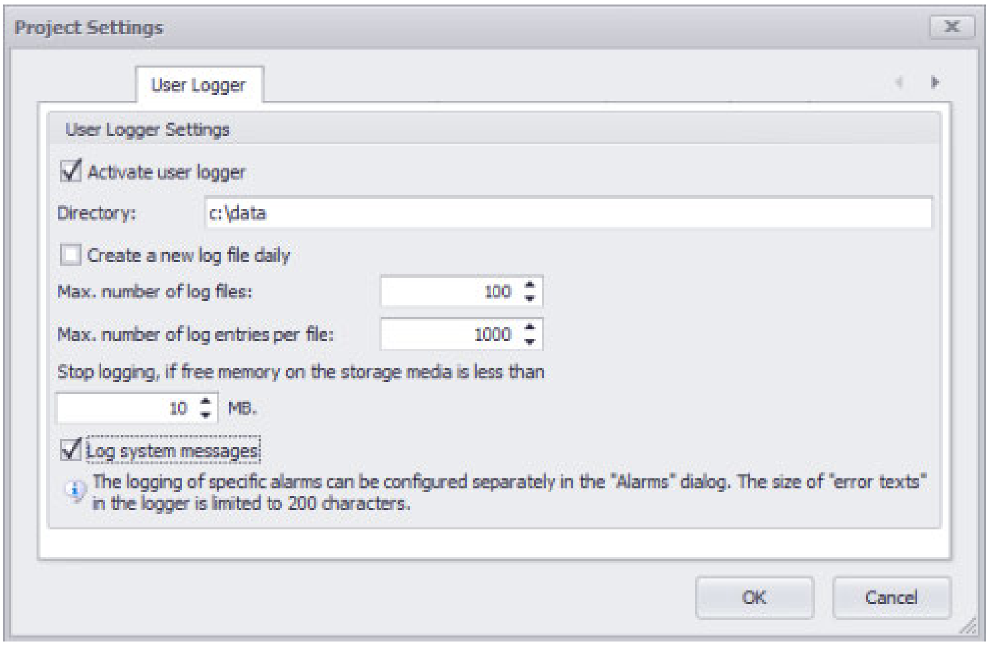
1. Enable the logger
2. Configure the logger settings
3. Create an ignore list
4. Select log functions

For a description of the functions that are supported, please refer to the → section "List of functions logged", page 947 section.

Enable the user logger in the project's settings

14. User Logger

Home\ Settings tab



Project Settings

User Logger

User Logger Settings

☑ Activate user logger

Directory: c:\data

☐ Create a new log file daily

Max. number of log files: 100

Max. number of log entries per file: 1000

Stop logging, if free memory on the storage media is less than

10 MB.

☑ Log system messages

The logging of specific alarms can be configured separately in the "Alarms" dialog. The size of "error texts" in the logger is limited to 200 characters.

Figure 589: Project settings window\User logger tab

The file will be in UTF16-LE format, i.e., a format with a type of encoding with variable length for Unicode characters. The individual columns will be tab-separated, and the file will have the .csv extension.

The filename will be determined by the runtime system based on the following rules:

If a new log file is generated, the filename will be made up of the project name, the current date, and a three-digit index. This index starts at 000 and ends at 999. The next available number for the current date will be used for the index,

☐ The functions will be disabled

☑ Activate user logger

If you enable the user logger, you will have to enter a storage location.
Activates the user logger for the current project. This means that logging will being after startup.
You can use the Logger_ON-OFF function to start and stop the user logger at runtime.
Please note that starting the user logger at runtime will only be possible if the user logger has been activated previously by enabling the corresponding checkbox while setting up the project.

Directory

Used to enter the storage location where the log files will be stored.
This directory must not be located on the same data storage device as the operating system on which GRS is running.

→ Configuring the directory incorrectly will result in a system error message at startup and will prevent the user logger from being used,

☑ Create a new log file daily

Prevents log entries from being overwritten
This option ensures that a log file will not contain entries belonging to different days. When the date changes and a new log entry is generated, a new log file will be automatically generated even if the maximum number of log entries per file has not been reached yet.

Max. number of log entries per file
The maximum number of entries per log file. If the maximum number of log entries is reached, a new log file will be automatically generated and the filename index will be incremented by 1.
If the date has changed since the last time a log file was generated, the index will start again from 0.

Max. number of log files
The maximum number of log files that will be generated.
If this number is reached, the oldest log file will be deleted before a new one is generated.

Stop logging, if free memory on the storage media is less than
The minimum amount of free space that there must be on the data storage device. The defined amount of memory must be available in order for a log entry to be generated. If the amount of free space available is less than the value specified here, a system error message will be shown every time an attempt is made to write to the log file. In order for the error message to stop appearing, enough free space must be made available or the user logger must be disabled.

☑ Log system messages
If this checkbox is enabled, GALILEO will also log system messages in the log file.

Log file format

Every log file will have a header with the following information at the beginning:

```
Log File :          C:\Log\CONTROLSTATUSDEMO_2012-06-12_014.CSV
Project :           \StorageCard\appl\CONTROLSTATUSDEMO
Program Version :   10.0.0 (12345)
```

The actual log entries will follow:

Timestamp	**User**	**Event**	**Tag/Function Type name**	**Old Value**	**New Value**
2012-05-23 08:00:30		<Info>	<Startlog>		
2012-05-23 08:05:15		<Function>	<Userlogin>		Müller
2012-05-23 14:20:12	Müller	<Tag Change>	<MachineSpeed>	100	120
2012-05-23 14:20:15	Müller	<Function>	<Userlogin>		Meier
2012-05-23 14:50:23	Meier	<Tag Change>	<ValveV1>	0	1
…					

…

When tag values are stored using the user logger, they will be stored with the format in which they are displayed, i.e., conversions and units will be applied.

→ You can exclude individual functions from being logged if necessary.
see → section "Ignore lists for the user logger", page 946.
To add tags to or exclude them from logging, use the tag table.
see → section "Working with the tags table", page 560.

14.1 Ignore lists for the user logger

To exclude specific functions from being logged, you will need to create what is referred to as an "ignore list."

- Functions that should not be logged

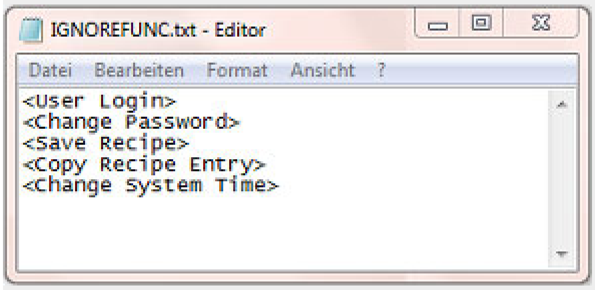
This file has a set name: "IGNOREFUNC.TXT" (for functions). It must be stored in ASCII format in the "DATA" directory.

The names for the functions must have the exact same syntax as in the log file (not case-sensitive).
One entry per line is permissible.
Each line must be separated by CRLF.

Look at an existing log file and copy the contents you want from it to an ignore list file.

Example of an ignore list for functions

```
Shut Down
Change Password
Execute external
```



```
<User Login>
<Change Password>
<Save Recipe>
<Copy Recipe Entry>
<Change System Time>
```

Tags are not removed from the user logger with an "ignore list." Instead, the logging behavior is managed in the tag table; please refer to
→ section "Working with the tags table", page 560

14.2 List of functions logged

Function	Item for Ignore List	Functions
Start of logging at startup	Start Logging at Startup	–
Start of logging	Start Logging	→ "Logger ON-OFF", page 951
Stopping of logging	Stop Logging	→ "Logger ON-OFF", page 951

General

Function	Item for Ignore List	Functions
System restart	Boot System	→ "Boot System", page 1110
Exit GRSW	Shut Down	→ "Shut down", page 1125
GRSW restart	Restart Runtime	→ "Restart Runtime", page 1116
Runtime finished	Runtime Closed	–
Change system time	Change System Time	→ "Set hour", page 1120→ "Set minute", page 1121, → "Set hour", page 1120, → "Set day", page 1119, → "Set year", page 1123
DST time setting	Switch Summer/Winter Time	→ "Summer time On/Off", page 1125
Delete entire error history	Clear All Histories	→ "Clear all Histories", page 882
Delete error history of current screen	Clear History in screen	→ "Clear History in Screen", page 882
Delete system history	Clear System History	→ "Clear System History", page 882
Call up of an external program	Execute external	→ "Execute", page 1114, → "Execute (dynamic)", page 1114

User management

Function	Item for Ignore List	Functions
Logging in of a user	User Login	→ "Login User", page 910 → "Login User x", page 911 → "Lock/Unlock User -userdef.", page 910
Logging out of a user	User Logout	→ "Logout User", page 912
Automatic logout of a user	Auto Logout	–
Modify the group assignment of a user	Change Group	→ "Change Group", page 902,
Change the password of a user	Change Password	→ "Change PW (own)", page 906→ "Change PW (other User)/ ChangePWOther", page 905 → "Change PW (User x)", page 903→ "Change PW (User x) -userdef.", page 904→ "Change PW (other User) -userdef.", page 904 → "Change PW (Own) -userdef.", page 904
Delete a user	Delete User	→ "Delete User", page 908

Function	Item for Ignore List	Functions
		→ "Delete User -userdef.", page 907
Lock a user	Lock User	→ "Lock/Unlock User", page 909 → "Lock/Unlock User -userdef.", page 910
Unlock a user	Unlock User	→ "Lock/Unlock User", page 909 → "Lock/Unlock User -userdef.", page 910
Define a new user	New User	→ "New User", page 913 → "New User -userdef.", page 912

Recipe handling

Function	Item for Ignore List	Functions
Start loading a recipe	Load Recipe Start	–
End loading a recipe	Load Recipe End	–
Save recipe	Save Recipe	→ "Save Recipe Entry", page 782 → "Save Recipe Entry As", page 783 → "Save Recipe Entry As userdef.", page 781 → "Save Recipe Entry As Var", page 783
Save recipe at next free location	Recipe Save to Next	→ "Save As to next", page 779 → "Save As to next userdef", page 780 → "Save As to next Var", page 780
Save recipe at new location	Recipe New to Next	→ "New to next", page 775 → "New to next userdef.", page 776
Save and load recipe	Save Recipe and Load	→ "Save As/Load Recipe Entry", page 782 → "Save As/Load Recipe Entry userdef.", page 779
Delete recipe	Clear Recipe	→ "Clear Recipe Entry", page 767
Rename recipe	Rename Recipe	→ "Change Recipe Entry Name", page 766 → "Change Recipe Entry Name userdef.", page 766
Copy recipe	Copy Recipe Entry	→ "Copy Recipe Entry", page 768
Paste recipe	Paste Recipe Entry	→ "Paste Recipe Entry", page 777
Copy recipe to next free location	Recipe Copy to Next	→ "Copy to next", page 767
Recipe Write protect	Protect Recipe	→ "Protect Recipe Entry", page 777
Recipe values redo	Restore Recipe	→ "Restore Default Values", page 778

Communication

Function	Item for Ignore List	Functions
Switch communication On/Off	Communication On/Off on Connection	→ "Comm. on Connection x ON/OFF", page 625
Switch communication persistent On/Off	Persistent on Connection	→ "Connection x persistent ON/OFF", page 625
Synchronization start	Start Sync to Connection	→ "Start Sync to Connection", page 626

Function	Item for Ignore List	Functions
Communication messages show/hide	Quiet On/Off on Connection	→ "Quiet on Connection x ON/OFF", page 626

14.3 User logger functions

14.3.1 Add String Tag Log

The value in a string tag can be written to the log file as a log entry.

Number of the Function	4099 (1003h)
Executable from the PLC	No
Function with ON/OFF state	No
Parameter 1	No
Function available in standard screens	Yes
Function available with keyboards	No
Function available in script	Yes
Function available in application entry script	Yes

14.3.2 Add Text Log

A predefined character sequence can be written into the protocol file as a log entry.

Number of the Function	5000 (1004h)
Executable from the PLC	No
Function with ON/OFF state	No
Parameter 1	Character sequence, which has to be written into the protocol file.
Function available in standard screens	Yes
Function available with keyboards	No
Function available in script	No
Function available in application entry script	Yes

14.3.3 Add Value Tag Log

The value in a value tag can be written to the log file as a log entry.

Number of the Function	4098 (1002h)
Executable from the PLC	No
Function with ON/OFF state	No
Parameter 1	Value tag, whose content has to be written into the protocol file.
Function available in standard screens	Yes
Function available with keyboards	No
Function available in script	Yes
Function available in application entry script	Yes

14.3.4 Flush Logger

The saving of log entries in the data buffer to the log file can be triggered manually with these functions.

Normally the runtime system automatically decides the time at which the collected data is written to the file.

Writing files too frequently may reduce the lifespan of the data storage device. Accordingly, this function should only be used in exceptional cases.

Number of the Function	4097 (1001h)
Executable from the PLC	Yes
Function with ON/OFF state	Yes
Function available in standard screens	Yes
Function available with keyboards	No
Function available in script	Yes
Function available in application entry script	Yes

14.3.5 Logger ON-OFF

Starts and ends user action logging.

Number of the Function	4096 (1000h)
Executable from the PLC	Yes
Function with ON/OFF state	Yes
Function available in standard screens	Yes
Function available with keyboards	No
Function available in script	Yes
Function available in application entry script	Yes

14.3.6 View Current Log File

The current log file can be displayed using an external program (e.g., NotepadCE).

To use the text editor that comes with Windows CE 5.0 (version 2.26.1 or higher), pass the following argument: "NotepadCE.exe"

Numbers of the Functions	5001 (1005h)
Executable from the PLC	No
Parameter1	File name of the program which will be used to display the protocol file.
Function with ON/OFF state	No
Function available in Standard screens	Yes
Function available with keyboards	No
Function available in script	Yes
Function available in application entry script	Yes

15. Scripts

Script Editor

In order to avoid having to integrate a PLC into a GALILEO project where unnecessary, you can use the easy-to-use Script Editor. This editor can be used to write short program codes to run specific functions in your project.

In GALILEO, functions can basically be implemented in two ways using:

Special functions – A function key is assigned to the visualization object in order to trigger a function

And

Script commands – The function is triggered using the GALILEO script language syntax in the Script Editor.

As soon as a script is selected, the Script tab will appear. This tab can be used to access and use the Script Editor's functions in an open script.

/*— —*/	Comment/Uncomment	Selected parts in the script
	Format	Formats the script based on prevalent conventions
	Find and Replace	Searches for the next occurrence of the selected string and replaces it, if applicable, with a different one

The editor features syntax highlighting and an autocomplete feature, making it easier to recognize and enter commands and tags.

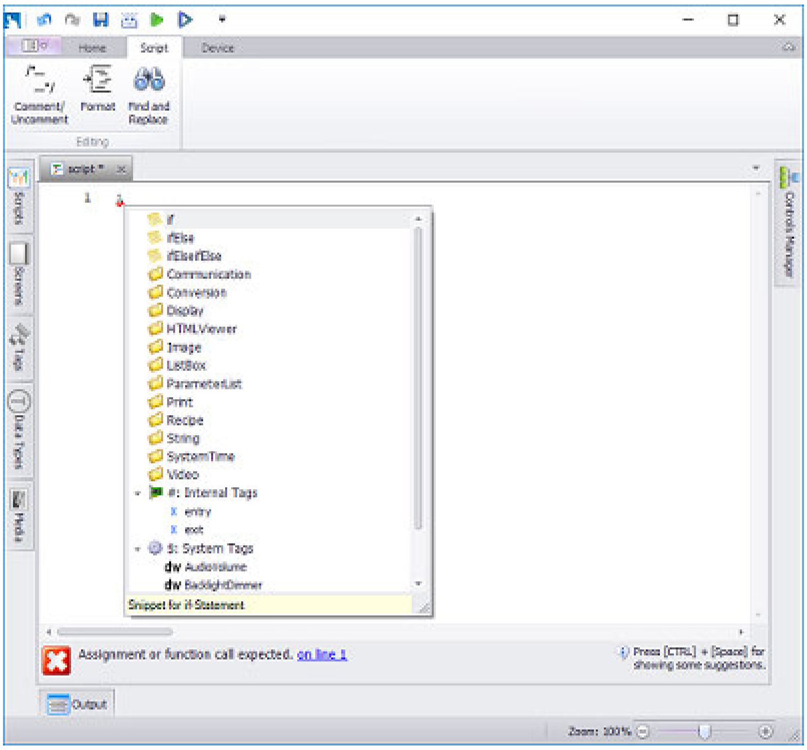
Syntax support in scripts

As soon as a GALILEO script language keyword is entered into a script, a code template for the corresponding statement will be offered as help. To use this template, click on the line you want and copy it by pressing the Enter key. The potential function or tag will then be offered for selection in the autocomplete box.

→ Unique naming
If an object such as a recipe or a graph block has the exact same name as a keyword
or a user, the script will not work.
If this happens, rename the object in question.

If an object such as a recipe, a graph block, or a user has the exact same name as a keyword, you will need to rename the object in the project.

15. Scripts



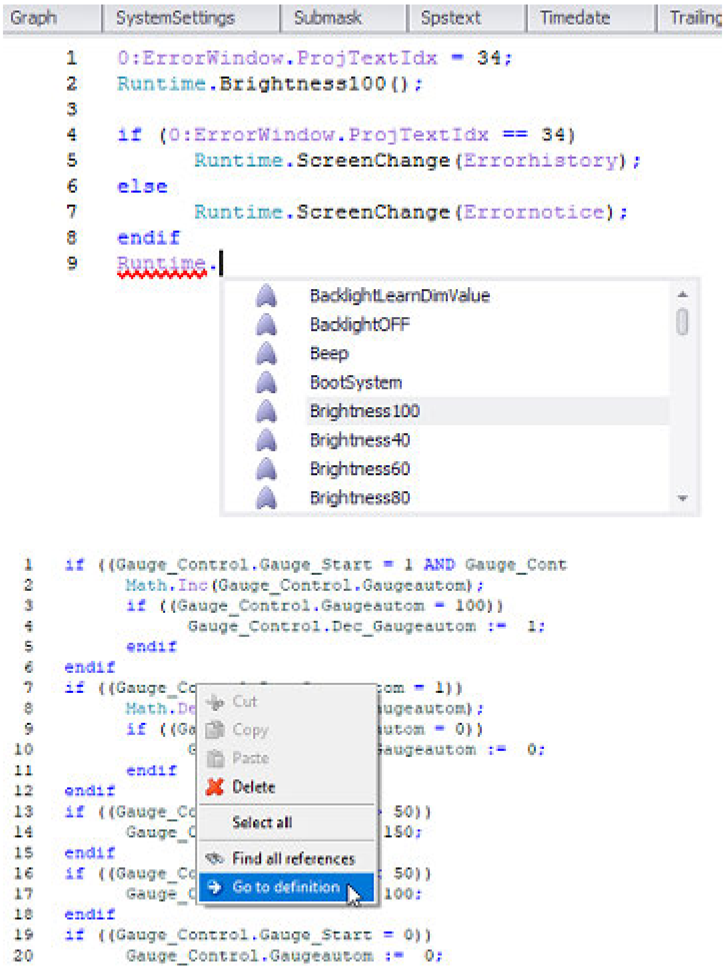
Snippet for if Statement

Assignment or function call expected. on line 1

Press [CTRL] + [Space] for showing some suggestions.

Scalable autocomplete box in the Script Editor

You can resize the autocomplete box in the Script Editor in order to make it easier to navigate and view long tag names.



```
0:ErrorWindow.ProjTextIdx = 34;
Runtime.Brightness100();

if (0:ErrorWindow.ProjTextIdx == 34)
        Runtime.ScreenChange(Errorhistory);
else
        Runtime.ScreenChange(Errornotice);
endif
Runtime.
```

```
if ((Gauge_Control.Gauge_Start = 1 AND Gauge_Cont
        Math.Inc(Gauge_Control.Gaugeautom);
        if ((Gauge_Control.Gaugeautom = 100))
                Gauge_Control.Dec_Gaugeautom :=  1;
        endif
endif
if ((Gauge_Co                    om = 1))
        Math.De                  ugeautom);
        if ((Ga                  utom = 0))
                G                augeautom :=  0;
        endif
endif
if ((Gauge_Co                     50))
        Gauge_C                  150;
endif
if ((Gauge_Co                     50))
        Gauge_C                  100;
endif
if ((Gauge_Control.Gauge_Start = 0))
        Gauge_Control.Gaugeautom :=  0;
```

Figure 590: Partial screenshots of Script Editor with context menu

Find all references	Searches for the visualization objects that call the selected event script or for the loop scripts that call the selected function. Display in the window Output/ Search Results
Go to definition	Jumps to the point in the project tree where, for example, the selected tag is defined.

How scripts are implemented: → "Using scripts to perform actions", page 956.

Scripts are written using text-based statements, just like in programming and macro languages. These scripts can be used to run multiple commands or actions one after the other.

The syntax check will be run in the compiler, and the result will be shown in the script status bar with the corresponding message, e.g., ✓ Script checked successfully.

Script calculation

The script calculation method matches the calculation method for other programming languages such as C# and CoDeSys.

Calculation examples

Assume that the "counter", "tmp", and "value" tags are all of type "WORD."

Calculation 1:

Counter: = 2;

tmp: = (Counter + 1) / 2

Value: = 2 * tmp

Calculation 2:

Counter: = 2;

Value: = 2 * ((Counter + 1) / 2)

The general expectation would be for these two operations to result in the same value for the "value" tag (i.e., a value of "2") because the mathematical calculations are the same.

See also

→ section "Using scripts to perform actions", page 956
→ section "Functions of the scripts", page 966
→ section "Using function keys to trigger actions", page 389

15.1 Using scripts to perform actions

15.1.1 Create a Script

To create a script, follow the steps below:

- Select the Scripts folder in *Project Tree\Scripts*.
- Open the context menu and select the New script option.

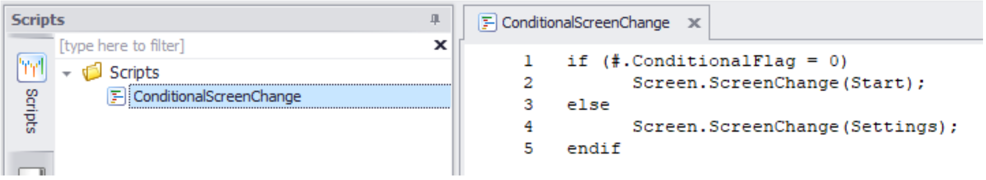
The script will be opened in the editor.

- Enter a name for the script.
- Enter the instruction you want for the script.

→ Pressing Ctrl + Space Bar will show a list of all available commands and command groups, as well as the tags available in the project.

For an overview of the syntax rules, please refer to the → section "Functions of the scripts", page 966 section.

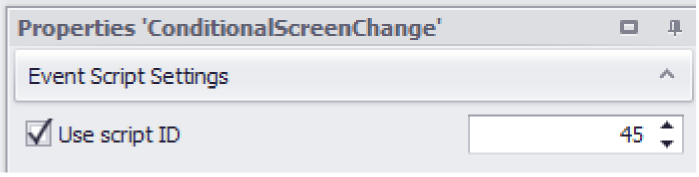
Home tab\Project Tree Scripts



```
if (#.ConditionalFlag = 0)
        Screen.ScreenChange(Start);
else
        Screen.ScreenChange(Settings);
endif
```

Figure 591: Script Editor with <ConditionalScreenChange> script

15.1.1.1 Script properties



☑ Use script ID 45

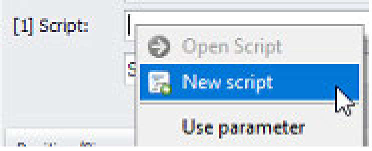
Figure 592: <ConditionalScreenChange> script properties

Use script ID: If this option is enabled, it will be possible for the PLC to trigger the corresponding script. To do this, the 5th data word in the Control structure must be used to enter the number (ID) of the script that should be triggered.

Alternative way to create a script

If the Properties pane for an object contains a section where a script can be selected, you can use the context menu for that section to immediately create a new script.

When you do this, the script will be assigned to the property and the Script Editor will be opened for editing.



15.1.2 Context menu Script

In addition to options for working with the scripts, the context menu provides access to a series of functions and tooltips.

However, the context menu for folders is different from the context menu for scripts!

→ In order for a script to be executed on the device, it must be ☑ active.
If a script is deactivated □, it will not be compiled and not assigned memory in the compiled code. It will be ignored.

New script	Insert new script.
Open script	Opens the script in the Script Editor
Find all references	Searches for the visualization objects that call the selected script. Display in the window Output/ Search Results

15.2 Local script tags

If, in the script, you will be using tags that are not directly related to the project but that are necessary for programming the script, these tags can be defined in the script itself.

This means that these tags do not need to be created in the project tree, since they are intended exclusively for use within the script and cannot be influenced outside of it.

Syntax:

```
var LocalVarName: Type;
```

Var:	Keyword used to declare local tags at the beginning of the script, right after any script parameters defined with the "param" keyword
LocalVarName:	The name of the local tag
Type:	The type of the local tag

You can declare multiple local tags. They must all be defined in the same line, separated by a ",":

```
var LocalVarName1: Type, LocalVarName2: Type, LocalVarName3: Type;
```

Arrays are supported as well

```
var LocalVarArrayName1: Type [Size];
```

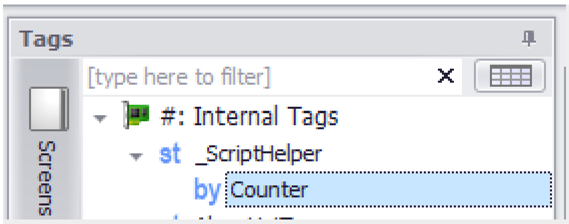
Size: The array size

The following types are supported:

Catalog Number	Description and value range
bit	Boolean (1 or 0, TRUE or FALSE)
bool	Boolean (1 or 0, TRUE or FALSE)
byte	Unsigned byte (0 to 255)
sbyte	Signed byte (-128 to 127)
word	Unsigned 16bit integer (0 to 65’535)
ushort	Unsigned 16bit integer (0 to 65’535)
short	Signed 16bit integer (-32’768 to 32’767)
dword	Unsigned 32bit integer (0 to 4’294’967’295)
uint	Unsigned 32bit integer (0 to 4’294’967’295)
int	Signed 32bit integer (-2’147’483’648 to 2’147’483’647)
float	Floating point ± 1.5e–45 to ± 3.4e38
string	String (the length must be entered between brackets, e.g.: string [10])

All local tags will have a starting value of 0.

Example featuring local script tags



Previously with local script tags:

```
if (x = 0)
    _ScriptHelper.Counter := _ScriptHelper.Counter + 1;
endif
if (y = 0)
    _ScriptHelper.Counter := _ScriptHelper.Counter + 1;
endif

if (z = 0)
    _ScriptHelper.Counter := _ScriptHelper.Counter + 1;
endif
if (_ScriptHelper.Counter > 1)
…
endif
```

Now with local script tags:

```
var counter: Byte;
    counter :=0;
if (x = 0)
    counter := counter + 1;
endif
if (y = 0)
    counter := counter + 1;
endif
if (z = 0)
    counter := counter + 1;
endif
if (counter > 1)
…
endif
…
```

15.3 Script parameters in the project

If you will be using identical scripts with identical logic but different tags for various tasks, you can define script parameters instead of tags.
The Galileo project tags will be passed as arguments

These script parameters need to be declared right at the start of the script.

After the script has been written, you can select the defined script parameters in every instance, for every execution, and for every application of the script.

Syntax:

```
param ParamVarName:[ref] Type;
```

Param: Keyword used to declare parameter tags
(has to be at the beginning of the script)

ParamVarName: The name of the parameter tag

[ref]: Optional keyword.
If a parameter tag is declared as "ref," changes to the passed tag will also be carried over to the project tag (at the end of the script execution)

Type: The type of parameter tag

You can also declare multiple parameter tags. In this case, they must all be declared on the same line, separated by a ",":

```
Param ParamVarName1: Type, ParamVarName2: Type, ParamVarName3: Type;
```

Arrays are supported as well

```
PAram ParamArrayName1: Type [Size];
```

Size: The array size

The following types are supported:

Keyword Type	Description and value range
bit	Boolean (1 or 0, TRUE or FALSE)
bool	Boolean (1 or 0, TRUE or FALSE)
byte	Unsigned byte (0 to 255)
sbyte	Signed byte (-128 to 127)
word	Unsigned 16bit integer (0 to 65’535)
ushort	Unsigned 16bit integer (0 to 65’535)
short	Signed 16bit integer (-32’768 to 32’767)
dword	Unsigned 32bit integer (0 to 4’294’967’295)
uint	Unsigned 32bit integer (0 to 4’294’967’295)
int	Signed 32bit integer (-2’147’483’648 to 2’147’483’647)

Keyword Type	Description and value range
qword	Unsigned 64bit integer (0 to 18’446’744’073’709’551’615)
ulong	Unsigned 64bit integer (0 to 18’446’744’073’709’551’615)
long	Signed 64bit integer (-9’223’372’036’854’775’808 to - 9’223’372’036’854’775’807)
float	Single precision floating point ±1.5e−45 to ±3.4e38
double	Double precision floating point ±5.0e−324 to ±1.7e308
error	Galileo Error tag (can only be used in read-only context within the script)
string	String (the length must be entered between brackets, e.g.: string[10])

Script parameter example

Two nearly identical scripts:

Script1

```
if (PLC_PRG.MachineCounter = 0)
   PLC_PRG.OnOffState := 0;
else
   PLC_PRG.OnOffState := 1;
endif
```

Script2

```
if (PLC_PRG.DriveCounter = 0)
   PLC_PRG.DriveState := 0;
else
   PLC_PRG.DriveState := 1;
endif
```

The use of parameters would mean that you would now only have to write and maintain one single script:

```
param ConditionalVar : byte, StateVar : ref bool;
if (ConditionalVar = 0)
   StateVar := 0;
else
   StateVar := 1;
endif
```

If you use the script in the project, transfer the "PLC_PRG.MachineCounter" and "PLC_PRG.OnOffState" tags as arguments for the parameters when you use it for the first time.

In contrast, you would pass "PLC_PRG.DriveCounter" and "PLC_PRG.DriveState" for script 2.

Parameters can be passed "by Value" or "by Reference."

By Value

Syntax:

```
param ParamName: Type;
```

If a parameter is passed "by Value," the script will work with a copy of the passed data. In other words, all changes to the parameter tags will only apply to the script, and the passed tag will remain unchanged.

Example

Say we had the following:

```
Param MyCounter: Byte;
MyCounter:= MyCounter + 1;
```

And that we passed the "DummyVar" Galileo tag as an argument for the "MyCounter" parameter. Moreover, assume that "DummyVar" had a value of 5 when passed to the script. After running the script, "DummyVar" would still have a value of 5.

By Reference

Syntax:

```
Param MyCounter: ref byte;
```

If a parameter is passed "by Reference," the script will share the data that is passed. In other words, all changes to the parameter tags in the script will also be reflected in the passed tag (at the end of the script execution).

Example

Say we had the following:

```
Param MyCounter: ref byte;
MyCounter:= MyCounter + 1;
```

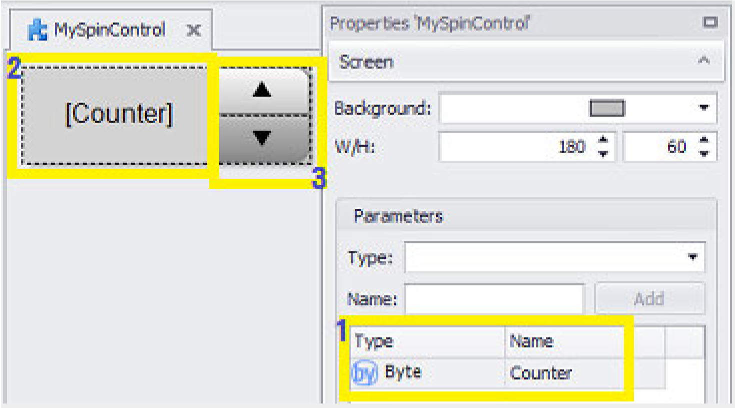
And that we passed the "DummyVar" Galileo tag as an argument for the "MyCounter" parameter. Moreover, assume that "DummyVar" had a value of 5 when passed to the script. After running the script, "DummyVar" would have a value of 6.

Using script parameters together with parameters in user controls

If you define user controls, you will have the option of defining and using (tag) parameters that will be filled with project data when the user control is used. Within the user control, you can also pass these parameters to a script as well.

Example

Say that the "MySpinControl" user control is defined:



Type	Name
Byte	Counter

1: With a parameter of type "byte"

2: With a "value display" object that shows the value of the "Counter" parameter

3: With two "function key" objects to which the "Call Script Function" function is assigned

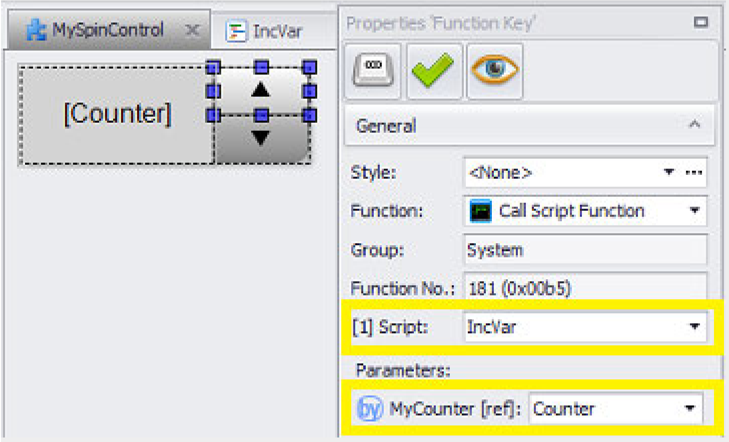
Now say that we have two scripts in the project.
Script 1, named "IncVar":

```
"IncVar":
Param MyCounter: ref byte;
MyCounter + = 1;
```

and Script 2 with the following content:

```
"DecVar":
Param MyCounter: ref byte;
MyCounter -= 1;
```

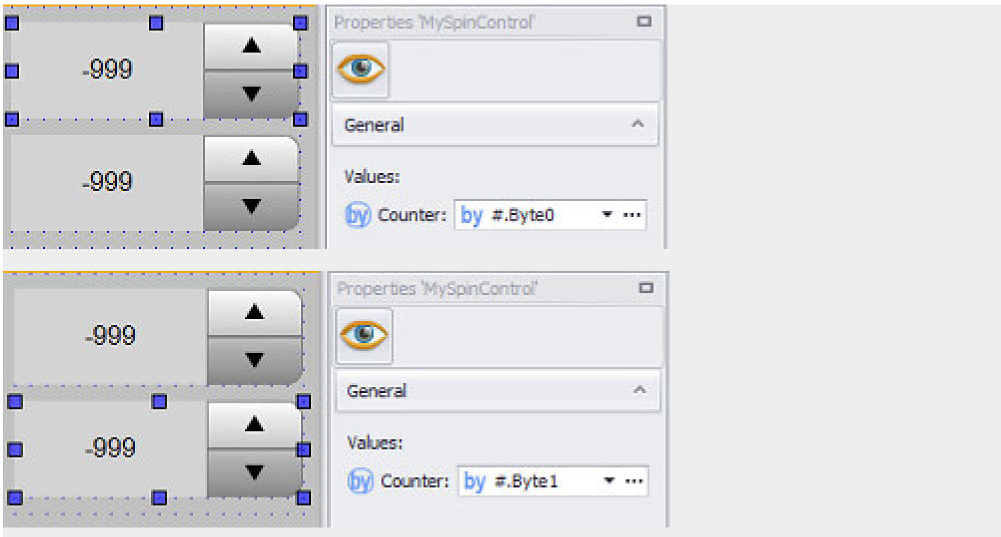
These two scripts are assigned to the two function keys, and the "Counter" parameter is passed as a script argument in both cases:



Style: <None>
Function: Call Script Function
Group: System
Function No.: 181 (0x00b5)
[1] Script: IncVar
Parameters:
MyCounter [ref]: Counter

The user control parameter must be passed as a parameter to the script.

This means that there is now a kind of "numeric spin control" that can be used anywhere in the project with numeric tags.



Properties 'MySpinControl'

General

Values:

Counter: by #.Byte0

Properties 'MySpinControl'

General

Values:

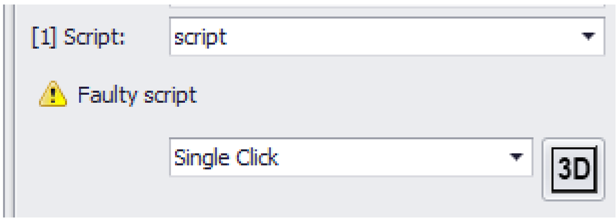
Counter: by #.Byte1

Limitations:

- It is usually not possible to use arrays as parameters for user controls. Needless to say, this limits the potential uses of parameters somewhat. However, there is a workaround you can use to pass arrays anyway, which is to use a custom data type that contains the array.
- It is not possible to simultaneously use, in combination, tag and script parameters with a user control. This means that in order to be able to pass parameters to scripts, the tags must be project tags.

Special notes regarding behavior.

- When a script has certain errors in it, it will not be possible to determine the script parameters correctly.
  If this script is assigned to a special function, a corresponding warning will be shown:



[1] Script: script

Faulty script

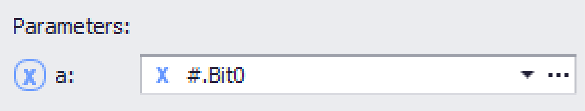
Single Click

- If the number or sequence of parameters within a script is changed, the special functions making the calls will not automatically change accordingly. This can result in parameter assignments shifting.

Example showing how parameter assignments can shift

Script parameter list

```
"script0": param a: bit;
```



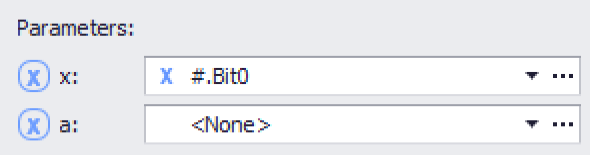
Parameters:

x a: x #.Bit0

Say the script changes as follows:

```
Param x : BEs, a: bit;
```

The special function will now show:



Parameters:

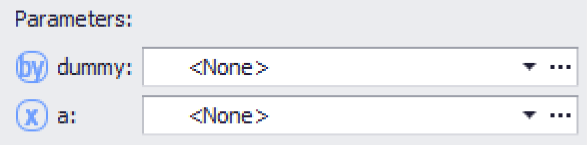
x x: x #.Bit0

x a: <None>

The list of arguments remains the same. The first script parameter is a "bit" that is compatible with the previously assigned first argument.

Say the script at the start of this example changes as follows:

param dummy: Byte, a: bit;



Parameters:

by dummy: <None>

x a: <None>

The first script parameter is now a "byte."
The previously assigned first argument was a "bit" tag, which is now no longer compatible.

15.4 Functions of the scripts

The following functions are already integrated into the GALILEO Runtime System firmware and cannot be expanded or modified by the user.

The names of these functions are always in English.

In GALILEO, functions are organized into various functional groups. The use of specific functions will depend on the script being used.

Groups	
Alarm	Alarm management functions
Communication	Communication Functions
Conversion	Conversion Functions
Display	Functions relevant to the device display
Event	available in loop scripts only
Graph Block	Graph Functions
Image	Image functions
HTMLViewer	Used to integrate URL addresses
Language	Functions for changing the language
ListBox	Functions for navigating through a list
Math	Advanced mathematical operations
Parameter List	Functions for navigating through a parameter list
Source file, print resolution (EPS, TIF)	Printer Functions
Recipe	Various recipe functions
Screen	Functions relevant to the screen
String	Text functions
System	Runtime Functions
SystemTime	Functions relevant to the device's or PLC's time
UserLogger	User Logger Functions
UserManagement	User management functions
Video	Functions used to control the video player
Webcam	Functions used to control the webcam
Operators	Arithmetic expressions

Functions

In GALILEO, functions can basically be implemented in two ways using:

Special functions – A function key is assigned to the visualization object in order to trigger a function

And

Script commands – The function is triggered using the GALILEO script language syntax in the Script Editor.

Some functions can only be called with the use of script commands.
Certain functions represent **s**pecial functions that can also be accessed with a

function key visualization object. These functions are marked with √ in the lists below. In this case, the corresponding parameters and the way the functions work will be the same as for the corresponding special functions.

The following tables show the various functions and the groups to which they have been assigned :

Alarm

Script command	Function	S
Clear all Histories	The history entries of the entire project are deleted. The history of the system error messages is preserved.	√
Clear History in Screen	Clears the history for the current alarm list. This function can only be used if the current screen contains an alarm list object. This function will not do anything in screens that do not contain an alarm list object.	√
Clear System History	Clears the system error message history	√
Error Autoscroll On/Off	Displays most recent error in an alarm list	√
Error confirm	Resets a selected error message in an alarm list object	√
Error confirm (actual window)	Resets the error messages in the alarm list on the current screen.	√
Error confirm all	Resets all existing error messages	√
Error Priority ON/OFF	Defines the order for the entries in the alarm list	√
Error Refresh	If error tags are used in scripts, the "Error Refresh" instruction must be called beforehand so that the values will be refreshed by the PLC. In order to avoid placing an excessive load on communications, this instruction should not be called cyclically.	–
Save Error History as File	Saves the error history in a file.	√
Save Error History as File Var	Saves the error history in a file defined by a tag	√
Save System History as File	Saves the system error messages in a file	√
Save System History as File Var	Saves the system error messages in a file defined by a tag	√
Show History On/Off	This function switches the alarm list visualization object to the history display. The error history is used to log error activities.	√
Show Sys. History On/Off	This function switches the alarm list to the history for the system error messages	√
Save History to CSV	This function saves the error messages in a .csv file.	√
Save History to CSV Var	This function saves the error messages in a .csv file. The path is passed as a tag.	√
Save History of previous day to CSV	This function saves the error messages of the last day in a .csv file	√
Save History of previous day to CSV Var	This function saves the error messages of the last day in a .csv file The path is passed as a tag.	√

Communication

Functions	Description	S
Comm. on Connection x ON/OFF	This function can be used to enable and disable communications at the corresponding interface number. If	√

	the device is restarted, communication will be enabled for all interfaces.	
Connection x persistent ON/OFF	This function can be used to enable and disable communications at the corresponding interface number. The selected state will be retained even after the device is restarted.	√
Quiet on Connection x ON/OFF	Used to show or ignore communication error messages for the corresponding interface.	√

Conversion

Functions	Description	S
Conversion of Group	Enables value conversion for a group.	√
Next Conversion of Group	Enables the next value conversion for a group.	√

Display

Functions	Description	S
Backlight: Learn Dim Value	Sets the current backlight brightness as a reduction value for the screensaver.	√
Backlight Off	Turns off the display's backlight	√
→ "Brightness 40%", page 1111	Reduces the display's brightness to 40%	√
Brightness 60%	Reduces the display's brightness to 60%	√
Brightness 80%	Reduces the display's brightness to 80%	√
Brightness 100%	Sets the display's brightness to the maximum value	√
Brightness Dynamic	Sets the display's brightness to the value of the tag passed	√
Brightness Static	Sets the display's brightness to the percentage value of the backlight setting	√
Set Touch Lock	Disables the touchscreen	√

Event

→ The functions in the Event group are only available in loop scripts.

Functions	Description	S
OnChangeTag	Check tag for change	–
OnInterval_daily	Checks day interval (specified time of a day, in which one or several weekdays can be specified)	–
OnInterval_dailyVar	Checks day interval (specified time of a day on the basis of a tag value)	–
OnInterval_hourly	Checks x minutes after the hour	–
OnInterval_hourlyVar	Checks x minutes after the hour on the basis of a tag value	–
OnInterval_min	Checks minute interval (x minutes)	–
OnInterval_minVar	Checks minutes interval (number of minutes on the basis of a tag value)	–
OnInterval_monthly	Checks the month interval (specific time of a specific day of the month)	–
OnInterval_monthlyVar	Checks the month interval (specific time of a specific day of the month on the basis of the tag values)	–
OnInterval_sec	Checks seconds interval (x seconds)	–
OnInterval_weekly	Checks the week interval (specific time of day in a specific	–

	weekday)	
OnInterval_weeklyVar	Checks the week interval (specific time of day on a specific weekday on the basis of a tag value)	–
OnInterval_yearly	Checks the year interval (specific time of day on a specific date of the year)	–
OnInterval_yearlyVar	Checks the year interval (specific time on a specific day of the year on the basis of the tag values)	–
OnPresetOK	Check whether data synchronization was successfully carried out	–
OnTime	Is true if the time/date entry matches	–
OnTimeVar	Is true if the time/date entry matches on the basis of tags	–
OnRecipeLoad	The function will return true if a new recipe entry has been loaded	–

Graph Blocks

Functions	Description	S
Delete Archive File	Delete an archive file	√
Delete Graph Buffer	Deletes the recorded values for a graph	√
Delete Viewer Buffer	Deletes the display in the Graph Blocks Control that is marked as the Graph Viewer.	√
Load Archive File Var	Loads an archive file defined by a tag	√
Load Archive File Full Path As Var	Loads an archive file defined by a tag that specifies the full path	√
Save Archive File	Saving an archive file	√
Save Archive File As Var	Saves an archive file using a name defined by a tag	√
Save Archive File Full Path As Var	Saves an archive file using a name defined by a tag that specifies the full path	√
Start/Stop Graph Block	Starts/stops plotting for a graph	√
View Archive File Var	Displays an archive in the graph viewer object	√
View Archive File Full Path As Var	Displays an archive file defined by a tag that specifies the full path.	√

HTML Viewer

Functions	Description	S
Load URL	Used to load other URL addresses (*.htm, *.html)	
Load URL Var	Used to load other URL addresses as a tag	

Image

Functions	Description	S
Load Image	Loads an image and shows it on the screen.	√
Load Image Var	Loads an image specified in a tag and shows it on the screen.	
Select Image	Opens a user keyboard for selecting an image that should be displayed on the screen.	
Select Image Var	Opens a user keyboard for selecting an image that should be displayed on the screen. The image directory is defined in a tag.	

Language

Functions	Description	S

→ "Change to language", page 660	Changes the language. The desired language is passed as an argument.	√

ListBox

Script command	Description	S
	Jumps to the first item in the list box.	√
	Jumps to the last item in the list box.	√
	Moves to the next Item in the List box.	√
	Moves to the previous Item in the List box.	√
	Jumps to the next page in the list box.	√
	Moves to the previous page in the List box.	√
	Jumps to the item that was specified as a parameter for the item No.	
	Jumps to the item that was specified as a parameter for the tag. The tag contains the item No.	

Math

Functions	Description	S
→ "BitwiseAND", page 979	Bitwise AND operation on a tag and a fixed value.	√
→ "BitwiseANDVar", page 979	Bitwise AND operation on a tag and the value of a second tag.	√
→ "BitwiseNAND", page 979	Bitwise NAND operation on a tag and a fixed value.	√
→ "BitwiseNANDVar", page 980	Bitwise NAND operation on a tag and the value of a second tag.	√
→ "BitwiseNOR", page 980	Bitwise NOR operation on a tag and a fixed value.	√
→ "BitwiseNORVar", page 980	Bitwise NOR operation on a tag and the value of a second tag.	√
→ "BitwiseNOT", page 980	Bitwise NOT operation on a tag.	√
→ "BitwiseOR", page 980	Bitwise OR operation on a tag and a fixed value.	√
→ "BitwiseORVar", page 981	Bitwise OR operation on a tag and the value of a second tag.	√
→ "BitwiseXOR", page 981	Bitwise XOR operation on a tag and a fixed value.	√
→ "BitwiseXORVar", page 981	Bitwise XOR operation on a tag and the value of a second tag.	√
→ "CheckBitSet", page 981	Checks whether the xth bit of the tag is set	√
→ "InvertBit", page 984	Invert the xth bit of a tag.	√
→ "ResetBit", page 993	Reset the xth bit of the tag (set to 0).	√
→ "SetBit", page 994	Set the xth bit of the tag (set to 1).	√

Print

Functions	Description	S
→ "Print History", page 1033	Prints the History of the accumulated error messages.	√
→ "Print Parameter List", page 1033	Prints the specified parameter list.	√
→ "Print Parameter List (no Header)", page 1034	Prints the specified parameter list without the header.	√
→ "Print Report", page 1034	Produce a print report	√
→ "Print Screen", page	Prints the current screen.	√

1034		

Recipe

Functions	Description	S
First Line (Recipe)	Jumps to the first entry in the recipe	√
Last Line	Jumps to the last entry in the recipe	√
Line Down	Jumps to the next entry in the recipe	√
Line Up	Jumps to the previous entry in the recipe	√
LoadLastRecipeEntry	Loads the values for the recipe entry that was last loaded	–
Load Recipe Entry	Loads the values for the recipe entry currently selected	√
LoadRecipeEntryVar	Loads the values for the recipe entry passed with a tag	–
LoadSelectedRecipeEntry	Loads the values for the recipe entry currently selected	–
Page Down	Jumps to the next page in the recipe	√
Page Up	Jumps to the previous page in the recipe	√
SaveActualRecipeEntry	Saves the current tag values, under the recipe entry currently loaded, in the recipe file.	–
Refresh Labels	This function reads the recipe entry names from the recipe file again.	√
Save As to next Var	Saves the current tag values, under the specified name, in the next free recipe entry in the recipe file.	√
Save Recipe Entry	Saves the current tag values under the recipe entry number in the recipe file.	√
Save Recipe Entry As Var	Saves the current tag values, under the name passed with a tag, in the recipe file.	√
Save Recipe Entry Var	Saves the current tag values in the recipe entry passed with a tag.	√

Screen

Functions	Description	S
OpenDialog	Opens a specific dialog box	√
CloseAllDialogs	Closes all open dialog boxes	√
CloseDialog	Closes the frontmost dialog box	√
CloseMenu	Closes the menu that is being actively displayed	–
GetActScreenNo	Determines the number of the currently visible mask.	–
Last Screen	Switches to the previous screen.	√
Save Screen as Bitmap	Saves the screen that is currently being displayed as a .bmp file in the project's data directory.	√
Save Screen as Bitmap Var	Saves the screen that is currently being displayed as a .bmp file. The path (including the filename) is passed as a tag.	√
ScreenChange	Changes the screen.	–
ShowMenu	Shows a defined menu	–

String

Functions	Description	S
Append	Adds the content of a String Tag to another String Tag	–
Assign	Assigns the content of one String Tag to another String Tag	–
AppendConstString	Adds a constant string value to a string tag	–

AssignConstString	Assigns a string tag the value of a constant string.	–
Compare	Compares the contents of two string tags. The comparison is not case-sensitive.	–
CompareCaseSens	Compares the contents of two string tags. These passwords are case-sensitive.	–
CompareConst	Compares the value of a string tag with a constant string. Not case-sensitive.	
CompareConstCaseSens	Compares the value of a string tag with a constant string. Case-sensitive.	
SetErrTextConst	Writes an error text to a string tag.	–
SetErrTextVar	Writes an error text to a string tag. The value of the tag passed specifies the ID of the error text that should be written.	–
SetTextConst	Writes a help text to a string tag.	–
SetHlpTextVar	Writes a help text to a string tag. The value of the tag passed specifies the ID of the help text that should be written.	–
SetTextConst	Assigns a Standard Text to a String Tag	–
SetTextDate	Writes the current date to a string tag.	–
SetTextDateTime	Writes the current date and time to a string tag.	–
SetTextTime	Writes the current time to a string tag.	–
SetTextVar	Writes a text to a string tag. The value of the tag passed specifies the ID of the text that should be written.	–

System

Functions	Description	S
Beep	Starts an audible signal from the PLC.	√
Boot System	Reboots the device	√
Call DLL Function	Calls a function within an external DLL.	√
Execute	Starts a program	√
Execute (dynamic)	Starts a program defined by the tag passed	√
→ "GetTickCount", page 1115	Returns the system time in milliseconds. Can be used to conduct performance analyses.	–
→ "ReReadTag", page 1116	Reads a tag from the communication system again.	–
→ "Restart Runtime", page 1116	Restarts runtime.	√
→ "Script", page 1118	Calls a script	Call Script Function
→ "Select ASCII Tag", page 1118	Selects a string tag for the input	√
→ "Select Numeric Tag", page 1119	Selects a numeric tag for the input	√
→ "Show Frame On/Off", page 1124	This function shows and hides the frame, incl. the window title, for the GRS sequential program.	√
→ "Shut down", page 1125	Shuts down the runtime system.	√
→ "SyncWriteTag", page 1126	Synchronously writes a tag value. The script will be aborted and will not be resumed until the tag has been written.	–

System Time

Functions	Description	S
→ "ReadPLCDateTime", page 1116	Reads the "Date_Time" system structure from the communication system passed as an argument.	√
→ "Summer time On/Off", page 1125	This function allows the internal clock to be advanced or delayed by one hour.	√
→ "US date and time On/Off", page 1126	This function is used to toggle between the US and 24 hour time/date display format.	√
→ "WritePLCDateTime", page 1128	Writes the "Date_Time" system structure to the communication system passed as an argument.	√
→ "WriteCalendarWeekToValue", page 1126	Writes the current calendar week to a tag defined as a parameter.	
→ "WriteDateTimeToValues", page 1127	Writes the current system time to tags defined as parameters.	
→ "WriteDateTimeFromValues", page 1127	Sets the current system date and time based on tags defined as parameters.	
→ "WriteDateFromValues", page 1127	Sets the current system date based on tags defined as parameters.	
→ "WriteTimeFromValues", page 1128	Sets the current system time based on tags defined as parameters.	
→ "WriteWeekdayToValue", page 1129	Writes the current weekday to a tag defined as a parameter.	

User Logger

Functions	Description	S
→ "Add String Tag Log", page 950	Writes the value of a string tag as a log entry to the log file.	√
→ "Add Text Log", page 950	Writes a set string as a log entry to the log file.	√
→ "Add Value Tag Log", page 950	Writes the value of a numeric tag as a log entry to the log file.	√
→ "Flush Logger", page 950	Makes the log entries in the data buffer be saved in the log file.	√
→ "Logger ON-OFF", page 951	Starts/Ends the recording of user actions.	√
→ "View Current Log File", page 951	Displays the current log file with the help of an external program	√

UserManagement

Functions	Description	S
→ "Change PW (other User)/ ChangePWOther", page 905	Used to change the password for a user.	–
→ "Change PW (own)", page 906	Changes the password of the currently logged in user	–
→ "Login User", page 910	Calls the user login screen.	√
→ "Login User -userdef.", page 911	Calls the custom user login screen	√
→ "Login User x", page 911	Logs a specific user in	√
→ "Login User x -userdef.", page 912	Calls the custom screen for logging in a specific user	√
→ "Logout User", page 912	Logs the current user out	√
→ "New User", page 913	Used to create a new user.	√
→ "RefreshAutoLogoutTime", page	If the auto logout time for the specified group is controlled	√

914	with a tag, the time will be refreshed with the current tag value.	
→ "UserLogin", page 919	Log in a user	–

Webcam

Functions	Description	S
→ "WebCam:Capture Image", page 519	Saves the current webcam image. The filename is defined in the GALILEO application.	√
→ "WebCam:Capture Image As", page 520	Saves the current webcam image. The device operator will need to enter the filename at runtime with the use of a keyboard.	√
→ "WebCam:Capture Image Var", page 520	Saves the current webcam image. The filename must be specified in a tag of data type STRING.	√
→ "WebCam:Capture Image Var As", page 521	Saves the current screen as a bitmap	√
→ "WebCam:Start", page 522	Starts the webcam.	√
→ "WebCam:Stop", page 522	Stops the webcam.	√

Video

Functions	Description	S
→ "Video:Load", page 502	This function can be used to load a video file in the video player visualization object.	√
→ "Video:LoadVar", page 503	This function can be used to load a video file in the video player visualization object. The name of the video player visualization object must be specified in a tag of data type STRING.	√
→ "Video:Pause", page 503	Pauses the video player visualization object.	√
→ "Video:Select", page 504	A system keyboard will appear at runtime so that the device operator can select one of the video files in a specific directory. This directory is defined in the GALILEO application.	√
→ "Video:SelectVar", page 504	A system keyboard will appear at runtime so that the device operator can select one of the video files in a specific directory. This directory must be specified in a tag of data type STRING.	√
→ "Video:Start", page 505	Starts the video player visualization object.	√
→ "Video:Stop", page 505	Stops the video player visualization object.	

Operators

All operators are available in loop and event scripts.

Operand	Description
=	Compare: equal
<>	"Not equal to" comparison
<	Compare: less than
>	Compare: greater than
>=	Compare: greater than/equal to
<=	Compare: less than/equal to
+	Addition
+=	Increases the value of a tag

-	Subtraction
-=	Decreases the value of a tag
*	Multiplication
/	Division
()	Brackets
%	Modulo
^	Potency
AND	Logic AND sequence
OR	Logic OR sequence
NOT()	Logic NOT
TRUE	Logic constant TRUE
FALSE	Logic constant FALSE

When calculating mathematical expressions internally, the device will use decimal places. The value will be truncated into an integer only after the calculation is complete. Accordingly, the only inaccuracy in the calculation is when decimal places are left out at the end.

See also

→ section "Mathematical operations", page 976

15.4.1 //

Adds a comment line.

Comments in the script are shown in green. Comment lines do not affect the device's behavior and can be used to describe the script or to comment out script lines.

→ section "SwitchRight endless", page 999

→ section "Using scripts to perform actions", page 956

15.4.2 Append

Adds the content of the source tag to the destination tag.

If the target tag does not have enough space for the source tag's content, the remaining characters will be truncated.

Parameter entered	Name of a string tag (Target Tag)
Parameter entered	Name of a string tag (Source Tag)
Function available in script	Yes
Function available in application entry script	No

15.4.3 AppendConstString

Adds the content of a string tag as a constant value to the target tag.

→ If the target tag does not have enough space for the source tag's content, the remaining characters will be truncated.

Parameter 1	Name of a string tag (Target Tag)
Parameter 2	Value of a string in quotation marks "..."
Function available in script	Yes
Function available in application entry script	No

15.4.4 Mathematical operations

The following are available for arithmetic operations in the Script Editor:

- Comparison operators
- Logical operators

  And
- Arithmetic operators

Assigning a value

Values are assigned with the following syntax:":="

Value assignment example

"MyTag := 4;"

Arithmetic operators	Description Operator for
+	Addition
+=	Increases the value of a tag
-	Subtraction
-=	Decreases the value of a tag
*	Multiplication
/	Division
()	Brackets
%	Modulo
^	Potency

Arithmetic operator example

"MyTag := (MyValue * 9/5) + 32;"

Operators "+ =" and "- ="

Syntax: "TagName + = Value;" / "TagName-= Value;"

TagName: The tag with the value that should be changed

Value: The value by which the tag will be increased or decreased.

"+ =" operator example

In order to increase a value, an assignment in which the tag is on both sides of the assignment can be used. This can quickly become confusing in the case of long tag names:

PLC _ PRG. Motor1.ParameterSet.X: = PLC _ PRG. Motor1.ParameterSet.X + 10;

or

The "+ =" operator can be used to simplify the assignment:

PLC _ PRG. Motor1.ParameterSet.X + = 10;

Comparison operators	Description Type of comparison
=	Equals
<>	Does not equal
<	Less than
>	Greater than
>=	Greater than/equal
<=	Less than/equal

Logical operators	Description
AND	Logic AND sequence
OR	Logic OR sequence
NOT()	Logic NOT
TRUE	Logic constant TRUE
FALSE	Logic constant FALSE
Round()	Round the value

→ When calculating mathematical expressions internally, the device will use decimal places. The value will be truncated into an integer only after the calculation is complete. Accordingly, the only inaccuracy in the calculation is when decimal places are left out at the end.

15.4.5 Array initialization

There are two ways to initialize a tag array:

- By adding a line for each array element in order to initialize the tag array with a value, e.g.:

```
MyTagArray[0]: = 1;
MyTagArray[1]: = 1;
MyTagArray[2]: = 1;
MyTagArray[3]: = 1;
```

or

- By writing this initialization in a single line, e.g.:

  MyTagArray[0..3]: = 1;

Syntax:

```
ArrayName [StartIndex .. Endindex]: = Value;
```

Arrayname: The tag array being initialized
Startindex: The index of the first array element being initialized
Endindex: The index of the last array element being initialized
Value: The value that should be assigned to the array elements

15.4.6 Array and UDT instance assignment

There are two ways to copy an array or a UDT instance:

- By adding a line to each array element and each UDT instance tag in order to copy the value, e.g.:

```
MyTagArray[0]: = TagArrayX[0];
MyTagArray[1]: = TagArrayX[1];
MyTagArray[2]: = TagArrayX[2];
MyTagArray[3]: = TagArrayX[3];

MyMotor1.OnOff: = MyMotor2.OnOff;
MyMotor1.Speed: = MyMotor2.Speed;
MyMotor1.Param[0]: = MyMotor2.Param[0]
MyMotor1.Param[1]: = MyMotor2.Param[1]
MyMotor1.Param[2]: = MyMotor2.Param[2]
MyMotor1.Param[3]: = MyMotor2.Param[3]
```

or

- - Or by writing this copy operation in a single line:

  MyTagArray: = TagArrayX;

  MyMotor1: = MyMotor2;

Syntax:

```
TargetArray: = SourceArray;
TargetUdtInstance: = SourceUdtInstance;
```

TargetArray/SourceArray:
Copies the values from the SourceArray to the TargetArray.
SourceArray and TargetArray are both tag arrays.

TargetUdtInstance/SourceUdtInstance:
Copies the values from the SourceUdtInstance to the TargetUdtInstance.
SourceUdtInstance and TargetUdtInstance are both instances of the same UDT tag.

15.4.7 Assign

Assigns the content of one String Tag to another String Tag.

→ If the target tag does not have enough space for the source tag's content, the remaining characters will be truncated.

Parameter entered	Name of a string tag (Target Tag)
Parameter entered	Name of a string tag (Source Tag)
Function available in script	Yes
Function available in application entry script	No

15.4.8 AssignConstString

Assigns a string tag the value of a constant string.

→ If there is not enough space in the string tag for the constant string, the remaining characters will be truncated.

Parameter 1	Name of a string tag (Target Tag)
Parameter 2	Value of a string in quotation marks "..."
Function available in script	Yes
Function available in application entry script	No

15.4.9 BitwiseAND

Bitwise AND operation on a tag and a fixed value.
The result will be written directly to the tag.

Parameter entered	Name of the Tags
Parameter entered	Value for the AND operation
Function available in script	Yes
Function available in application entry script	Yes

15.4.10 BitwiseANDVar

Bitwise AND operation on a tag and the value of a second tag.
The result of the operation will be written directly to the first tag.

Parameter entered	Name of the first tag
Parameter entered	Name of the second tag for the AND operation
Function available in script	Yes
Function available in application entry script	Yes

15.4.11 BitwiseNAND

Bitwise NAND operation on a tag and a fixed value.
The result of the operation will be written directly to the tag.

Parameter entered	Name of the Tags
Parameter entered	Value for the NAND operation
Function available in script	Yes

Function available in application entry script	Yes

15.4.12 BitwiseNANDVar

Bitwise NAND operation on a tag and the value of a second tag.
The result of the operation will be written directly to the first tag.

Parameter entered	Name of the first tag
Parameter entered	Name of the second tag for the NAND operation
Function available in script	Yes
Function available in application entry script	Yes

15.4.13 BitwiseNOR

Bitwise NOR operation on a tag and a fixed value.
The result of the operation will be written directly to the tag.

Parameter entered	Name of the Tags
Parameter entered	Value for the NOR operation
Function available in script	Yes
Function available in application entry script	Yes

15.4.14 BitwiseNORVar

Bitwise NOR operation on a tag and the value of a second tag.
The result of the operation will be written directly to the first tag.

Parameter entered	Name of the first tag
Parameter entered	Name of the second tag for the NOR operation
Function available in script	Yes
Function available in application entry script	Yes

15.4.15 BitwiseNOT

Bitwise NOT operation on a tag.
The result of the operation will be written directly to the tag.

Parameter entered	Name of the Tags
Function available in script	Yes
Function available in application entry script	Yes

15.4.16 BitwiseOR

Bitwise OR operation on a tag and a fixed value.
The result of the operation will be written directly to the tag.

Parameter entered	Name of the Tags
Parameter entered	Value for the OR operation
Function available in script	Yes
Function available in application entry script	Yes

15.4.17 BitwiseORVar

Bitwise OR operation on a tag and the value of a second tag.
The result of the operation will be written directly to the first tag.

Parameter entered	Name of the first tag
Parameter entered	Name of the second tag for the OR operation
Function available in script	Yes
Function available in application entry script	Yes

15.4.18 BitwiseXOR

Bitwise XOR operation on a tag and a fixed value.
The result of the operation will be written directly to the tag.

Parameter entered	Name of the Tags
Parameter entered	Value for the XOR operation
Function available in script	Yes
Function available in application entry script	Yes

15.4.19 BitwiseXORVar

Bitwise XOR operation on a tag and the value of a second tag.
The result of the operation will be written directly to the first tag.

Parameter entered	Name of the first tag
Parameter entered	Name of the second tag for the XOR operation
Function available in script	Yes
Function available in application entry script	Yes

15.4.20 CheckBitSet

Checks whether the xth bit of the tag is set (1).

Parameter entered	Name of the tag that should be checked
Parameter entered	Bit number [0..x]
ReturnVal	True: If bit is set (1) False: If bit is not set (0)
Function available in script	Yes
Function available in application entry script	Yes

15.4.21 Compare

Compares the contents of two string tags. This comparison is not case-sensitive.

Parameter entered	Name of a string tag
Parameter entered	Name of a string tag
Return parameter	True if the content of both tags is the same
Function available in script	Yes
Function available in application entry script	No

15.4.22 CompareCaseSens

Compares the contents of two string tags. These names are case-sensitive.

Parameter entered	Name of a string tag
Parameter entered	Name of a string tag
Return parameter	True if the content of both tags is the same
Function available in script	Yes
Function available in application entry script	No

15.4.23 CompareConst

Compares the value between a string tag and a constant tag.
Not case-sensitive.

Parameter entered	Name of the string tag that should be compared
Parameter entered	Value of the string that will be used for the comparison
Return parameter	True if the content of both tags is the same
Function available in script	Yes
Function available in application entry script	No

15.4.24 CompareConstCaseSens

Compares the value of a string tag with a constant string. Case-sensitive.

Parameter entered 1	Name of the string tag that should be compared
Parameter entered 2	Value of the string that will be used for the comparison
Return parameter	True if the content of both strings is the same
Function available in script	Yes
Function available in application entry script	No

15.4.25 Dec

Reduce tag value by 1

Parameter entered	Name of the Tags
Function available in script	Yes
Function available in application entry script	No

15.4.26 Control structures - if, else, elseif, foreach

These commands can be used in various combinations in order to program conditionals with statements.

Use: if, endif

The "if" command describes the conditions required for the execution of particular instructions. The condition must always be defined in parentheses. An "if" command should always have an "endif" command to end the condition concerned. If the condition is true, the commands between the "if" and "endif" command are processed.

If the condition is not true, however, the program jumps directly to the command after the "endif" command.

The result of the conditions must always be a Boolean value.

Use: if, else, endif

If required the "else" command can be used between the "if" and "endif" commands. If the condition is true, the commands between the "if" and "else" command are processed. If, however, the condition is not true (i.e. not fulfilled), the commands between "else" and "endif" are processed.

elseif branch

Can be used to expand an "if" control structure with alternative instructions if the original "if" condition is not met.

The "elseif" statement makes it possible to write code that is easier to read without having to nest a large number of "else" followed by "if" statements.

The conditions will be evaluated in the corresponding order until one of them is met. Once one of these conditions is met, the statement corresponding to it will be executed and the multiple branching structure will end. If none of the conditions is met, the else "part" will be executed (if there is one).

foreach, endforeach loops

The "foreach" statement can be used in order to run through the elements in an array and check or use the individual values.
This makes it very easy, for example, to count states within an array or to perform operations with the values.

It is also possible to only iterate over part of an array.

→ Please note that it is not possible to assign values to the elements in an array while the array is being iterated over with "foreach."

Different operators are available within a condition. You can nest multiple control structure instructions if necessary.

only if and endif

```
if (tag = 10)
   tag := 100;
endif
```

if, else and endif

```
if (tag = 10)
   tag := 100;
else
   tag := 0;
endif
```

if, elseif, else and endif

```
if (tagA > tagB)
```

```
   tagC := 10;
elseif (tagA = tagB)
   tagC := 20;
else
   tagC := 0;
endif
```

elseif

```
if (x = 0)
   ...
elseif (x = 1)
   ...
elseif (x = 2)
   ...
elseif (x = 3)
   ...
or
   ...
Endif
```

foreach: With local tags and operator +=

```
Var errorElement: Piece;
Foreach (errorElement in MyErrorArray) if (errorElement = 1)
   countErrors += 1;Endif
       endforeach
```

foreach: Iterating over part of an array

```
Var errorElement: Piece;
Foreach (errorElement in MyErrorArray[0..2]) if (errorElement
= 1)
   countErrors += 1;Endif
endforeach
```

15.4.27 Inc

Increase tag value by 1

Parameter entered	Name of the Tags
Function available in script	Yes
Function available in application entry script	No

15.4.28 InvertBit

Invert the xth bit of a tag.

Parameter entered	Name of the Tags
Parameter entered	Bit number [0..x]
Function available in script	Yes
Function available in application entry script	No

15.4.29 OnChangeTag

Checks, whether a tag value was changed. Whenever the tag was changed the function is "true".

→ This loop script function from the Event function group can only be used as a condition for an if command.

→ This function will return "true" only ONCE for each tag change and accordingly should only be used at a single location in a script.

Parameter entered	Name of the Tags
Return parameter	True, if tag value was changed
Function available in script	Yes
Function available in application entry script	No

15.4.30 OnInterval_daily

This function can be used to have a script instruction sequence be executed at the defined weekdays at a specific time.

→ This loop script function from the Event function group can only be used as a condition for an if command.

→ If the device's date or time is changed manually, this function will return a value of TRUE once
if the time exceeds the configured time event.

Parameter entered	Weekday in quotation marks "...", multiple weekdays separated with a **:**
Parameter entered	Daily at this time in quotation marks "..."
Return parameter	True if the time on defined weekdays matches
Function available in script	Yes
Function available in application entry script	No

The weekdays are defined as follows:

Bit 0:	Sunday
Bit 1:	Monday
Bit 2:	Tuesday
Bit 3:	Wednesday
Bit 4:	Thursday
Bit 5:	Friday
Bit 6:	Saturday

Sample script:

Requirement: An acoustic signal should be emitted every Monday, Tuesday, and Friday at 7:00 AM.

▶ Add the following lines to the loop script:

```
if (Event.OnInterval_daily("Sun:Mon:Fri", "7:00:00"))
 System.Beep();
endif
```

15.4.31 OnInterval_dailyVar

This function enables a script command sequence to be executed at the defined weekdays at a specific time. The values of the selected tags define the time and weekday required.

→ This loop script function from the Event function group can only be used as a condition for an if command.

→ If the device's date or time is changed manually, this function will return a value of TRUE once
if the time exceeds the configured time event.

Parameter entered	Name of a tag: weekday
Parameter entered	Name of a tag: hours
Parameter entered	Name of a tag: minutes
Parameter entered	Name of a tag: seconds
Return parameter	True if the time on the defined weekdays matches
Function available in script	Yes
Function available in application entry script	No

The weekdays are defined as follows:

Bit 0:	Sunday
Bit 1:	Monday
Bit 2:	Tuesday
Bit 3:	Wednesday
Bit 4:	Thursday
Bit 5:	Friday
Bit 6:	Saturday

15.4.32 OnInterval_hourly

This function enables a script command sequence to be executed every x minutes after the hour.

→ This loop script function from the Event function group can only be used as a condition for an if command.

→ Please note:
The device's startup time will be used as a basis when calculating the time intervals. Time intervals <= 0 minutes are not permitted.
If the time interval is <= 0 minutes, the function will always be TRUE.

→ If the device's date or time is changed manually, this function will return a value of TRUE once
if the time exceeds the configured time event.

Parameter entered	n minutes after every full hour
Return parameter	True, if time interval elapsed
Function available in script	Yes
Function available in application entry script	No

15.4.33 OnInterval_hourlyVar

This function enables a script command sequence to be executed every x minutes after the hour. The tag value defines the minutes.

→ This loop script function from the Event function group can only be used as a condition for an if command.

→ Please note:
The device's startup time will be used as a basis when calculating the time intervals. Time intervals <= 0 minutes are not permitted.
If the time interval is <= 0 minutes, the function will always be TRUE.

→ If the device's date or time is changed manually, this function will return a value of TRUE once
if the time exceeds the configured time event.

Parameter entered	Name of a tag: n minutes after every full hour
Return parameter	True, if time interval elapsed
Function available in script	Yes
Function available in application entry script	No

15.4.34 OnInterval_min

This function enables a script command sequence to be executed every n minutes.

→ This loop script function from the Event function group can only be used as a condition for an if command.

→ Please note:
The device's startup time will be used as a basis when calculating the time intervals. Time intervals <= 0 minutes are not permitted.
If the time interval is <= 0 minutes, the function will always be TRUE.

If the device's date or time is changed manually, this function will return a value of TRUE once
if the time exceeds the configured time event.

Parameter entered	Time interval in number of minutes
Return parameter	True, if time interval elapsed
Function available in script	Yes
Function available in application entry script	No

Sample script:

```
if (Event.OnInterval_min(2))
 System.Beep();
endif
```

15.4.35 OnInterval_minVar

This function enables a script command sequence to be executed every n minutes. The tag value for the number of minutes is set as the time interval.

If the tag value is <20>, the interval will be 20 minutes. If 15 minutes in an interval since the script instructions were last executed have already passed and the tag value is changed to 13 minutes, the function condition will be met immediately. In other words, the script instructions will be executed and the interval will start over.

This loop script function from the Event function group can only be used as a condition for an if command.

Please note:
The device's startup time will be used as a basis when calculating the time intervals. Time intervals <= 0 minutes are not permitted.
If the time interval is <= 0 minutes, the function will always be TRUE.

If the device's date or time is changed manually, this function will return a value of TRUE once
if the time exceeds the configured time event.

Parameter entered	Name of a tag: interval in minutes
Return parameter	True, if time interval elapsed
Function available in script	Yes
Function available in application entry script	No

15.4.36 OnInterval_monthly

This function enables a script command sequence to be executed in the selected months at the defined day at the specified time.

This loop script function from the Event function group can only be used as a condition for an if command.

→ If the device's date or time is changed manually, this function will return a value of TRUE once
if the time exceeds the configured time event.

Parameter entered	Day of the month
Parameter entered	Time in quotation marks "..."
Parameter entered	Month in quotation marks "..." multiple months separated with a :
Return parameter	True if time matches
Function available in script	Yes
Function available in application entry script	No

15.4.37 OnInterval_monthlyVar

This function enables a script command sequence to be executed in the selected months at the defined day at the specified time. The values of the selected tags define the time, day and month required.

The months are defined as follows: Bit 0 ... Bit 11 = January ... December

→ This loop script function from the Event function group can only be used as a condition for an if command.

→ If the device's date or time is changed manually, this function will return a value of TRUE once
if the time exceeds the configured time event.

Parameter entered	Name of a tag: Day of the month
Parameter entered	Name of a tag: Hour
Parameter entered	Name of a tag: Minute
Parameter entered	Name of a tag: Second
Parameter entered	Name of a tag: Months
Return parameter	True if time matches
Function available in script	Yes
Function available in application entry script	No

15.4.38 OnInterval_sec

This function enables a script command sequence to be executed every n seconds.

→ This loop script function from the Event function group can only be used as a condition for an if command.

→ The device's start time will be used as a basis when calculating the interval.

→ If the device's date or time is changed manually, this function will return a value of TRUE once
if the time exceeds the configured time event.

Parameter entered	Time interval in number of seconds

Return parameter	True, if time interval elapsed (minimum value 5)
Function available in script	Yes
Function available in application entry script	No

Sample script:

```
if (Event.OnInterval_sec( 20 ) )
Math.Inc( xy )
endif
```

In the above example, tag "xy" is incremented every 20 seconds. For example: if the device is started at 13:23:07, the instruction sequence will be executed for the 1st time at 13:23:27 and then again every 20 seconds.

15.4.39 OnInterval_weekly

This function enables a script command sequence to be executed at the defined weekday at a specified time.

→ This loop script function from the Event function group can only be used as a condition for an if command.

→ If the device's date or time is changed manually, this function will return a value of TRUE once
if the time exceeds the configured time event.

Parameter entered	Weekday in quotation marks "..."
Parameter entered	Daily at this time in quotation marks "..."
Return parameter	True if the time on the defined weekday matches
Function available in script	Yes
Function available in application entry script	No

15.4.40 OnInterval_weeklyVar

This function enables a script command sequence to be executed at the defined weekday at the specified time (only a weekday should be selected, otherwise the OnInterval_dailyVar should be used at the specified time). The values of the selected tags define the time and weekday required.

→ This loop script function from the Event function group can only be used as a condition for an if command.

→ If the device's date or time is changed manually, this function will return a value of TRUE once
if the time exceeds the configured time event.

Parameter entered	Name of a tag: Weekday
Parameter entered	Name of a tag: Hour
Parameter entered	Name of a tag: Minute
Parameter entered	Name of a tag: Second

Return parameter	True if the time on the defined weekday matches
Function available in script	Yes
Function available in application entry script	No

The weekdays are defined as follows:

Value = 0:	Sunday
Value = 1:	Monday
Value = 2:	Tuesday
Value = 3:	Wednesday
Value = 4:	Thursday
Value = 5:	Friday
Value = 6:	Saturday

15.4.41 OnInterval_yearly

This function enables a script command sequence to be executed yearly in the selected month on the selected day at the specified time.

→ This loop script function from the Event function group can only be used as a condition for an if command.

→ If the device's date or time is changed manually, this function will return a value of TRUE once
if the time exceeds the configured time event.

Parameter entered	Month
Parameter entered	Day of the month
Parameter entered	Time in quotation marks "..."
Return parameter	True if time matches
Function available in script	Yes
Function available in application entry script	No

15.4.42 OnInterval_yearlyVar

This function enables a script command sequence to be executed yearly in the selected month on the specified day at the selected time. The values of the selected tags define the time, day and month required.

→ This loop script function from the Event function group can only be used as a condition for an if command.

→ If the device's date or time is changed manually, this function will return a value of TRUE once
if the time exceeds the configured time event.

Parameter entered	Name of a tag: Month
Parameter entered	Name of a tag: Day of the month

Parameter entered	Name of a tag: Hour
Parameter entered	Name of a tag: Minute
Parameter entered	Name of a tag: Second
Return parameter	True if time matches
Function available in script	Yes
Function available in application entry script	No

15.4.43 OnPresetOK

Checks whether data synchronization has been completed successfully when the device is restarted. This function returns the state of bit 1 of the 1st data word in the Status structure.

→ This loop script function from the Event function group can only be used as a condition for an if command.

→ This function will be run exactly ONCE after the MMI program is restarted.

Return parameter	True after successfully synchronization of all tags
Function available in script	Yes
Function available in application entry script	No

15.4.44 OnRecipeLoad

This function is "true" if a new recipe entry was loaded.

→ This loop script function from the Event function group can only be used as a condition for an if command.

→ This function will return "true" only ONCE for each recipe entry change.

Parameter entered	Name of recipe
Return parameter	True if a recipe entry was loaded.
Function available in script	Yes
Function available in application entry script	No

15.4.45 OnTime

Checks whether the device's internal date / time match the date / time being checked

→ This loop script function from the Event function group can only be used as a condition for an if command.

→ If the device's date or time is changed manually, this function will return a value of TRUE once
if the time exceeds the configured time event.

Parameter entered	Time in quotation marks "..."

Parameter entered	Date in quotation marks "..."
Return parameter	Once only true if time / date matches
Function available in script	Yes
Function available in application entry script	No

15.4.46 OnTimeVar

Checks whether the device's internal date / time match the date / time defined with the tags.

→ This loop script function from the Event function group can only be used as a condition for an if command.

→ If the device's date or time is changed manually, this function will return a value of TRUE once
if the time exceeds the configured time event.

Parameter entered	Name of a tag: Hour
Parameter entered	Name of a tag: Minute
Parameter entered	Name of a tag: Second
Parameter entered	Name of a tag: Day
Parameter entered	Name of a tag: Month
Parameter entered	Name of a tag: Year
Return parameter	Once only true if time / date matches
Function available in script	Yes
Function available in application entry script	No

15.4.47 ResetBit

Reset the xth bit of the tag (set to 0).

Parameter entered	Name of a tag
Parameter entered	Bit number [0...x]
Function available in script	Yes
Function available in application entry script	No

15.4.48 Round

Generates an integer from the floating-point number. If the value of the first decimal place is between 5 and 9, the number will be rounded up; otherwise, it will be rounded down.

Parameter entered	Name of a tag
Function available in script	Yes
Function available in application entry script	No

15.4.49 Save History of previous day to CSV

This function saves the error messages of the last day in a .csv file

Number of the Function	216 (D8h)
Executable from the PLC	No
Function with ON/OFF state	No
Function available in standard screens	Yes
Function available with keyboards	Yes
Function available in script	Yes
Function available in application entry script	Yes

15.4.50 Save History of previous day to CSV Var

This function saves the error messages of the last day in a .csv file

The path is passed as a tag.

Number of the Function	217 (D9h)
Executable from the PLC	No
Function with ON/OFF state	No
Function available in standard screens	Yes
Function available with keyboards	Yes
Function available in script	Yes
Function available in application entry script	Yes

15.4.51 Save History to CSV

This function saves the error messages in a .csv file.

Number of the Function	214 (D6h)
Executable from the PLC	No
Function with ON/OFF state	No
Function available in standard screens	Yes
Function available with keyboards	Yes
Function available in script	Yes
Function available in application entry script	Yes

15.4.52 Save History to CSV Var

This function saves the error messages in a .csv file.
The path is passed as a tag.

Number of the Function	215 (D7h)
Executable from the PLC	No
Function with ON/OFF state	No
Function available in standard screens	Yes
Function available with keyboards	Yes
Function available in script	Yes
Function available in application entry script	Yes

15.4.53 SetBit

Set the xth bit of the tag (set to 1).

Parameter entered	Name of a tag
Parameter entered	Bit number [0..x]
Function available in script	Yes
Function available in application entry script	No

15.4.54 SetErrTextConst

Writes an error text to a string tag. The arguments that need to be passed are the string tag and the project text ID. This makes it possible to write language-dependent texts to a tag.

Parameter entered	Name of a string tag
Parameter entered	Error Text ID
Function available in script	Yes
Function available in application entry script	No

15.4.55 SetErrTextVar

Writes an error text to a string tag. The string tag and the tag for the Error Text ID are specified as parameters. This makes it possible to write language-dependent texts to a tag.

Parameter entered	Name of a string tag
Parameter entered	Tag with Error Text ID
Function available in script	Yes
Function available in application entry script	No

15.4.56 SetHlpTextConst

Writes a help text to a string tag. The arguments that need to be passed are the string tag and the project text ID. This makes it possible to write language-dependent texts to a tag.

Parameter entered	Name of a string tag
Parameter entered	Help Text ID
Function available in script	Yes
Function available in application entry script	No

15.4.57 SetHlpTextVar

Writes a help text to a string tag. The arguments that need to be passed are the string tag and the tag for the help text ID. This makes it possible to write language-dependent texts to a tag.

Parameter entered	Name of a string tag
Parameter entered	Tag with Help Text ID
Function available in script	Yes
Function available in application entry script	No

15.4.58 SetTextConst

Writes a standard text to a string tag. The arguments that need to be passed are the string tag and the project text ID. This makes it possible to write language-dependent texts to a tag.

Parameter entered	Name of a string tag
Parameter entered	Standard Text ID
Function available in script	Yes
Function available in application entry script	No

15.4.59 SetTextDate

Writes the current date as a text (e.g., "29-12-2014") to a string tag. The argument that needs to be passed is the string tag.

Parameter entered	Name of a string tag; defines a data point on the device or PLC. The assigned addresses link the PLC's and MMI's data points.
Function available in script	Yes
Function available in application entry script	No

15.4.60 SetTextDateTime

Writes the current date and time as a text (e.g., "29-12-2014 16-02-32") to a string tag. The argument that needs to be passed is the string tag.

Parameter entered	Name of a string tag
Function available in script	Yes
Function available in application entry script	No

15.4.61 SetTextTime

Writes the current time as a text (e.g., "16-02-32") to a string tag. The argument that needs to be passed is the string tag.

Parameter entered	Name of a string tag
Function available in script	Yes
Function available in application entry script	No

15.4.62 SetTextVar

Assigns a Standard Text to a string tag. The string tag and the tag for the Project Text ID are specified as parameters. This makes it possible to write language-dependent texts to a tag.

Parameter entered	Name of a string tag
Parameter entered	Tag with Standard Text ID
Function available in script	Yes
Function available in application entry script	No

15.4.63 SHL

Shifts the bit sequence in a tag n bits to the left. The bits to the left of the shifted sequence will always have a value of 0.

Parameter entered	Name of a tag
Parameter entered	Number of bits to be shifted
Function available in script	Yes
Function available in application entry script	No

SHL example

```
SHL(tag, 3)
```

1001 1101 Tag value before shift

1110 1000 Tag value after shift

15.4.64 SHR

Shifts the bit sequence in a tag n bits to the right. The bits to the left of the shifted sequence will always have a value of 0.

Parameter entered	Name of a tag
Parameter entered	Number of bits to be shifted
Function available in script	Yes
Function available in application entry script	No

SHR example

```
SHR(tag, 3)
```

1001 1101 Tag value before shift

0001 0011 Tag value after shift

15.4.65 SwitchLeft

Shifts a bit one position to the left. Only one of the bits should have a value of 1. If multiple bits have a value of 1, the least significant bit with a value of 1 will be shifted to the left and all the other bits will be set to 0. If the bit is shifted beyond the number of positions available, the result will be 0.

→ Only one of the bits should have a value of 1.

Parameter entered	Name of a tag
Parameter entered	Number of steps
Function available in script	Yes
Function available in application entry script	No

Example

SwitchLeft (tag, 4)

0000 **0001** Tag value before shift

0000 **0010** Tag value after 1st shift

0000 **0100** Tag value after 2nd shift

0000 **1000** Tag value after 3rd shift

0000 **0000** Tag value after 4th shift

SwitchLeft (tag, 8)

0000 1000 Tag value before shift

0001 0000 Tag value after 1st shift

0010 0000 Tag value after 2nd shift

0100 0000 Tag value after 3st shift

1000 0000 Tag value after 4st shift

0000 0000 Tag value after 5st shift

15.4.66 SwitchLeft endless

Shifts a bit one position to the left. Only one of the bits should have a value of 1. If multiple bits have a value of 1, the least significant bit with a value of 1 will be shifted to the left and all the other bits will be set to 0. If the bit is shifted beyond the number of positions available, it will be moved to the least significant bit.

Only one of the bits should have a value of 1.

Parameter entered	Name of a tag
Parameter entered	max. Number of steps
Function available in script	Yes
Function available in application entry script	No

Switch endless example

`SwitchLeft [endless](tag, 4)`

0000 **0001** Tag value before shift

0000 **0010** Tag value after 1st shift

0000 **0100** Tag value after 2nd shift

0000 **1000** Tag value after 3rd shift

0000 **0001** Tag value after 4th shift

`SwitchLeft [endless](tag, 8)`

0000 1000 Tag value before shift

0001 0000 Tag value after 1st shift

0010 0000 Tag value after 2nd shift

0100 0000 Tag value after 3st shift

1000 0000 Tag value after 4st shift

0000 0001 Tag value after 5st shift

15.4.67 SwitchRight

Shifts a bit one position to the right. Only one of the bits should have a value of 1. If multiple bits have a value of 1, the least significant bit with a value of 1 will be shifted to the right and all the other bits will be set to 0. If the bit is shifted beyond the least significant bit, the result will be 0.

→ Only one of the bits should have a value of 1.

Parameter entered	Name of a tag
Parameter entered	Number of steps
Function available in script	Yes
Function available in application entry script	No

Switch Right example

```
SwitchRight (tag, 4)
```

0000 **1000** Tag value before shift

0000 **0100** Tag value after 1st shift

0000 **0010** Tag value after 2st shift

0000 **0001** Tag value after 3rd shift

0000 **0000** Tag value after 4th shift

```
SwitchRight (tag, 8)
```

0010 0000 Tag value before shift

0001 0000 Tag value after 1st shift

0000 1000 Tag value after 2nd shift

0000 0100 Tag value after 3st shift

0000 0010 Tag value after 4nd shift

15.4.68 SwitchRight endless

Shifts a bit one position to the right. Only one of the bits should have a value of 1. If multiple bits have a value of 1, the least significant bit with a value of 1 will be shifted to the right and all the other bits will be set to 0. If the bit is shifted beyond the least significant position, it will be moved to the position defined with the number of positions.

→ Only one of the bits should have a value of 1.

Parameter entered	Name of a tag
Parameter entered	Number of steps
Function available in script	Yes
Function available in application entry script	No

SwitchRight endless example

```
SwitchRight [endless](tag, 4)
```

0000 **1000** Tag value before shift

0000 **0100** Tag value after 1st shift

0000 **0010** Tag value after 2st shift

0000 **0001** Tag value after 3rd shift

0000 **1000** Tag value after 4rd shift

```
SwitchRight [endless](tag, 8)
```

0000 1000 Tag value before shift

0000 0100 Tag value after 1st shift

0000 0010 Tag value after 2st shift

0000 0001 Tag value after 3st shift

1000 0000 Tag value after 4st shift

15.4.69 Trunc

Truncates the decimal places in a floating-point number, generating an integer as a result.

Parameter entered	Name of a tag
Function available in script	Yes
Function available in application entry script	No

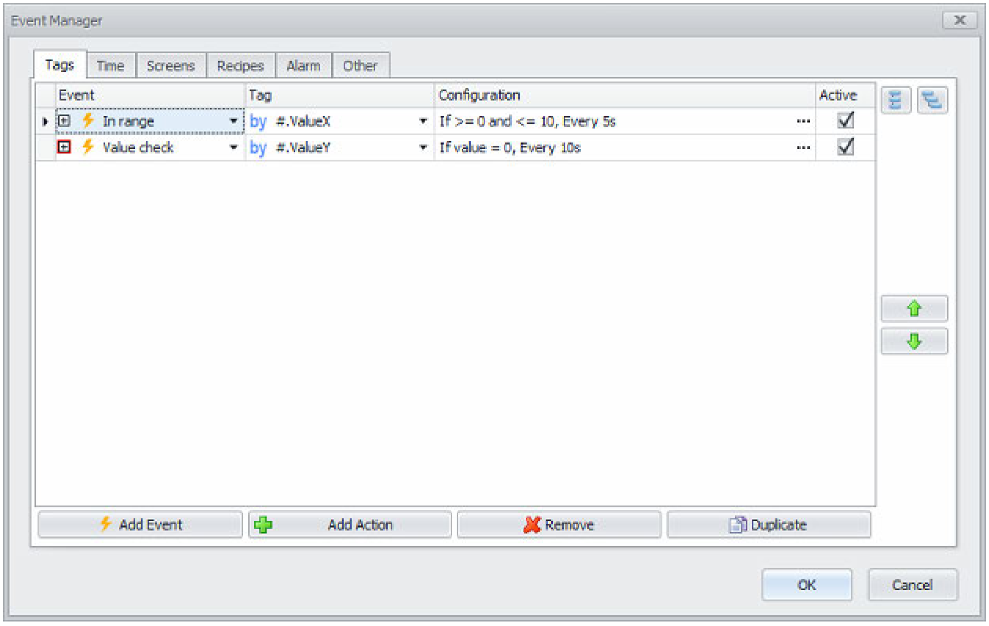
Script example

```
WordVar:=Math.Trunc(FloatVar);
```

16. Event Manager

The Event Manager can be used to configure project events and trigger specific actions at this time. An event can have multiple actions assigned.

Home tab\ Event manager



Event	Tag	Configuration	Active
In range	by #.ValueX	If >= 0 and <= 10, Every 5s	☑
Value check	by #.ValueY	If value = 0, Every 10s	☑

Figure 593: Event Manager dialog box

The following basic functionalities are available for all tabs:

Add Event	Adds a new event configuration
Add	Adds a new action to the selected event
Remove	Removes the selected event/action
Duplicate	Creates a copy of the selected event.
⬇ ⬆	GALILEO will run the events in the order defined here. This means that if conditions for multiple events are met at the same time, the topmost event (or rather the assigned script) will be run first. You can accordingly use the arrows to change the prioritization if necessary.
	Expand/Collapse all events and their settings

The configuration of all the different event types have following in common:

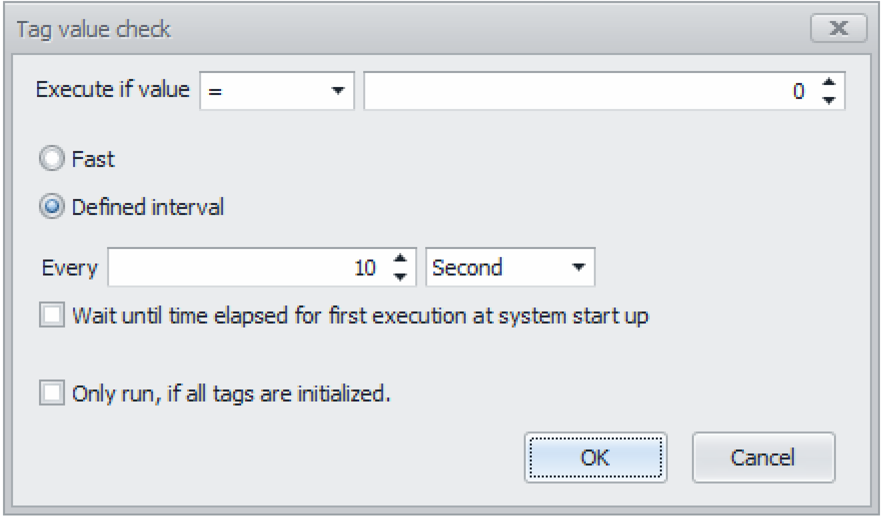
Event	Selection of the specific event type
Actions	For every event, multiple actions can be assigned/configured (e.g.,

	a script to execute)
Active	If the checkbox is enabled, the system will check for the event at runtime. If it is disabled, the event will be ignored. This can be useful during development or for test purposes, for example.

Some event types (like Tag events) support an additional configuration. The cell of such a Configuration-column will show a textual representation of the configuration. Clicking on the ...-icon in the Configuration cell will open a new dialog box with the settings corresponding to the event.

There are certain options shared by all events. These options are explained here.

Home tab\ \Configuration...



Tag value check

Execute if value = 0

○ Fast

● Defined interval

Every 10 Second

□ Wait until time elapsed for first execution at system start up

□ Only run, if all tags are initialized.

Figure 594: Event Manager Configuration ... dialog box

Interval settings

In certain circumstances, this setting will not be available for every event type.

The time interval at which the system should check for the event. For example, when checking for a specific tag value, you may want to run the corresponding script only every five seconds.

- Fast: Runs the check as fast as possible for the GALILEO Runtime System (but not faster than at a 100 ms interval).
- Defined interval: This time can be configured using seconds or minutes as the interval unit. There is also an option in the context menu to specify a tag instead of a fixed value. In this case, the value in the tag will specify the interval.

Wait with first execution at system start until the time interval is reached: This option plays a role when the GALILEO Runtime System starts. If the option is disabled and the event condition is already met when starting, the script will be run immediately. If the option is enabled, the system will wait for the configured time interval to elapse before the first check for the event condition.

> Example of a case where it is a good idea
> to disable this option: Activity logging
> to enable this option: Backing up graph archives .

Run only until all tags have been initialized

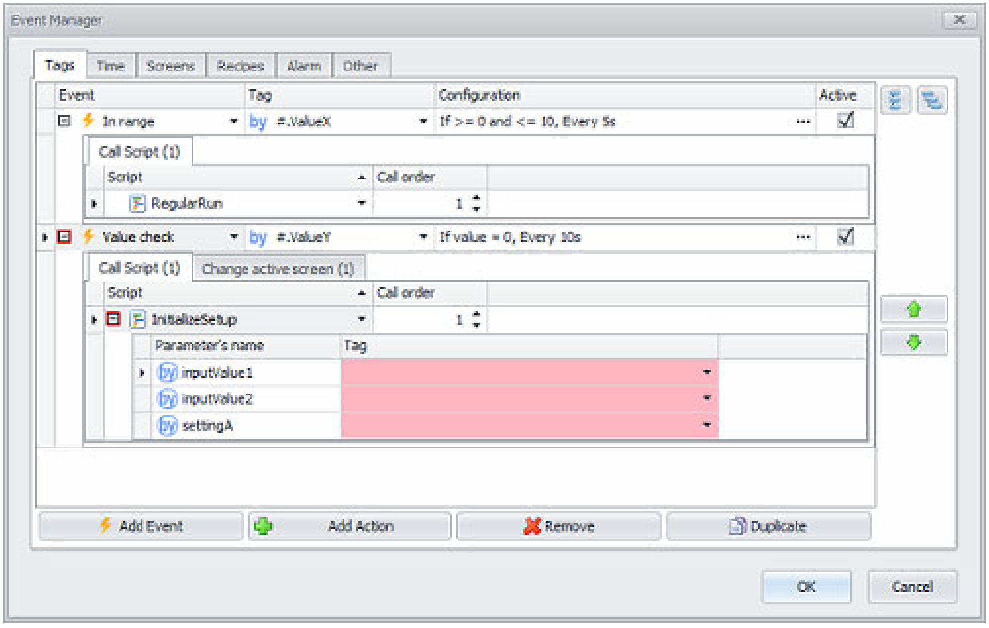
When communicating with a PLC, it is possible for the condition for an event to be met before the PLC gets the chance to retrieve all the initial tag values. Since this can result in problems and undesired effects in certain circumstances, this option can be enabled to prevent this. If a repeat time interval (depending on the event type) has been configured, the next check for the event will not be carried out before the time interval is reached.

If the option is disabled, the script will be run when the event occurs regardless.

Actions

An event can have multiple actions assigned. When adding actions to an event, the expanded view could look like this:

Home tab\ Event manager



Event	Tag	Configuration	Active
In range	#.ValueX	If >= 0 and <= 10, Every 5s	✓
Value check	#.ValueY	If value = 0, Every 10s	✓

In range – Call Script (1)

Script	Call order
RegularRun	1

Value check – Call Script (1), Change active screen (1)

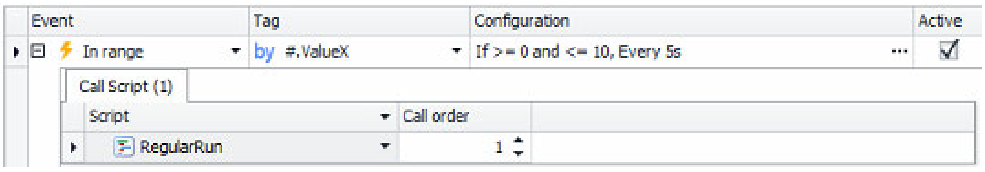
Script	Call order
InitializeSetup	1

Parameter's name	Tag
inputValue1	
inputValue2	
settingA	

Figure 595: Pane Event Manager with Event and Action

Under every event, the different available action types (e.g., Call Script, Change active screen) are shown in separate tabs.

When assigning a Script action, the script to execute can be selected. If the script has no parameters, the configuration is already completed like this:



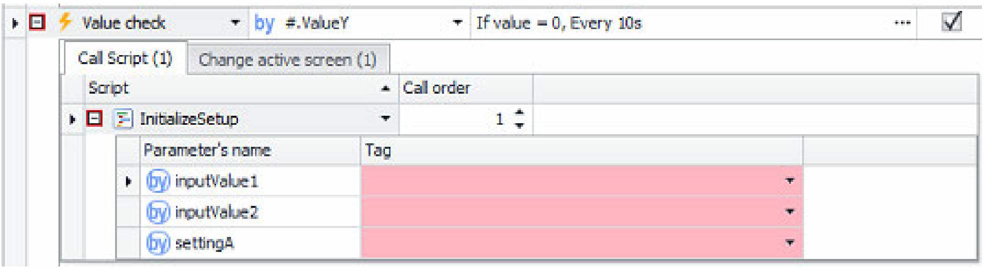
Event	Tag	Configuration	Active
In range	by #.ValueX	If >= 0 and <= 10, Every 5s	☑

Call Script (1)

Script	Call order
RegularRun	1

This configuration means: If the #.ValueX is >= 0 and <= 10, the script RegularRun will be executed. The check will be done every 5 seconds.

If the assigned script takes additional input parameters, the action can be expanded to configure the required input parameters like this:



Value check	by #.ValueY	If value = 0, Every 10s	☑

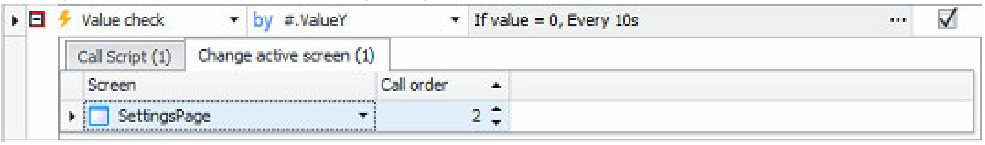
Call Script (1) | Change active screen (1)

Script	Call order
InitializeSetup	1

Parameter's name	Tag
inputValue1	
inputValue2	
settingA	

A red colored cell as well as read framed expand/collapse symbols (⊞⊟) means: The configuration is faulty or incomplete.

For every action also a specific Call order can be assigned. This call order is similar to the order of the events itself, mentioned above. But here it is possible to control the execution order of actions within an event. This is required in this form, because the actions can be of different types (e.g., Call Script, Change active screen) and the configuration is on different tabs. Here, for example, it's visible that the Screen change action to screen SettingsPage has the Call order = 2:



Value check	by #.ValueY	If value = 0, Every 10s	☑

Call Script (1) | Change active screen (1)

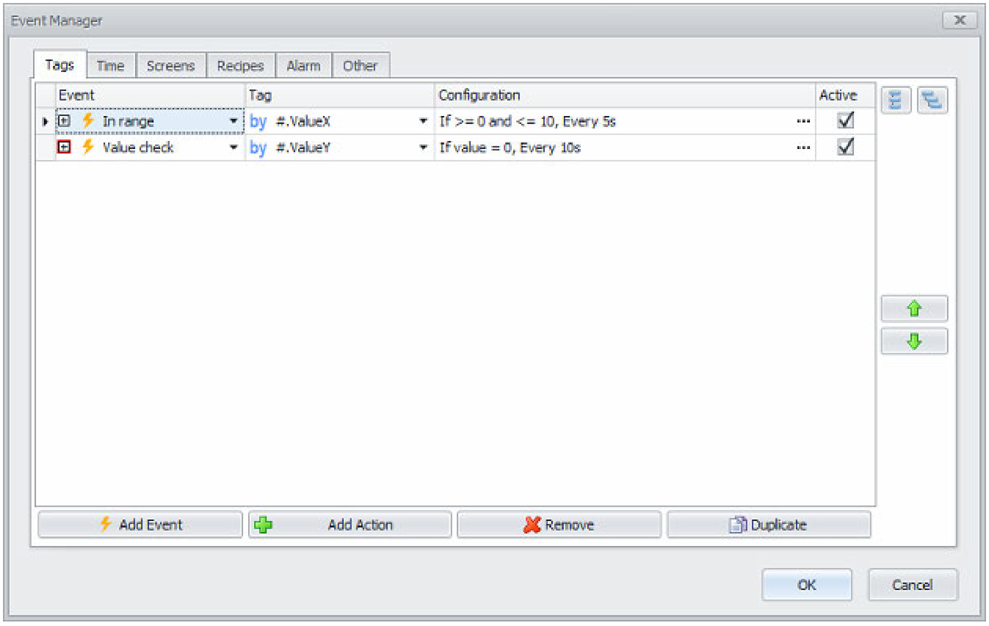
Screen	Call order
SettingsPage	2

Togehter with the screenshot before, the following event is configured: If the #.ValueY is = 0, the script InitializeSetup will be executed first and afterwards a screen change to SettingsPage is executed. The check will be done every 10 seconds.

16.1 Tags events

Tag events make it possible to check tag values and trigger actions if the corresponding condition is met.

Home tab\ Event manager



Event	Tag	Configuration	Active
In range	by #.ValueX	If >= 0 and <= 10, Every 5s	☑
Value check	by #.ValueY	If value = 0, Every 10s	☑

Figure 596: Event Manager dialog box, Tags, tab

The following event types are supported:

Value change

The script will run if there is a change in the tag values. The configuration is restricted to the generally available options.

Value check

If the tag value meets the configured condition, the script will be run.
The following value comparison operators are available:

=, ≠, >, >=, <, <=

You can either specify a fixed numeric value or another tag (with the context menu) as a comparison value.

In Range

If the tag value falls within the configured range, the script will be run.

You can specify either a fixed numeric value or another tag (with the context menu) as upper and lower limits.

You can also use the configured limits for the event tag.
To do this, enable the "Apply event tag limits" option.

Outside range

If the tag value falls outside the configured range, the script will be run.

You can specify either a fixed numeric value or another tag (with the context menu) as upper and lower limits.

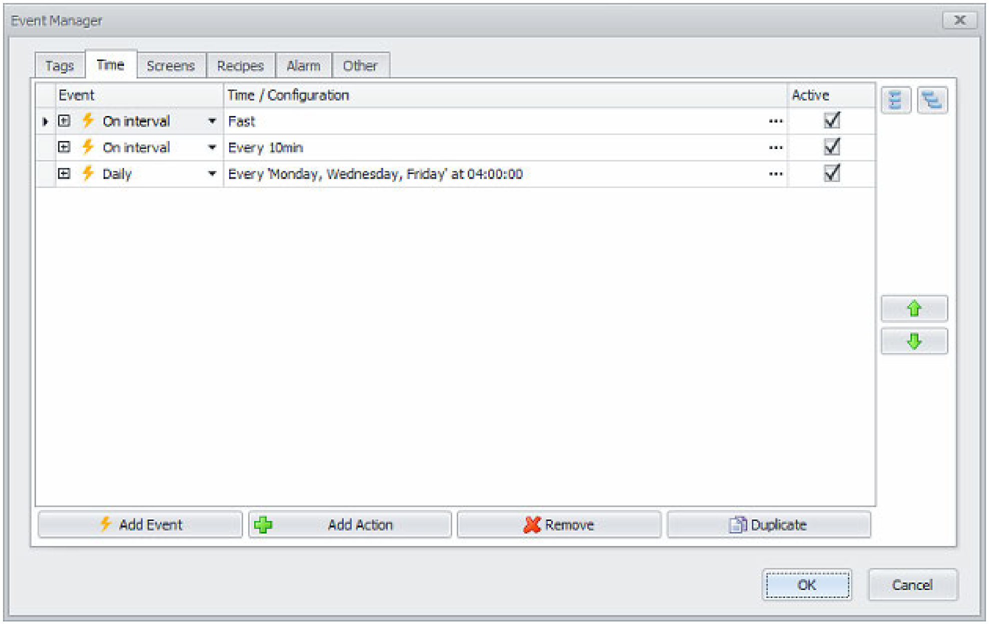
You can also use the configured limits for the event tag.
To do this, enable the "Apply event tag limits" option.

In the Tag column, the tag relevant to the event check is selected.

16.2 Time events

Time events can be used to trigger actions at a specific time or at a defined time interval.

Home tab\ Event manager



Event	Time / Configuration	Active
On interval	Fast	☑
On interval	Every 10min	☑
Daily	Every 'Monday, Wednesday, Friday' at 04:00:00	☑

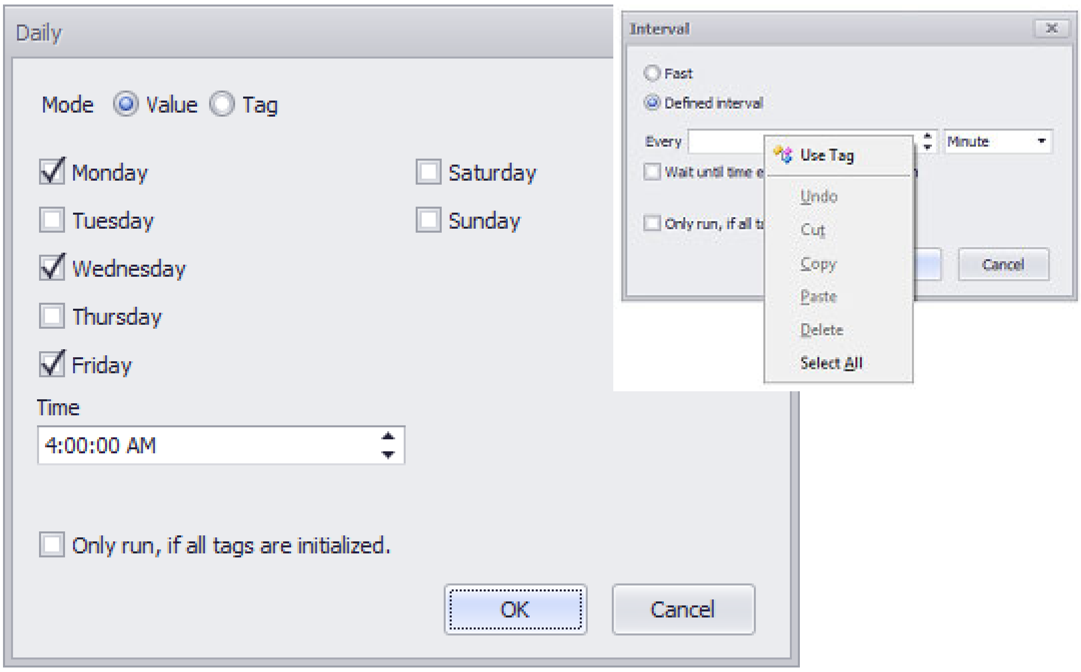
Figure 597: Event Manager dialog box, Time tab

The following event types are supported:

- Interval
- Time point
- Hourly
- Daily
- Weekly
- Monthly
- Yearly

→ The scripts that were referred to as "loop scripts" up to GALILEO 10.4.x can now be implemented as follows:
"Interval" event type, "Fast" configuration.

For all event types, you can switch between fixed values or tags in the configuration (either with the context menu or with the radio buttons)



Daily

Mode ● Value ○ Tag

☑ Monday ☐ Saturday
☐ Tuesday ☐ Sunday
☑ Wednesday
☐ Thursday
☑ Friday

Time
4:00:00 AM

☐ Only run, if all tags are initialized.

Interval

○ Fast
● Defined interval

Every Minute

☐ Wait until time e

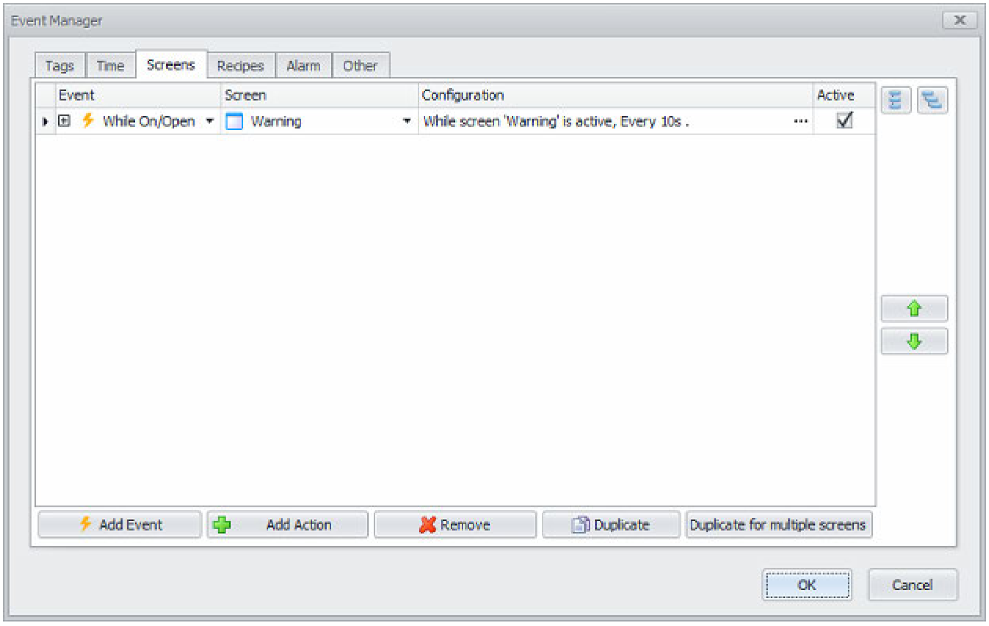
☐ Only run, if all t

Figure 598: Event Manager dialog box, Time tab, "..." button

16.3 Screen events

Screen events can be used to trigger actions if certain screen conditions are met.

Home tab\ Event manager



Event	Screen	Configuration	Active
While On/Open	Warning	While screen 'Warning' is active, Every 10s .	☑

Figure 599: Event Manager dialog box, Screens tab

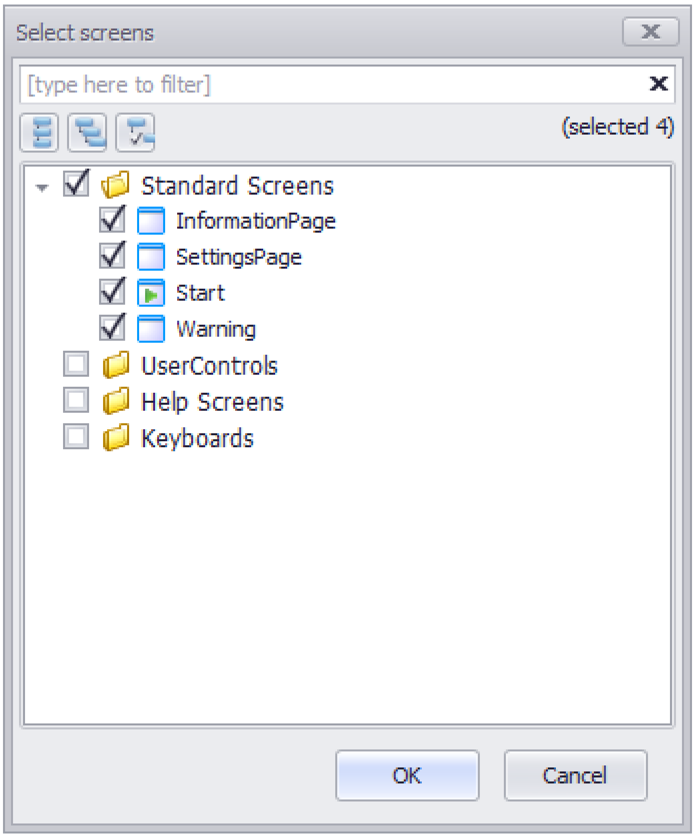
The only event type currently available here is: As long as is open.
The script will be run, based on the interval configuration, as long as the defined screen is open.

Please note that there can be multiple open screens at runtime under certain circumstances.
The following scenario illustrates this:

- Say there is a screen X for which an "As long as is open" event has been configured with script A
- And there is a dialog box Y for which an "As long as is open" event has been configured with script B
- If screen X is opened at runtime, script A will run continuously.
- If dialog box Y is opened while running screen X, script B will be run as well. As long as dialog box Y is open, this means that both scripts (A and B) will be run.

Duplication for multiple screens

The "Duplication for multiple screens" function makes it possible to conveniently duplicate an existing configured event for as many additional screens as you want. When you click on the ... button, the following dialog box will appear and you can select the screens:



(selected 4)

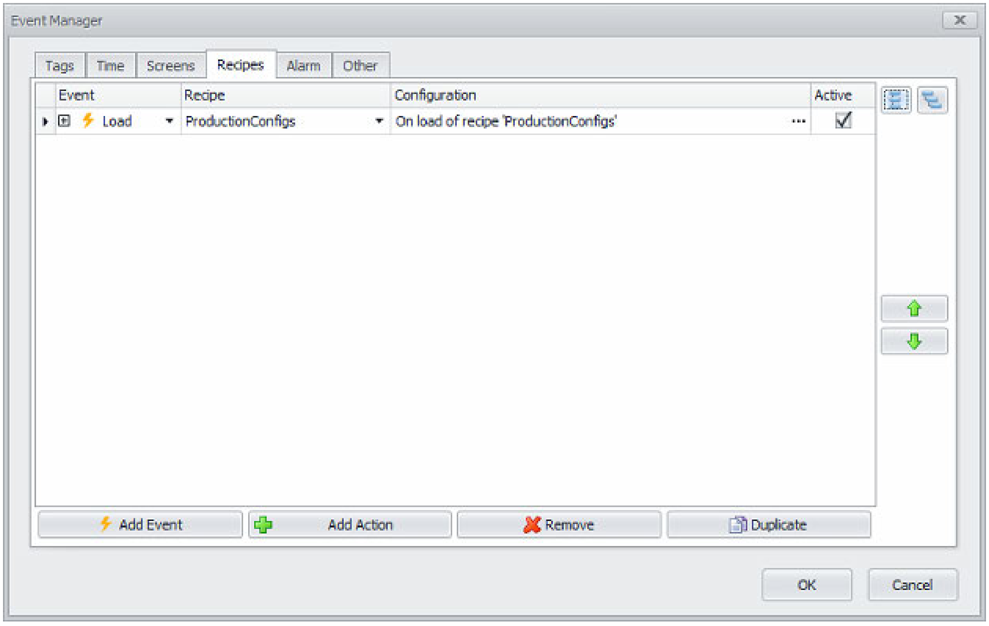
- ☑ Standard Screens
  - ☑ InformationPage
  - ☑ SettingsPage
  - ☑ Start
  - ☑ Warning
- □ UserControls
- □ Help Screens
- □ Keyboards

Figure 600: Event Manager dialog box, Screens tab, ... button

16.4 Recipe events

Recipe events can be used to trigger actions if certain recipe related conditions are met.

Home tab\ Event manager



Event	Recipe	Configuration	Active
Load	ProductionConfigs	On load of recipe 'ProductionConfigs'	☑

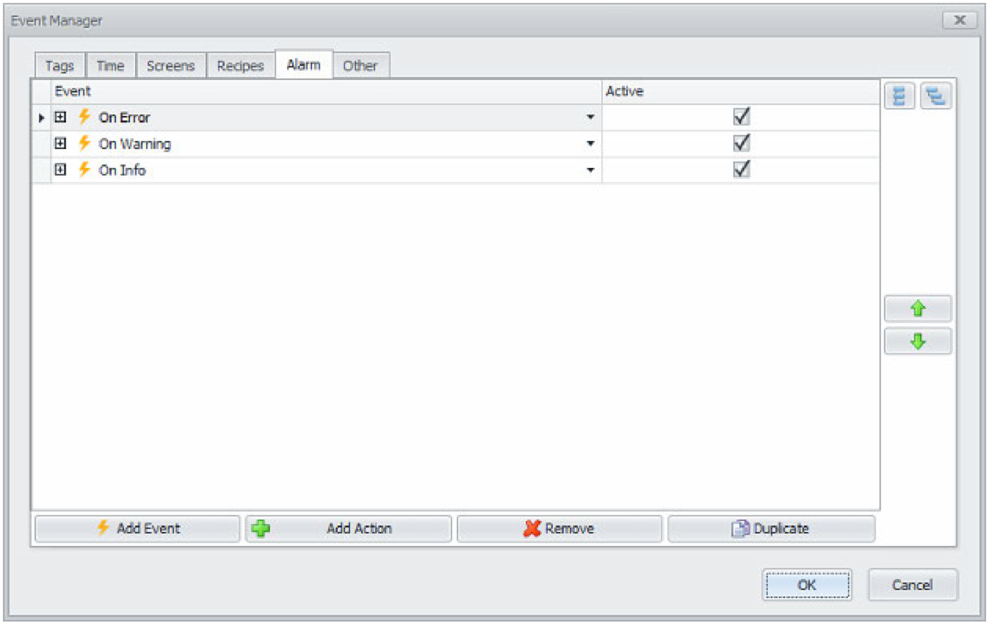
Figure 601: Event Manager dialog box, Recipes tab

The following is the only event type that is currently available: Load

16.5 Alarm events

Alarm events can be used to trigger actions, if a new alarm is noticed.

Home tab\ Event manager



Event Manager

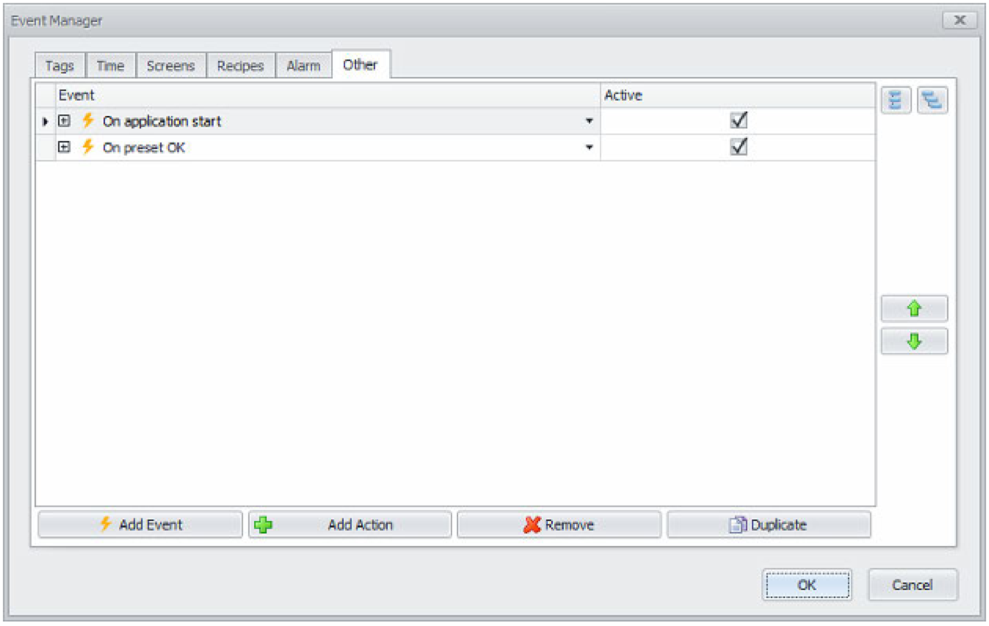
Event	Active
On Error	☑
On Warning	☑
On Info	☑

Figure 602: Event Manager dialog box, Alarm tab

16.6 Other events

The tab page Other allows configuration of events, which don't fit into one of the other categories.

Home tab\ Event manager



Event	Active
On application start	☑
On preset OK	☑

Figure 603: Event Manager dialog box, Others tab

On application startup: Script that should run when the GALILEO application starts

on Preset OK: Script that should be run after all tag values are synchronized with the PLC.

16.7 Restrictions – Script function limitations

16.7.1 Restrictions for application entry script

Functions that are not allowed to be used in an application entry script:

- System.→ "Restart Runtime", page 1116
- System.→ "Boot System", page 1110

16.7.2 Restriction when using script functions with short time intervals

Running certain functions frequently at very short time intervals can result in problems such as the GALILEO apparently freezing due to memory-intensive/computing-intensive tasks having to be completed in the background.

If these script functions are used for events with short time intervals, the GALILEO compiler will run a check and output a corresponding error message or warning.

The following applies to the following functions:

- If these functions are used for an event with a regular "Fast" execution interval, a compiler error will be output.
  The cyclical, quickly recurring function execution would result in massive memory and/or performance drops, which is why it needs to be avoided.
- If these functions are used to an event with a regular execution interval < 60 sec., a compiler warning will be output.
  If the application is well thought out, the execution will not result in any noticeable drops in performance; however, the use of the functions should be checked again (e.g., the minute-by-minute storage of large data volumes on a slow storage device can still result in problems).
  The compiler warning can be disabled in the Build and Deploy settings.
- If these functions are used for an event with an execution interval >= 60 sec.
  or the event is an event that is run one time only (e.g., when loading a recipe), a compiler message will not be output.

16.7.2.1 List of these functions in the individual groups

Alarm

→ section "Error Autoscroll On/Off", page 883
→ section "Error Priority ON/OFF", page 887
→ section "Save Error History as File", page 888
→ section "Save Error History as File Var", page 888
→ section "Save System History as File", page 888
→ section "Save System History as File Var", page 889
→ section "Save History to CSV", page 994

→ section "Save History to CSV Var", page 994
→ section "Save History of previous day to CSV", page 993
→ section "Save History of previous day to CSV Var", page 994

Communication

→ section "Error Autoscroll On/Off", page 883
→ section "Quiet on Connection x ON/OFF", page 626
→ section "CAN PDO ON/OFF", page 629
→ section "CAN PDO Persistent ON/OFF", page 629
→ section "CAN SDO Node x Persistent ON/OFF", page 628

Display

→ section "Backlight: Learn Dim Value", page 1109

Graph Blocks

→ section "Delete Archive File", page 722
→ section "Load Archive File Var", page 725
→ section "Load Archive File Full Path As Var", page 724
→ section "Save Archive File As Var", page 727
→ section "Save Archive File Full Path As Var", page 727
→ section "View Archive File Var", page 733
→ section "View Archive File Full Path As Var", page 734

HTML viewer - all functions

→ section "Backlight: Learn Dim Value", page 1109
→ section "Load URL Var", page 437

Image - all functions

→ section "Load Image", page 443
→ section "Load Image Var", page 443
→ section "Select Image ", page 444
→ section "Select Image Var", page 445

ListBox - all functions

→ section "List box: First Line", page 404
→ section "List box: Last Line", page 405
→ section "List box: Line down", page 405
→ section "List box: Line up", page 406
→ section "List box: Page down", page 406
→ section "List box: Page up", page 407
→ section "List box: Select", page 407
→ section "List box: SelectVar", page 408

Parameter list - all functions including printing

→ section "List box: First Line", page 404
→ section "List box: Last Line", page 405
→ section "List box: Line down", page 405
→ section "List box: Line up", page 406
→ section "List box: Page down", page 406
→ section "List box: Page up", page 407
→ section "List box: Select", page 407
→ section "List box: SelectVar", page 408

→ section "Change to Printer", page 1033
→ section "Print History", page 1033
→ section "Print Parameter List", page 1033
→ section "Print Parameter List (no Header)", page 1034
→ section "Print Report", page 1034
→ section "Print Screen", page 1034

Recipe

→ "First Line (Recipe)", page 772
→ section "Last Line", page 773
→ section "Line Down", page 773
→ section "Line Up", page 773
→ section "Page Down", page 776
→ section "Page Up", page 776

→ section "Change Recipe Entry Name", page 766
→ section "Change Recipe Entry Name userdef.", page 766

→ section "Clear Recipe Entry", page 767

→ section "Copy Recipe Entry", page 768
→ section "Copy to next", page 767

→ section "Display: Copy Entry", page 768
→ section "Display: Loaded Entry", page 769
→ section "Display: New Entry", page 769
→ section "Display: Selected Entry", page 769

→ section "Find Recipe Entry", page 770
→ section "Find Recipe Entry (start)", page 771
→ section "Find Recipe Entry userdef.", page 771
→ section "Find Rcp Entry userdef. (start) ", page 771
→ section "Find Next Recipe Entry", page 772
→ section "Find Next Recipe Entry (start)", page 770

→ section "LoadSelectedRecipeEntry", page 775

→ section "Save Recipe Entry", page 782
→ section "Save Recipe Entry As", page 783
→ section "Save Recipe Entry As userdef.", page 781
→ section "Save Recipe Entry Var", page 784
→ section "Save Recipe Entry As Var", page 783
→ section "Save As/Load Recipe Entry", page 782
→ section "Save As/Load Recipe Entry userdef.", page 779
→ section "SaveActualRecipeEntry", page 782
→ section "Save As to next", page 779
→ section "Save As to next userdef", page 780
→ section "Save As to next Var", page 780

→ section "Paste Recipe Entry", page 777
→ section "Protect Recipe Entry", page 777

→ section "New to next", page 775
→ section "New to next userdef.", page 776
→ section "Restore Default Values", page 778
→ section "Select Loaded Recipe Entry", page 784
→ section "Refresh Labels", page 778

Screen

→ section "CloseAllDialogs", page 377
→ section "CloseDialog", page 378
→ section "CloseMenu", page 1114
→ section "ShowMenu", page 1124
→ "Save Screen as Bitmap Var", page 1117

System

→ "ReReadTag", page 1116
→ "SyncWriteTag", page 1126
→ section "Call DLL Function", page 1113
→ section "Execute", page 1114
→ section "Execute (dynamic)", page 1114
Select Numeric Tag→ "Select ASCII Tag", page 1118
→ "Select Numeric Tag", page 1119

User Logger - all functions

→ "Add String Tag Log", page 950
→ "Add Text Log", page 950
→ "Add Value Tag Log", page 950
→ "Logger ON-OFF", page 951
→ "Flush Logger", page 950

→ "View Current Log File", page 951

User Management - all functions

→ "Change Group", page 902
→ "Change Group -userdef.", page 903
→ "Change PW (User x)", page 903
→ "Change PW (User x) -userdef.", page 904
→ "Change PW (other User) -userdef.", page 904
→ "Change PW (Own) -userdef.", page 904
→ "Change PW (other User)/ ChangePWOther", page 905
→ "Change PW (own)", page 906
→ "Delete User -userdef.", page 907
→ "Delete User", page 908
→ "Lock/Unlock User", page 909
→ "Lock/Unlock User -userdef.", page 910
→ "Login User", page 910
→ "Login User -userdef.", page 911
→ "Login User x", page 911
→ "Login User x -userdef.", page 912
→ "Logout User", page 912
→ "New User -userdef.", page 912
→ "New User", page 913
→ "RefreshAutoLogoutTime", page 914
→ "Select Group", page 914
→ "Select Group -userdef.", page 914
→ "Select User", page 915
→ "Select User -userdef.", page 915
→ "Show_all_Groups", page 915
→ "Show all Groups -userdef.", page 916
→ "Show all Users of Group", page 916
→ "Show all Users of Group -userdef.", page 916
→ "Show All Users -userdef.", page 917
→ "Show Groups of User", page 917
→ "Show Groups of User -userdef.", page 917
→ "Show all Users", page 917
→ "Unlock User", page 918
→ "Unlock User -userdef.", page 919
→ "Unlock user x", page 919
→ "UserLogin", page 919

Video player- all functions

→ "Video:Load", page 502
→ "Video:LoadVar", page 503
→ "Video:Pause", page 503
→ "Video:Select", page 504
→ "Video:SelectVar", page 504
→ "Video:Start", page 505
→ "Video:Stop", page 505

Web - all functions

→ "DisconnectAllClients", page 627

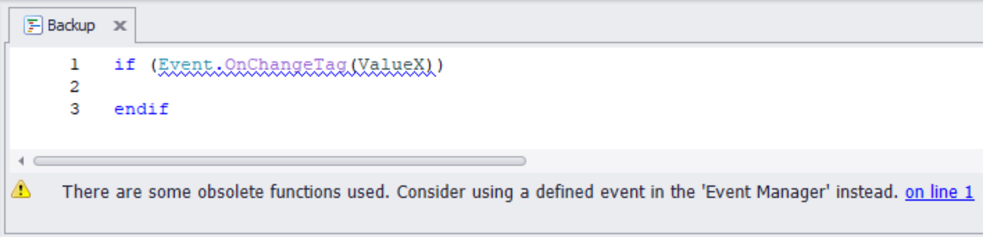
WebCam- all functions

→ section "WebCam:Capture Image", page 519
→ section "WebCam:Capture Image As", page 520
→ section "WebCam:Capture Image Var", page 520
→ section "WebCam:Capture Image Var As", page 521
→ section "WebCam:Start", page 522
→ section "WebCam:Stop", page 522

16.7.2.2 Deprecated functions

Script functions in GALILEO versions lower than 10.5 are still supported currently due to compatibility reasons.
However, it is recommended to replace these script functions with matching, newly available event configurations.

The corresponding functions will be identified with a blue squiggly line in the Script Editor.



```
if (Event.OnChangeTag(ValueX))

endif
```

There are some obsolete functions used. Consider using a defined event in the 'Event Manager' instead. on line 1

The following table provides an overview of the functions and the corresponding recommended event type that should be used instead.

Function	Event type
Event.OnChangeTag	"Value change" tag event
Event.OnInterval...	Corresponding time event
Event.OnPresetOK	Other, "on Preset OK"
Event.OnRecipeLoad	Load recipe event

16.8 Events in Web visualization

Events are defined project-wide. This means that each defined event is available in all GALILEO instances (GALILEO local on the device, web visualisations).
The principle of operation is described in the → section " Web visualization", page 1070.

When using a web visualisation, you can define which actions are to be executed per event on which instance.
Depending on the event, it is better that an action is executed only for the local instance and not on the web client.

As an example

An event is defined that cyclically checks a PLC tag for a certain value.

- If the tag has this value, further PLC variables are to be set by script.
- In addition, a defined screen change is to take place

The setting of the PLC variables by script should happen only once, even if e.g. 2 web visualizations are active plus the local GALILEO instance. It is unlikely that the script to set the tag should be executed three times.

Also, it may be that the screen change should be defined differently per visualization.

For the configuration of all these possibilities, additional options appear in the Event Manager when using a web visualization.The following configuration is used to explain the options:

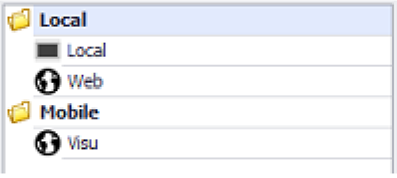
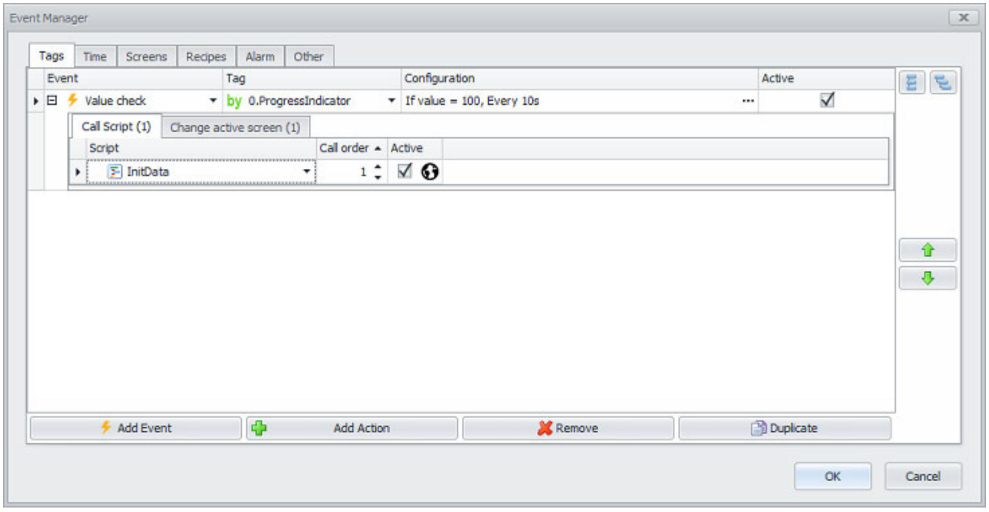


Figure 604: Device tab\ Visualization

Active

For each action, the 🌐-icon appears in the "Active" field.

Home tab\ Event manager



Event Manager

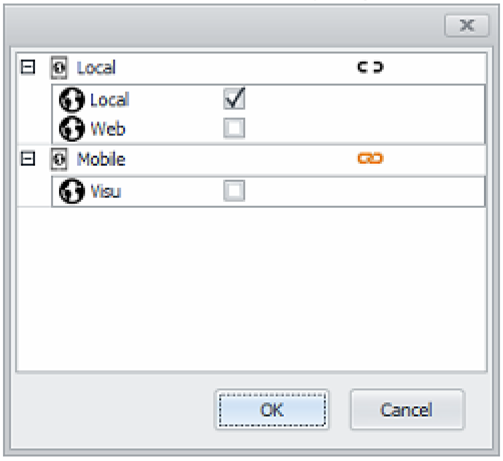
Event	Tag	Configuration	Active
Value check	0.ProgressIndicator	If value = 100, Every 10s	☑

Call Script (1) | Change active screen (1)

Script	Call order	Active
InitData	1	☑ 🌐

Figure 605: Pane Event Manager tabTags with configured web visualization

Clicking on the 🌐-icon opens a new dialog in which you can specify the visualization/visualization group for which this action is to be executed:



⊟ Local		⊂ ⊃
🌐 Local	☑	
🌐 Web	☐	
⊟ Mobile		🔗
🌐 Visu	☐	

For each individual visualization it can be configured here whether the action should be executed or not.

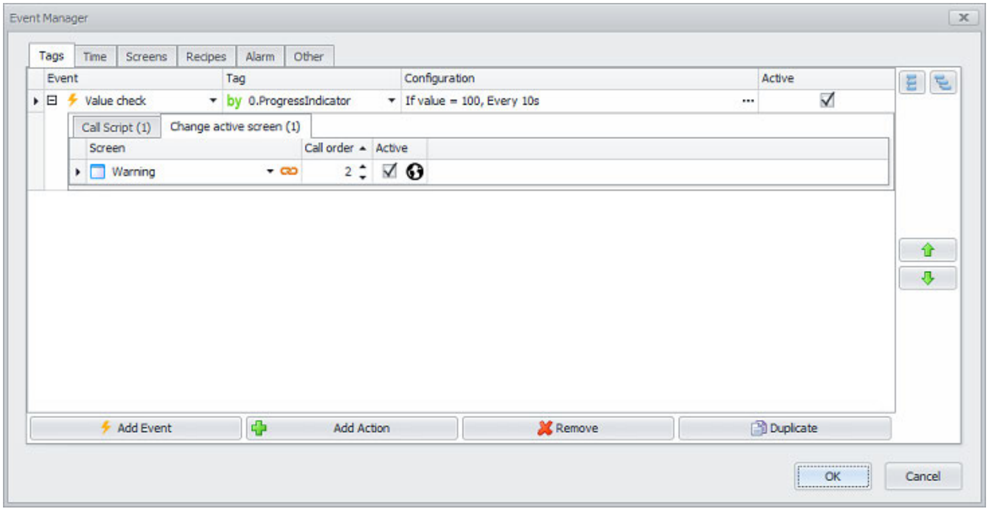
By clicking on the 🔗 icon (linked) or ⊂ ⊃ (unlinked) the configuration option can be changed:

🔗	Activation/deactivation of an action applies to all visualizations in this group.
⊂ ⊃	Action can be activated/deactivated per visualization in the group.

Linked/Separated screen definition for screen change

For screen change actions the target screen can be defined as linked/disconnected:

Home tab\ Event manager



Event Manager

Event	Tag	Configuration	Active
Value check	0.ProgressIndicator	If value = 100, Every 10s	☑

Call Script (1) | Change active screen (1)

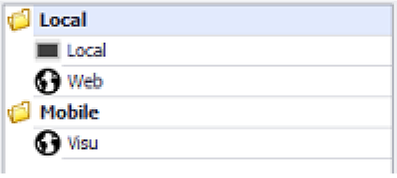
Screen	Call order	Active
Warning	2	☑

Figure 606: Pane Event ManagerTab Tags with web visualization configured

🔗 The configured target mask is used for all visualizations of the actively selected visualization group

⛓️‍💥 The target mask can be defined separately for each visualization of the visualization group

To change the active visualization group or the visualization to be edited in the Event Manager, the visualization must first be changed in the GALILEO Quick Access Toolbar:



17. Print

17.1 What can be printed?

Following print functions are possible:

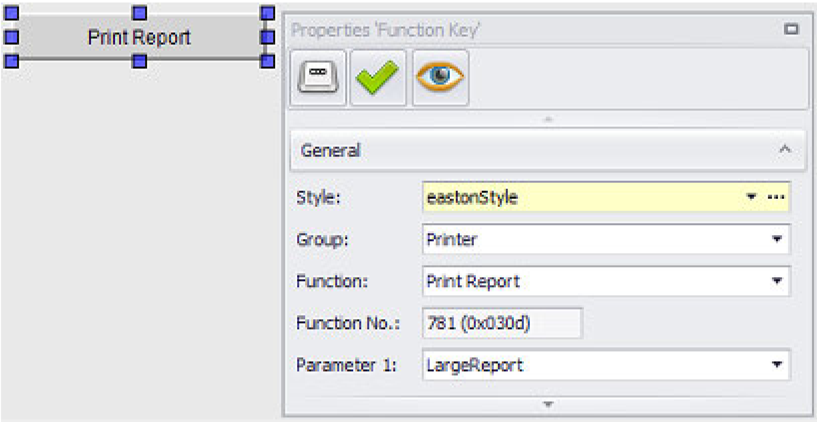
Function:	Description
Printing the Alarm List	If the Alarm List printout is activated, all errors will be printed directly when they occur, when they disappear and when they are acknowledged. See also: Help Screen Help Text Help Picture → section " Alarm list", page 867
Printing out the error history	Activate history to print out all errors listed there. See also: → section " Alarm list", page 867 → section "Print History", page 1033
Printreport	You can use print reports to print out current tag values. Print reports are created the same way as standard screens. The height and width of these reports can be defined freely. Printing will be carried out in the background, i.e., the actual screen will not be shown on the device. See also: → section "Print Report", page 1034 → section "Print report", page 1029
Printing the current screen	The screen currently being shown on the device will be printed out. See also: Help Screen Help Text Help Picture → section "Print Screen", page 1034

17.2 Triggering printing

Generally speaking, the following options can be used to trigger printing:

- A function key visualization object that calls a printer function, e.g., Print Report.
- A script that calls a printer function, e.g., using the following command line: <Printer.PrintReport (SmallReport);>.
- The Control data block
  4th data word: function number
  5th data word: screen number
  3rd data word, control bit 15: set to <1>

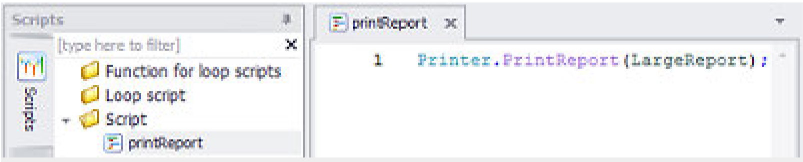
Example: Printing with a function key



Style: eastonStyle
Group: Printer
Function: Print Report
Function No.: 781 (0x030d)
Parameter 1: LargeReport

Figure 607: Screen with function key visualization object that calls the Print Report printer function in order to print out the <LargeReport> print report.

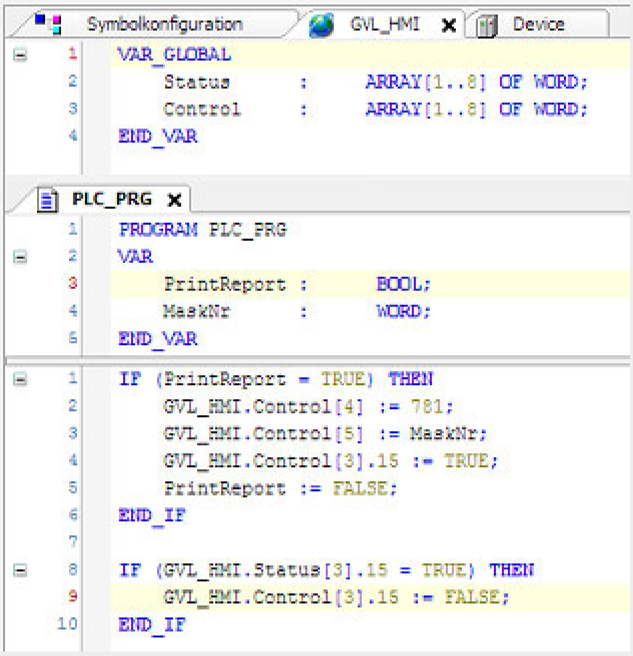
Example: Printing with a script



```
1   Printer.PrintReport(LargeReport);
```

Figure 608: Excerpt from the script with the function call used to print out the <LargeReport> print report.

Example: Printing with the control structure



```
VAR_GLOBAL
    Status      :     ARRAY[1..8] OF WORD;
    Control     :     ARRAY[1..8] OF WORD;
END_VAR
```

```
PROGRAM PLC_PRG
VAR
    PrintReport :      BOOL;
    MaskNr      :      WORD;
END_VAR
```

```
IF (PrintReport = TRUE) THEN
    GVL_HMI.Control[4] := 781;
    GVL_HMI.Control[5] := MaskNr;
    GVL_HMI.Control[3].15 := TRUE;
    PrintReport := FALSE;
END_IF

IF (GVL_HMI.Status[3].15 = TRUE) THEN
    GVL_HMI.Control[3].15 := FALSE;
END_IF
```

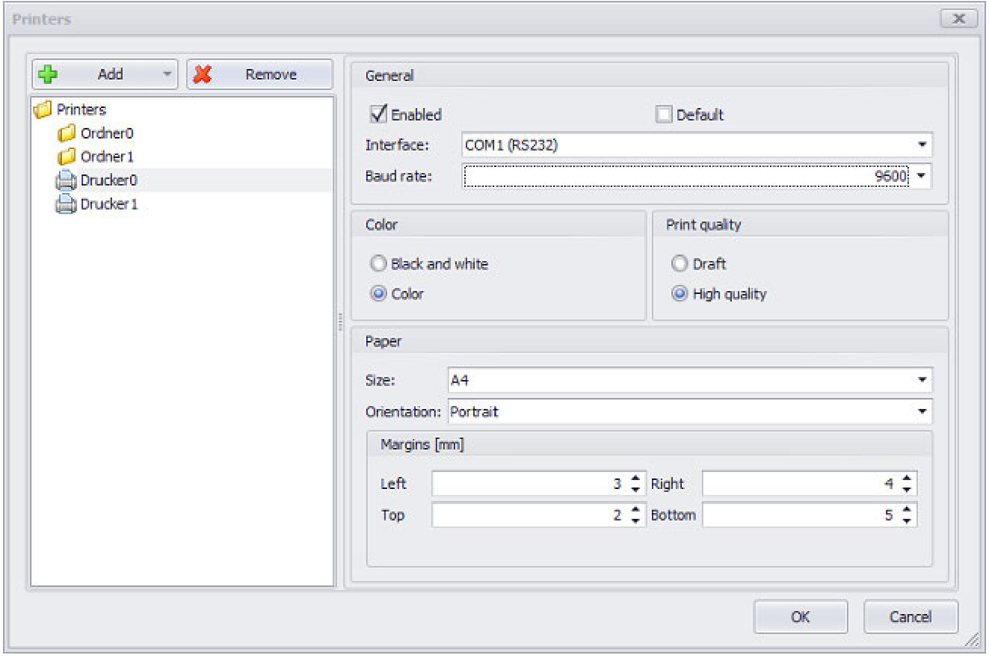
Figure 609: Excerpt from the PLC program used to set the required parameter

17.3 Printer

There are two ways to connect a printer to the device: connecting a serial printer using the serial port or connecting a network printer using the Ethernet port. For the device ports' actual pinout, please refer to the relevant device description.

The printer properties described below will be found in a text file with the PCL extension in the case of PCL-compatible printers. This printer driver file will be generated and saved for each individual printer that has been set up. For a description of printer drivers, please refer to the → section "Printer Driver Files (*.pcl)", page 1031 section.

Home\Project Configuration\Printer



- Printers
  - Ordner0
  - Ordner1
  - Drucker0
  - Drucker1

General

☑ Enabled □ Default

Interface: COM1 (RS232)

Baud rate: 9600

Color

○ Black and white

● Color

Print quality

○ Draft

● High quality

Paper

Size: A4

Orientation: Portrait

Margins [mm]

Left 3 Right 4

Top 2 Bottom 5

Figure 610: Window Printer

Add

Adds a new folder or printer to the project

To assign a name of your choice, click on the name. Spaces and special characters are not allowed.

Remove

Removes the selected printer or folder from the project

General

Enabled

Default

If this checkbox is enabled, the corresponding printer will be the default printer.

	If this checkbox is not enabled for any of the printers, GALILEO will use the first printer in the folder in which the printer driver files are located. The default printer will be used whenever a printer is not selected explicitly, as is the case, for example, with the Change to Printer printer function.
Interface:	COM1 (RS232) Ethernet1 USB

Printing via a serial port

If you select a serial port, e.g., <COM1 (RS232)>, you will also need to define the baud rate.

The following serial ports can be available for selection:

XV-100/XV-400: RS485 or SYSTEM PORT

XV-300: <COM1(RS…)>, …,<COMn(RS…)>

XP-500: RS232 or RS485

The remaining serial port parameters are as follows:

- Data width: 8 Bit
- 1 Stop Bit
- No parity
- Hardware handshake

Printing via an Ethernet port

Only printers that support the PCL protocol can be connected. PCL is a language that was originally developed by HP® but that is now supported by other printer companies as well.

The "Print job ... on queue ... has finished printing." printer message will not be shown if the following registry key is set (not equal to 0):

[HKEY_LOCAL_MACHINE\Printers\NoNotifyWhenPrinted]

The printer can either be used via a Windows PC's shared printer or directly using the RAW or LPR protocols.

Sharing Printer

All printouts are carried out via the selected Windows PC. The printer is selected as follows:

Multiple Line	Example
\\ <Network name of PC>\<Printer name>	\\PC-001\ColorPrinter

Where <Computer network name> should be the name of the computer (IP addresses are not supported) and <Printer name> should be the name of the shared printer.

RAW or LPR protocol

If the printer or printer server supports printing using the RAW or LPR protocols, the MICRO PANEL can activate the printer directly. This makes it unnecessary to have a PC with the Windows operating system in the network.

Multiple Line	Example
RAW: <IP address or network name>:<port number>	RAW:192.168.1.1
LPR: <IP address or network name> / <port name>	LPR:192.168.1.1/P2

Where <IP address or network name> should be the IP address or network name of the printer or printer server. Entering a port number or port name is optional. Port number 9100 and port name P1 will be used by default. Please refer to the specifications for your printer or print server.

Color

Defines whether the printout is to be in color or black and white.

Print quality

Used to select a print quality and, accordingly, a printing speed (draft or best quality)

Paper

Size Used to select a paper size, with the following options being available: A4, B4, B5, Legal, Letter.

Alignment Used to select an orientation: Portrait or Landscape.

Margins Used to define the margins that will be used when printing.
The unit is millimeters.

17.4 Print report

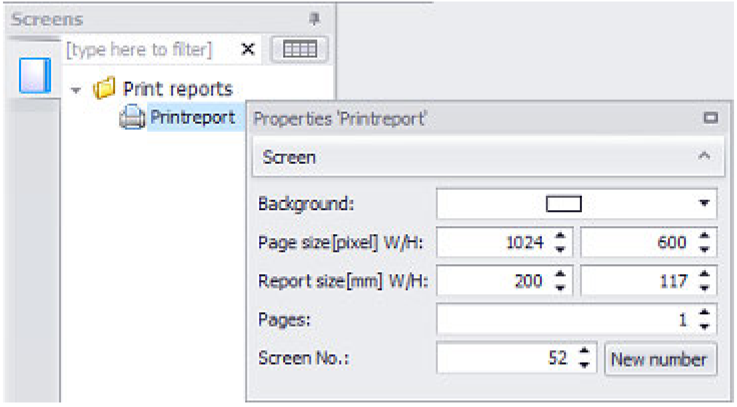
Print reports are ideal for providing labels, data sheets, and other product information for goods being produced. For example, they can be used to print out a label with up-to-date basic information for every roll of fabric produced at a textile plant so that each label can be affixed to the corresponding roll.

Print reports are screens that are only used for printouts. All visualization objects can be used for print reports, with the exception of the recipe, parameter list, alarm list, and help button visualization objects.

Print reports are created the same way as standard screens. The corresponding visualization objects are placed on the print report in order to design the printout. In addition, graph visualization objects can be printed even during plotting.

The height and width of these reports can be defined freely. Printing will be carried out in the background, i.e., the actual screen will not be shown on the device.

Project Tree\Print reports folder\"Printreport"



Background:

Page size[pixel] W/H: 1024 600

Report size[mm] W/H: 200 117

Pages: 1

Screen No.: 52

Figure 611: Properties pane for the "Printreport" print report

Background color

You can change the color if you want. The following options are available for this purpose:

- Transparent
- Standard Colors
- More Colors...

The most recently used colors will be shown (user-specific).

For detailed information on the available colors, please refer to the following section:
→ section "Color choice - Object's Color Palette", page 234

Page Size [pixel] W/H

Page width and height, in pixels.

Report Size: [mm] W/H

Display for entire print report with the configured width and height in mm.

Screen No. New number

When created in a project, every screen will automatically be assigned a screen number as a unique ID. A project cannot contain more than one screen with the same number. When assigning a screen number manually, you can assign any number of your choice.

These screen numbers become relevant when the PLC needs to trigger a print report. For a detailed description of how the PLC can be used to trigger a print report, please refer to the → section "Triggering printing", page 1024 and → section "Controlling functions with the PLC", page 602 sections.

If you click on New number, the next free screen number in the project will be automatically entered into the Screen No. field. Screen numbers freed up when a screen is deleted will also be taken into account when determining the next free screen number.

17.5 Printer Driver Files (*.pcl)

Printer driver files contain all the settings and definitions required for print jobs for the PCL-compatible printer connected to the device.

These print jobs can be started by function keys on the device or by a connected PLC; please refer to → "Printer functionalities (printer)", page 1033.

The printer driver files will be generated during compilation and saved in the *.\appl* directory. These files can be edited (to change the IP address for a network printer, for example) without having to open GALILEO, change the project, compile it again, and transfer it to the device.

Example showing printer driver for HP-compatible printer

```
// ************************************************
// *** PCL Printer Driver for HP Laserjet/Deskjet ***
// ************************************************
//
#D00# "Paper size (0=A4, 1=B4, 2=B5, 3=Letter, 4=Legal)" < 2 >
#D01# "Orientation (0=Portrait, 1=Landscape)" < 1 >
#D02# "Left Margin in mm" < 15 >
#D03# "Right Margin in mm" < 20 >
#D04# "Top Margin in mm" < 10 >
#D05# "Bottom Margin in mm" < 25 >
#D06# "Color mode (0=b/w 1=color)" < 0 >
#D07# "Print resolution (0=draft, 1=high)" < 0 >

#T00# "Printer driver name (standard: PCL Inkjet)" < PCL
Inkjet >
#T01# "Default Printer path (or f.e. LPT1: or COM1: 19200 or
USB:)" < \\Desktop\hpdeskjet3820_usb >
```

17.5.1 PLC File Settings

#D00# Paper Size	Used to define the paper size 0=A4 1=B4 2=B5 3=Letter 4=Legal
#D01# Orientation	Defines the sheet orientation. 0=Portrait 1=Landscape
#D02# Left Margin	Defines the left margin in [mm] that the printer must allow for.
#D03# Right Margin	Defines the right margin in [mm] that the printer must allow for.
#D04# Top Margin	Defines the top margin in [mm] that the printer must allow for.
#D05# Bottom Margin	Defines the bottom margin in [mm] that the printer must allow for.
#D06# Color Mode	Defines whether the printout is to be in color or black and white. 0=Black and white 1=color
#T00# "Printer Driver Name"	Defines the name of the printer driver . The name of the default printer driver is "PCL Inkjet".

17.5.1.1 #T01# "Default Printer Path"

This parameter is used in different ways, depending on the type of printer used:

Catalog Number	Meaning of the parameter	Example
Serial printer	Port and baud rate	COM1: 19200
Parallel Printer	Interface	LPT1:
Network Printer	Network path	\\Desktop\hpdeskjet3820_usb

17.6 Printer functionalities (printer)

17.6.1 Change to Printer

This function switches the print job to a different printer.

After the function has been executed, all print jobs are output to the selected printer.

Number of the Function	800 (320h)
Executable from the PLC	No
Function with ON/OFF state	No
Function available in standard screens	Yes
Function available with keyboards	No
Function available in script	No
Function available in application entry script	No

17.6.2 Print History

Prints the History of the alarm list's accumulated error messages.

Number of the Function	768 (300h)
Executable from the PLC	Yes
Function with ON/OFF state	No
Function available in standard screens	Yes
Function available with keyboards	No
Function available in script	Yes
Function available in application entry script	Yes

Error messages can be printed out as follows:

Column	Description
Status	This column shows the current status of the error. 1 - Error active 0 - Error no longer active X - Error confirmation mode
Code	Shows the error number
Date / Time	Date/Time entry in format: "DD.MM.YY hh:mm.ss" and with US time format "MM-DD-YY hh:mm.ss am/pm"
Error text	Error description, which is displayed with every error..

17.6.3 Print Parameter List

This function enables the printing of parameter lists. The parameter list to be printed must be defined as a parameter.

Number of the Function	801 (321h)
Executable from the PLC	No
Function with ON/OFF state	No
Parameter entered	Parameter list
Function available in standard screens	Yes
Function available with keyboards	No

Function available in script	Yes
Function available in application entry script	No

17.6.4 Print Parameter List (no Header)

This function enables the printing of parameter lists. The parameter list to be printed must be defined as a parameter.

Unlike the Print Parameter List function, no header line is printed.

Number of the Function	802 (322h)
Executable from the PLC	No
Function with ON/OFF state	No
Parameter entered	Parameter list
Function available in standard screens	Yes
Function available with keyboards	No
Function available in script	Yes
Function available in application entry script	No

17.6.5 Print Report

This function can be used to print a print report. In GALILEO, print reports can be set up like a screen and can have any height and width. Please note that there are certain objects that cannot be used in screens of type print report.

The values will be acquired when the function is triggered.

Number of the Function	781 (30dh)
Executable from the PLC	Yes
Function with ON/OFF state	No
Parameter 1	Print report screen name
Function available in standard screens	Yes
Function available with keyboards	No
Function available in script	Yes
Function available in application entry script	Yes

17.6.6 Print Screen

Saves the current screen contents when the function was triggered and prints it out. The printing is carried out in the background. This means that the printout may be slightly slower than normal.

A header line with title, date and time is printed at the beginning of the page.

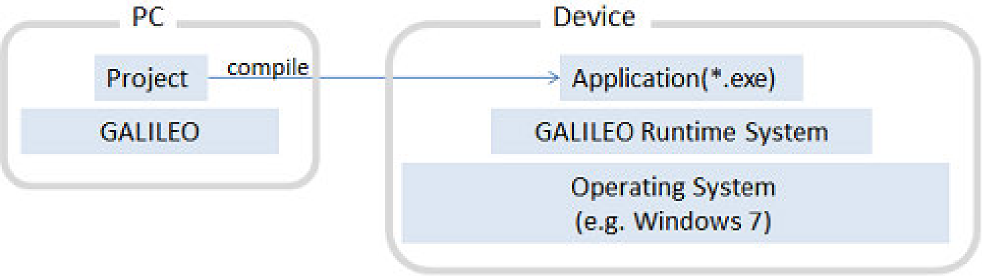
Number of the Function	788 (314h)
Executable from the PLC	Yes
Function with ON/OFF state	No
Function available in standard screens	Yes
Function available with keyboards	No
Function available in script	Yes
Function available in application entry script	No

18. Compile and transfer

This section focuses primarily on how to load a project onto a touch panel and start running it.
In order for a project to be executable on a device, it needs to be compiled first. Once it is successfully compiled, all the project files for the runtime program will be generated. This means that there is always a source and a runtime program for each project.

The runtime program is used by the GALILEO Runtime System (GRS), while the source project is used by the Automatization Software GALILEO program.

The runtime program is transferred to the device.



The executable file can be transferred to any storage location connected to the computer via FTP, e.g., to a memory card inserted in the computer, to a device's memory, or to a memory card inserted in a device. Some of these options may not be available depending on the specific device model being used.

18.1 Building and deploying

In order to be able to load a project onto a touch panel, you will first need to decide how you want to deploy it. There are basically tree deployment methods available:

- FTP connection
  Loads the project onto the touch panel via FTP
- Filesystem directory
  Used to save the project on your computer.
  After this, you can use a data storage device, such as an SD card, to transfer the GRS manually from your computer to your touch panel.
- Booting from an SD memory card
  When using this option, the project needs to be on an SD memory card. Every time the device boots up, the project will be started from the SD memory card inserted in it.

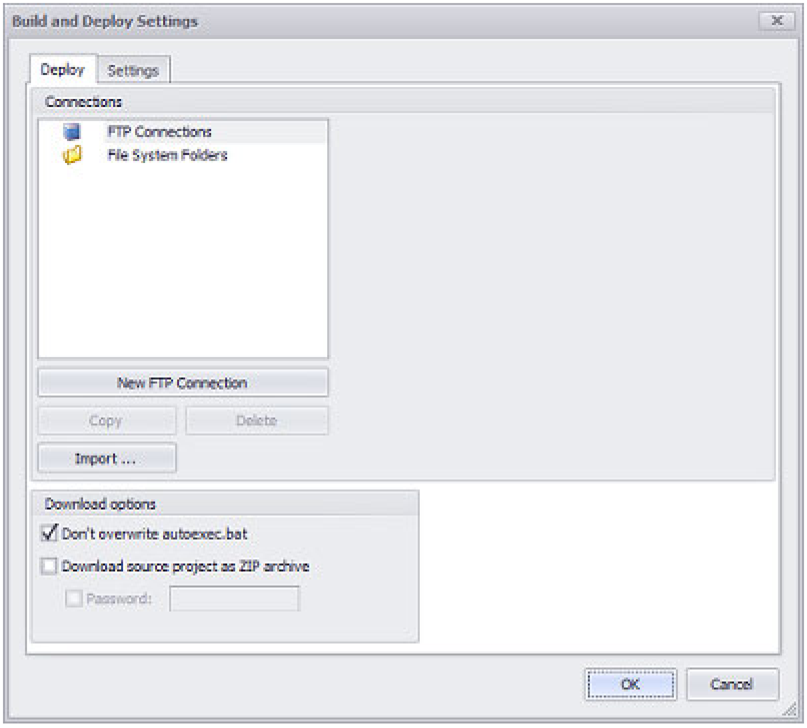
18.1.1 Transferring the Project to touch panel

In order to deploy a project, you must first define the connection that will be used for this purpose.

► Click on in Home▶\ Build and Deploy to open the Build and Deploy Settings dialog box

The <Build and Deploy Settings> dialog box is used to define the project's destination directory, as well as how the project will be exported as a GRS.

Home\Simulator\Build and Deploy settings



Build and Deploy Settings

☑ Don't overwrite autoexec.bat

□ Download source project as ZIP archive

□ Password:

Figure 612: Build and Deploy Settings dialog box

18.1.1.1 Transferring a project via FTP

To load the project onto a touch panel using an FTP server, you will first need to create an FTP connection. In addition, the FTP server application must be running on the device in order to be able to transfer the data: FTPSVR.EXE.

→ FTP connections are only supported by touch panel devices with an Ethernet interface.

Starting the FTP server

Depending on the specific device being used, there will be various ways to start the FTP server.

For devices in the XV100, XV300, XV400 product groups:

1. To start the FTP server on the device, click or tap on the following: *Windows Start menu\Programs\Communication\FTP Server*

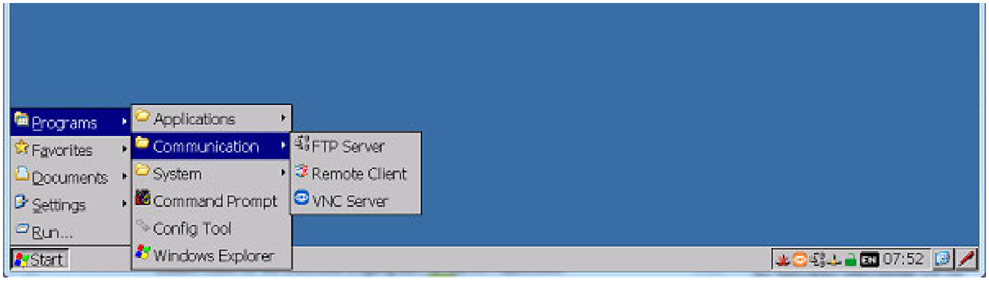


Figure 613: XV300 screenshot

The FTP server is described in detail in the system description for windows; please refer to → section "Further usage information", page 1143.

2. To start the FTP server with a line in the autoexec.bat file, open the file in GALILEO by clicking on *Home\CE-Configuration\Autoexec.bat tab* and add the following line: "START FtpSvr.exe -h".

For information on how to start the FTP server in other cases: Refer to the standard Windows 7 documentation for XP500 devices and to the description for the relevant target operating system for GALILEO OPEN.

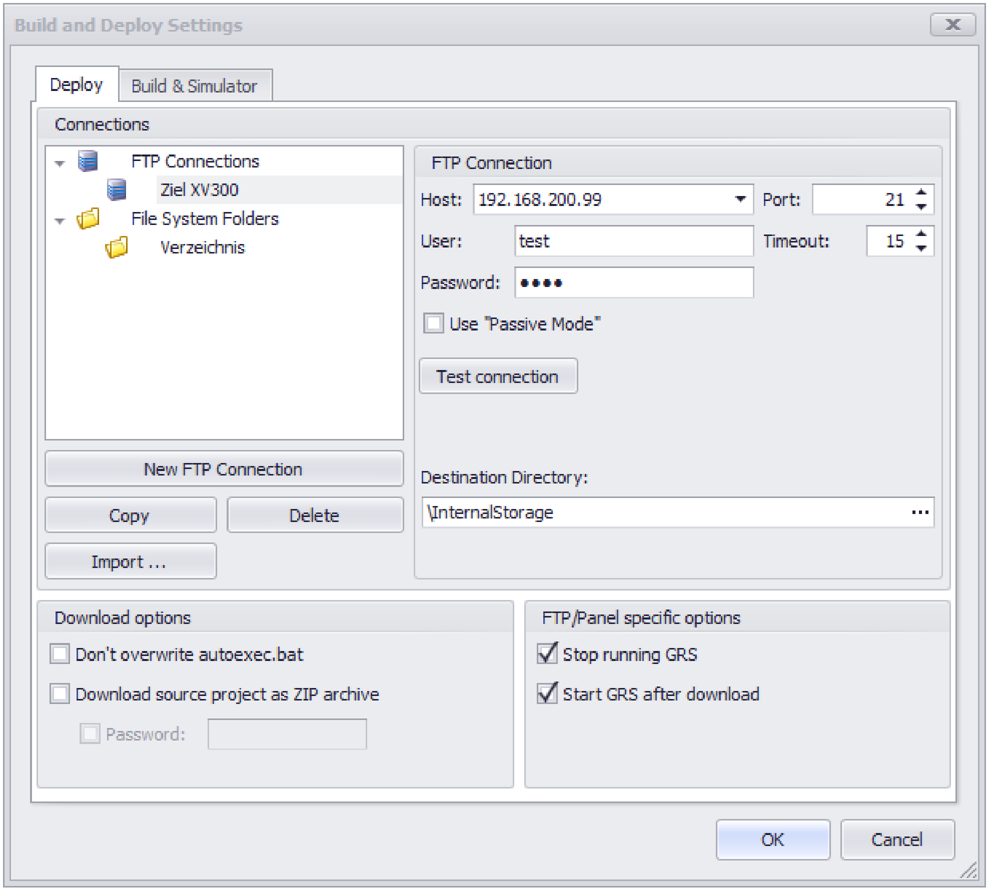
Creating an FTP connection

- Select a FTP Connection
- Click on New FTP Connection.

The dialog box will show the various options that can be used to configure the FTP connection you just created.

- Enter the name you want for the FTP connection

Home\Simulator\Build and Deploy settings



Host: 192.168.200.99
Port: 21
User: test
Timeout: 15
Password: ••••
☐ Use "Passive Mode"
Destination Directory: \InternalStorage

☐ Don't overwrite autoexec.bat
☐ Download source project as ZIP archive
☐ Password:

☑ Stop running GRS
☑ Start GRS after download

Figure 614: Dialog box Build and Deploy Settings

▶ Configure the FTP connection

Host

The target system's (touch panel or PC) IP address
This IP address can be shown or changed on the touch panel by going to:
"Start/Programs/Control Panel/Network/ONBOARD1" or
"Start/Programs/Config Tool/Network/Edit Configuration/ENET1|ENET2"

User

The touch panel's FTP username
The FTP username can be shown on the touch panel, provided it has been configured for the FTP server, by going to:
"*Start/Programs/Control Panel/Owner"* or
"Start/Programs/Config Tool/Advanced Settings/Users"

Password

The touch panel's password
The password can be shown on the touch panel, provided it has been configured for the FTP server, by going to:

	"*Start/Programs/Control Panel/Owner"* or *"Start/Programs/Config Tool/Advanced Settings/Users"* The touch panel does not necessarily need to have a password. If a password has not been set on the touch panel, leave this field blank.
Use "Passive Mode"	By default, "Passive Mode" will be enabled for the FTP connection so that data can be transmitted faster. If the FTP connection is blocked by security IT components such as antivirus software or firewalls, this mode can be disabled.
Port	The port used for FTP communications. Default setting: 21
Timeout [s]	If the timeout set here elapses before a connection to the touch panel can be established, the system will stop trying to establish a connection to the touch panel.

▶ Run Test Connection.

> *CAUTION*
>
> There must be an active FTP server on the touch panel. Otherwise, it will not be possible to establish an FTP connection between your computer and the touch panel.
>
> → The FTP server can normally be shown on the touch panel by going to: *"Start/All Programs/Communication/FTP Server"*

If the connection test fails, check to make sure that:

- Your target system is connected and active.
- Your IP address is correct.
- The FTP server on your target system is running.

The connection test must be successful.

Once the connection test is successful, you can define the destination directory for the project.

▶ Define the destination directory for the current project.

More Settings

The FTP/Panel specific options and Download options sections can be used to configure additional settings that determine how the FTP connection will behave.

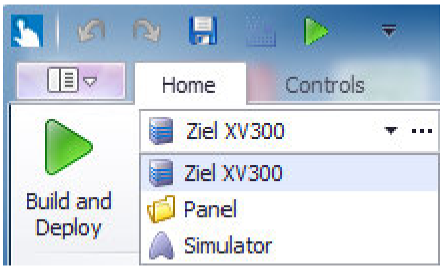
If you want the GRS to be tested immediately after being loaded onto the device, enable the ☑Stop running GRS and ☑Start GRS after download options.

You have now created a new FTP connection.

Transferring the project with the FTP connection you just created

Now that you have successfully established an FTP connection to your touch panel, you can follow the steps below to load your program onto the touch panel.

- ▶ Use the Simulator drop-down menu in the Build and Deploy group under the Home tab to select your FTP connection.

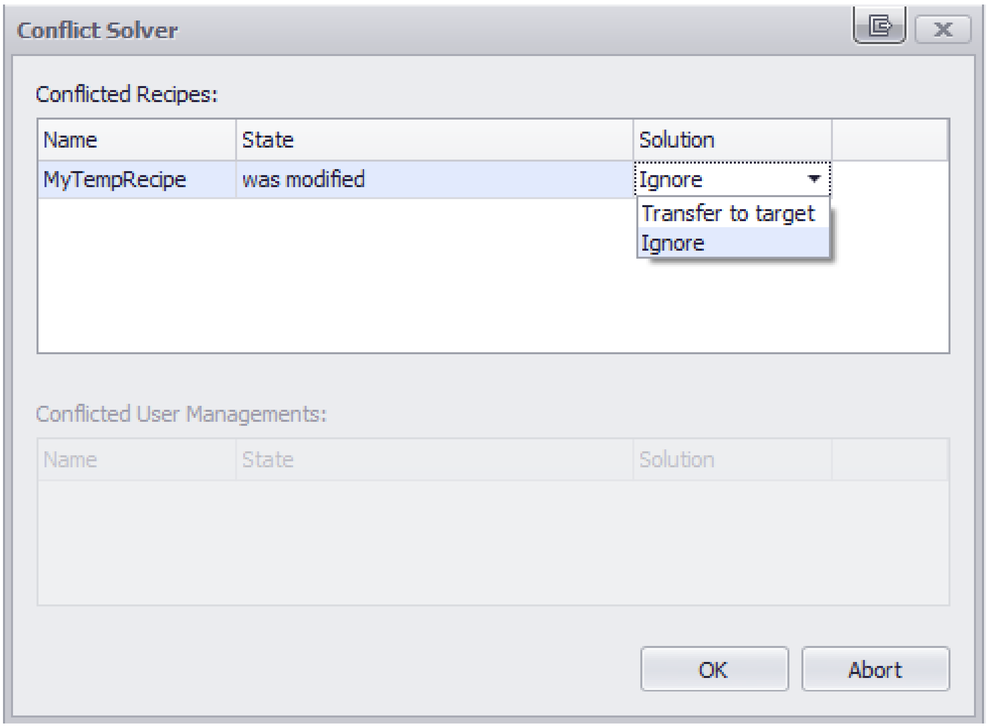


- ▶ Click on Build and Deploy

The project will be compiled and transferred to the configured destination directory via FTP.

If the following discrepancies are found between the files on the PC and the files on the device, a prompt will appear asking whether you want the corresponding files to be copied to the device:

- The operating system on the PC and the operating system on the device have different versions. The following prompt appears:
  If you click on the Yes button, the operating system with the newer version will be loaded onto the device together with the GALILEO application.
  If the operating system is loaded onto the device, the device will be automatically rebooted.
- The device's destination directory does not contain the same GALILEO project. A system message indicating the project conflict will appear.
  If you click on OK, the old project on the device will be deleted and the compiled project will be loaded onto the device.
  If the FTP/Panel specific options\Start GRS after download option is enabled, the project will be started automatically.
- Recipe files (.TAB, .ACT) in the "DATA" subdirectory
  These recipe files will only be found on the PC if they have been created with the Recipe Editor or transferred to the PC with Upload Data (Local/FTP). If recipes have been changed in the GALILEO application using the Recipe Editor dialog box, a prompt will appear when compiling the new *.TAB and *.ACT recipe files and ask if you want them to be loaded onto the device so that the recipe files already on the device will be overwritten.
  The following system message will appear: Conflict solver



Conflict Solver

Conflicted Recipes:

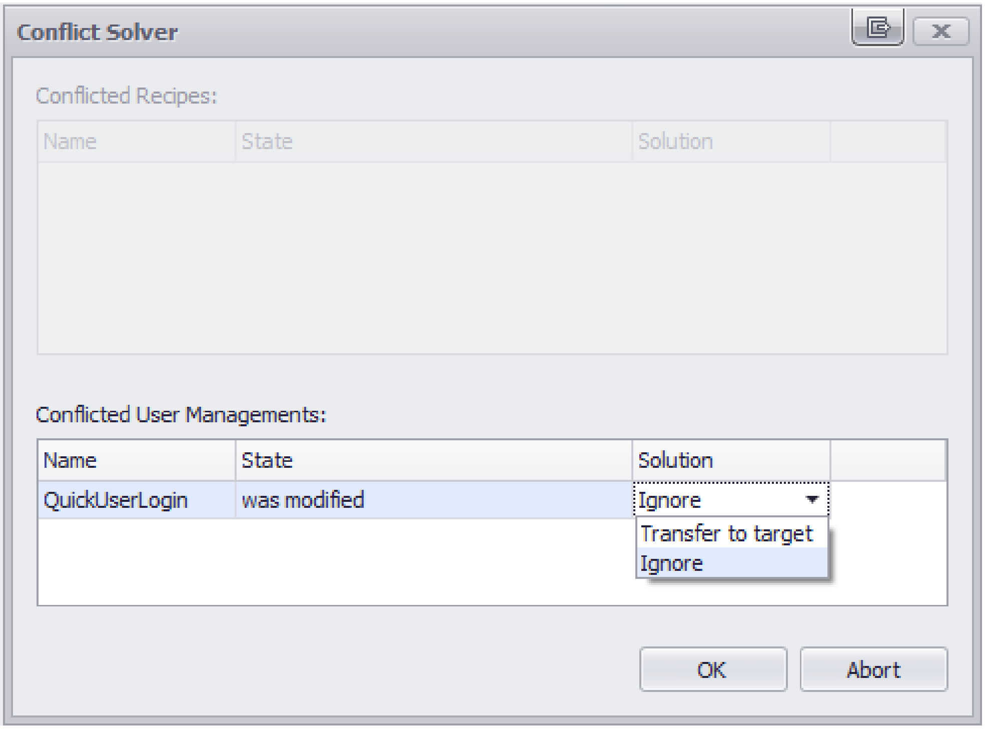
Name	State	Solution
MyTempRecipe	was modified	Ignore

Conflicted User Managements:

Name	State	Solution

If you select the <Transfer to target system> option in the Solution column and then click on OK, the recipe file will be loaded from the PC onto the device. Accordingly, the recipe file already on the device will be overwritten.

- User management file (*.PWM) in the "DATA" subdirectory
  User management data will only be found on the PC if it has been created in the User Management dialog box. If new users are added in the GALILEO application using the User Management dialog box, or if user parameters are changed, a prompt will appear when compiling the new *.PWM user management file and ask if you want it to be loaded onto the device so that the user management file already on the device will be overwritten. The following system message will appear:



Conflict Solver

Conflicted Recipes:

Name	State	Solution

Conflicted User Managements:

Name	State	Solution
QuickUserLogin	was modified	Ignore

If you select the <Transfer to target system> option in the Solution column and then click on OK, the user management file will be loaded from the PC onto the device. Accordingly, the user management file already on the device will be overwritten.

→ Once you select an option in the <Conflict Resolver> dialog box, your choice will be saved in the project. This means that the Conflict Resolver dialog box will not be opened again (in the event of discrepancies between files on the PC and files on the device) until after the project is closed and then opened again.

Once the GALILEO application is loaded onto the touch panel and no errors are shown in the GALILEO Output pane, the touch panel should start behaving as configured with the FTP/Panel specific options; please refer to → section "FTP/Panel specific options", page 1059 as well.

Starting the GRS manually

▶ Open the directory on the touch panel.

▶ Open the runtime tab.

Depending on the specific panel model, this folder will contain a file named GRS_CE5.exe, GRS_CE7.exe, or GRS_Win32.exe. You can use this file to start your GALILEO application on the touch panel.

18.1.2 Transferring the project to a filesystem directory

To load a project onto a touch panel using a filesystem directory, a directory must first be created for this purpose.

Creating a file directory

This pane is used to set up the destination directory for the project.

- Select File System Folders
- Click on New File System Folder.
- Name the directory you just created.

The destination directory needs to be specified in the section to the right.

Home\Simulator\Build and Deploy settings

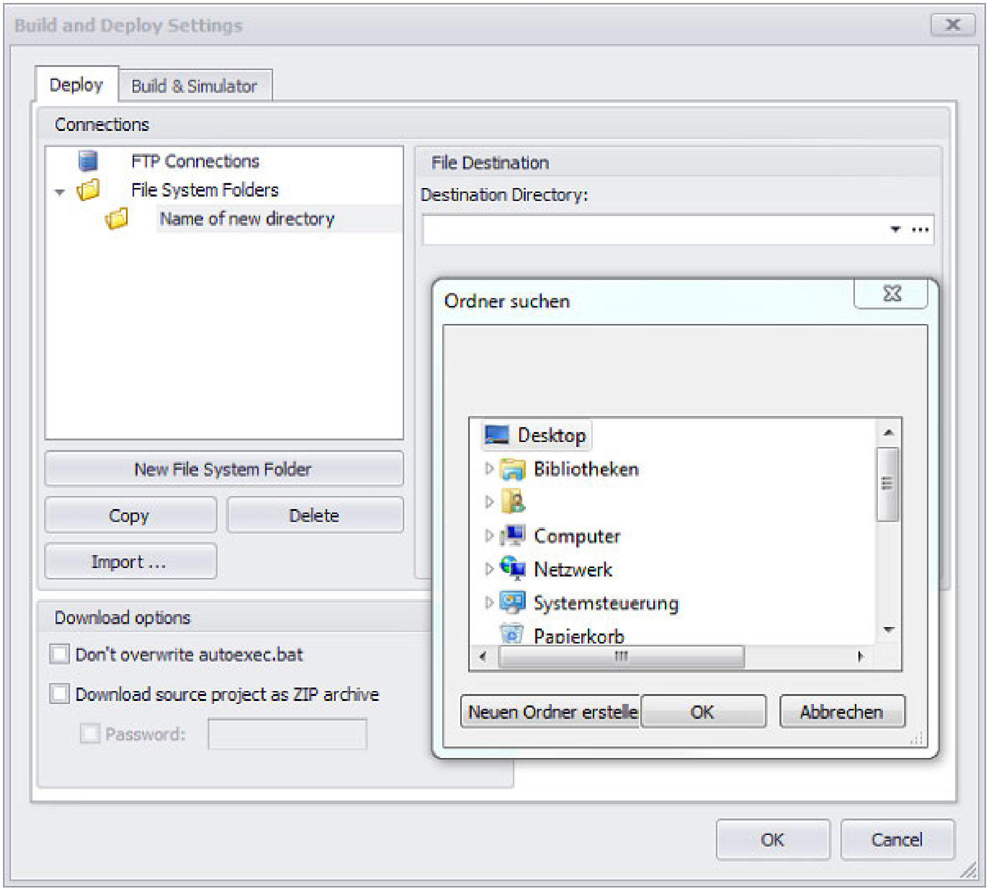


Figure 615: Build and Deploy Settings dialog box

- Define the destination directory for the GALILEO application. Create a new folder if necessary.

More Settings

You can use the options in the Download options section to configure additional settings that will determine how the connection will behave. For a detailed description of these download options, please refer to → "Download options", page 1061

You have now created a new filesystem directory.

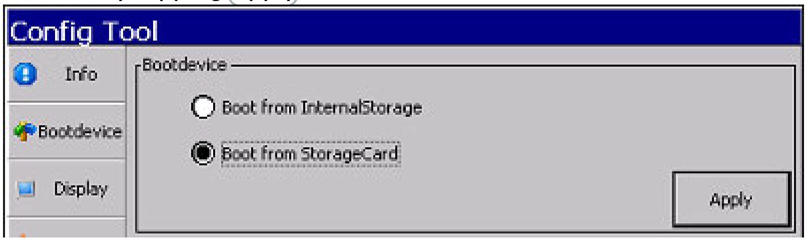
18.1.3 Booting from an SD memory card

To boot a device from an SD memory card, the GALILEO project needs to be built and then deployed to the SD memory card. In addition, the settings on the device must be configured in such a way that the device will boot from the SD memory card.

To do so, follow the steps below:

Settings on device

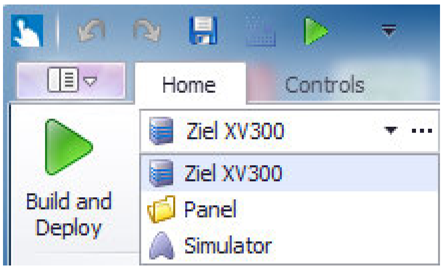
- Double-tap Config Tool on the device display.
- Open the Bootdevice tab and enable the <Boot from StorageCard> option.
- Confirm by tapping Apply.



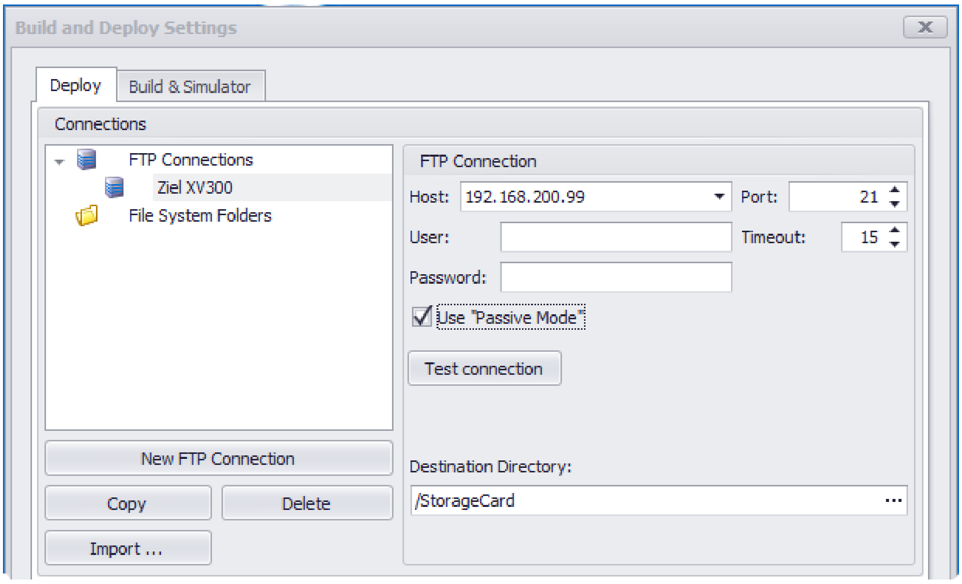
Alternatively, you can also change the settings on the device by using the <autolaunch.inf> file, which will be run automatically when the SD memory card is inserted. The <autolaunch.inf> file is described in detail in a different manual; please refer to Windows Embedded Compact 7 System Description.

Saving the project on the SD memory card in the device

- Insert the SD memory card into the device's slot.
- Open the Home tab and select your FTP connection from the Simulator drop-down menu for Build and Deploy.



- In the Destination Directory, select the SD memory card with /StorageCard.



Host: 192.168.200.99 Port: 21
User: Timeout: 15
Password:
☑ Use "Passive Mode"
Destination Directory: /StorageCard

- Click on OK.
- Click on Build and Deploy

The GALILEO application will be compiled and saved to the SD memory card in the device.

→ Do not forget to check the destination directory!
When you select an FTP connection, < /InternalStorage> will be selected as the Destination Directory by default.

The project in the SD memory card will start automatically if the <Start GRS after download> option is enabled in the <Build and Deploy Settings> dialog box.

18.1.4 Building and deploying projects with a command line call

→ The FTP server must already be running on the device so that the <deployment> will work!

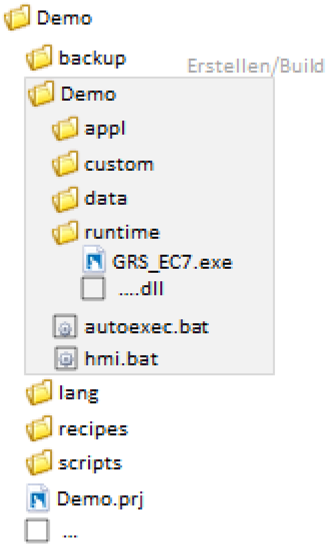
GALILEO can be called with or without a project by using the command line.

Likewise, a GALILEO project can be built and deployed with a command line call on a PC or other device. This command line call will be executed without any additional user prompts or dialog boxes. This deployment method can be used for all XV devices.

By using the command line calls in a batch file, you can create an automated routine for compiling projects and loading them onto devices. This way, you will not have to open the GALILEO application separately and carry out the necessary commands by hand there.

Create

When a GALILEO project is built, it is compiled. By default, a new subdirectory will be created in the directory in which the GALILEO project is found. This subdirectory will have the same name as the GALILEO project. Moreover, four subfolders will be created in it: \APPL, \CUSTOM, \DATA, and \RUNTIME. These subfolders will contain all the files for the compiled application, as well as the AUTOEXEC.BAT and HMI.BAT files. By default, an .exe file will be generated based on the panel model selected in *Home\Panel Type*, e.g., GRS_EC7.exe for XV300. This file will be found in the \RUNTIME directory.



```
Demo
    backup
    Demo
        appl
        custom
        data
        runtime
            GRS_EC7.exe
            ....dll
        autoexec.bat
        hmi.bat
    lang
    recipes
    scripts
    Demo.prj
    ...
```

Figure 616: Directory structure example; the Demo folder is created and filled with the required files when the project is built

Deploy

When a project is deployed, the GALILEO application is loaded onto the device. The \APPL, \CUSTOM, \DATA, and \RUNTIME folders will be loaded into the directory defined in GALILEO *Home\Build and Deploy\Deploy\Field: Destination Directory*.

18.1.4.1 GALILEO start

GALILEO can be run using the following command line call:

```
[Path]\Galileo.exe
```

Examples:

```
"C:\Program Files\Eaton\Galileo\Galileo.exe"
"C:\Program Files\Eaton\Galileo\Galileo" -help
```

18.1.4.2 Starting GALILEO with a project and additional parameters

GALILEO can be run using the following command line call:

```
[Path]\Galileo.exe -proj=[projectfile] -look=[style] -lang=
[language] -resetLayout
```

Examples:

```
"C:\Program Files\Eaton\Galileo\Galileo.exe" -
proj="C:\Demo\Demo.prj"
```

```
"C:\Program Files\Eaton\Galileo\Galileo.exe" -
proj="C:\Demo\Demo.prj" -lang=en -look=dark
```

18.1.4.3 Building a GALILEO application

→ To pass this parameter, switch to the directory where GALILEO is installed in Command Prompt. If you do not switch to this directory, the call will not do anything.

→ The -build , -builddeploy, and -deploy parameters must be passed together with the -nogui parameter.

→ The command must be entered without any line breaks.

To compile a GALILEO project and build an application, follow the steps below:

GALILEO -nogui -proj=[projectfile] -build

Example:

```
Galileo -proj="C:\Demo\Demo.prj" -build -nogui
```

Creates the following in the "C:\Demo\" directory: a "Demo" subdirectory with four subfolders named \APPL, \CUSTOM, \DATA, \RUNTIME. These subfolders will contain all the files for the compiled application, while the AUTOEXEC.BAT and HMI.BAT files will be found in "C:\Demo\Demo".

Deploying a GALILEO application

GALILEO -nogui -proj=[projectfile] -deploy
-deploytarget=[target]|-deploytargetname=[targetname] -timeout=[timeout]
-level=[level] -keepautoexec -conflicts=[solution] -zip -zippw=[password] -startgrs

Building and deploying a GALILEO application

GALILEO -nogui -proj=[projectfile] -builddeploy
-deploytarget=[target]|-deploytargetname=[targetname] -timeout=[timeout]
-level=[level] -keepautoexec -conflicts=[solution] -zip -zippw=[password] -startgrs

18.1.4.4 Overview of build and deploy command line calls

The following parameters are available for command line calls:

→ The parameters are case-sensitive!

Fill out the brackets with the permitted characters or values you want.

Parameters for starting GALILEO	Description
-help	This parameter can only be used without the "EXE" filename extension. Lists all command line parameters together with a brief description. Example: galileo -help
-proj=[projectfile]	The GALILEO project with which GALILEO should start.

	Example: galileo -proj="C:\Demo\Demo.prj"

-look=[style]	GALILEO user interface style
	style=Dark;
	style=Light;
	Please refer to *Home\Style*

-lang=[language]	GALILEO user interface language
	language=de: German
	language=en: English
	Please refer to *Home\Language*

-resetLayout	GALILEO's layout will be reset the next time the program is started. The layout for the user interface panes in GALILEO will be set to its default. Please refer to *Home\Reset Layout*
Build and deploy parameters	
-nogui	Starts GALILEO in a mode without a user interface; can be used to run automated "build" and "deploy" processes with the use of batch files.
-build	To pass this parameter, switch to the directory where GALILEO is installed in Command Prompt. This parameter will only work if -nogui is passed at the same time. This parameter will only work if the GALILEO project has been defined in the command line: -proj Builds the GALILEO application, creates a subdirectory with the project's name in the -proj directory, and stores the application in this subdirectory.
-builddeploy	To pass this parameter, switch to the directory where GALILEO is installed in Command Prompt. This parameter will only work if -nogui is passed at the same time. This parameter will only work if the GALILEO project has been defined in the command line: -proj A destination directory must be declared with -deploytarget or -deploytargetname. Otherwise, an error message will appear in Command Prompt. Builds the GALILEO application, creates a subdirectory with the project's name in the -proj directory, and stores the application in this subdirectory. After this, the application will be loaded onto the target specified with the -deploytarget or -deploytargetname parameter.
-deploy	To pass this parameter, switch to the directory where the GALILEO application is installed in Command Prompt. This parameter will only work if -nogui is passed at the same time. This parameter will only work if the GALILEO project has been defined in the command line: -proj The GALILEO project must have already been built. A destination directory must be declared with -deploytarget or -deploytargetname. Otherwise, an error message will appear in Command Prompt.

	Loads the GALILEO application that has already been built onto the target specified with the -deploytarget or -deploytargetname parameter.

-level=[level]	You can use [level] to manage the granularity of the output messages in GALILEO.
	Level=0: Fine; (default)
	Level=1: Medium;
	Level=2: Coarse

-deploytarget=[target]	[target] is the destination directory into which the compiled GALILEO application will be loaded. This destination directory can be a folder or a memory card on the PC. The destination directory can also be a folder on a device. In this case, it will be necessary to transfer the GALILEO application with the use of FTP. The syntax in this case will be as follows: ftp://[<user>[:<pass>]@]<host>[:<port>]/<dir> Example: -deploytarget=ftp://192.168.200.99/InternalStorage Example: -deploytarget=C:\temp For an example, please refer to the diagram → "Dialog box Build and Deploy Settings", page 1039 -deploytarget=ftp://test:1234@192.168.200.99:21/InternalStorage
-deploytargetname= [targetname]	[targetname] is the destination directory [target] into which the compiled GALILEO application will be loaded. It is defined with an FTP connection name. The name of the FTP connection can be defined GALILEO in ***Home\Build and Deploy\Build and Deploy Settings\Deploy tab\FTP Connection***. For an example, please refer to the diagram → "Building and deploying", page 1036 -deploytargetname=Destination XV300
-timeout=[timeout]	Time, in seconds, after which an error will be reported if an FTP connection cannot be established.
Deploy parameters	
-zip	This parameter will only work if the GALILEO project has been defined in the command line: -proj The GALILEO project will be saved in the destination directory's \BACKUP subdirectory as a ZIP file. You can set the destination directory in GALILEO ***Home\Build and Deploy\Build and Deploy Settings\Deploy***
-zippw=[password}	This parameter will only work if -zip is passed at the same time. You can assign a password that will be required in order to open the ZIP file.
-conflicts=[solution]	This parameter will only work if the destination directory has ben set with the -deploytargetname or - deploytarget parameter in the command line. Used to define how potential conflicts regarding recipes and user management should be handled. Within this context, a "conflict" is when a

	recipe/the user management is different between the GALILEO project and the destination directory. solution = 0: (Default Settings) The files in the destination directory will be kept solution = 1: The files in the destination directory will be overwritten If the storage location for a recipe or the destination directory has been changed in the meantime, the recipes/user management will be transferred to the target system regardless.
-keepautoexec	When this parameter is passed and an autoexec.bat file already exists on the target system, the file will not be overwritten. If this parameter is not passed, or if there is no autoexec.bat file on the target system, the autoexec.bat file will be transferred.
-startgrs	This parameter will only work if -deploy or -builddeploy is passed at the same time. Starts the GALILEO application after it has been deployed on the device, If the application will not start, open the GALILEO project and check in ***Home\Panel Type*** to make sure you selected the correct device.

18.1.4.5 What will happen in the event of an error?

In the event of an error, such as when the connection to the device is terminated because the wrong panel model has been selected in GALILEO, the compilation (building) process will be aborted and an error will be shown in the form of red text in GALILEO's Output pane.

Following is a list of potential conflict situations and how they are handled:

- If a compiled, but not identical GALILEO application is already found in the destination directory, the application and all data, such as recipes and the user management data, will be deleted.
- If the GALILEO Runtime System is already running on the device, it will be terminated automatically.
- If the target platform cannot be determined, e.g., because an outdated operating system is running, the GALILEO application will still be transferred, even if it is not possible to check for compatibility.
- If the destination directory on the device does not contain an "OS" directory, this directory will be created and the operating system will be loaded onto the device together with the application.
- If the operating system is loaded onto the device together with the application, the device will be rebooted no matter what.

18.1.4.6 Building and deploying a project in such a way that all files are overwritten and the GALILEO application is started afterwards

→ The command must be entered without any line breaks.

```
Galileo -nogui -proj="e:\Projects\demo.prj" -builddeploy
-deploytarget=ftp://10.130.23.12/StorageCard -conflicts=1 -startgrs
```

Command Prompt, on the PC, will show the individual build and deploy steps:

Transferring recipe<s> and user management ...

Creating directory '/StorageCard\data' ...

```
Transferring file 'RECIPE.tab' …
Transferring file 'RECIPE.act' …
Transferring file 'xv-100-demo.Pwm' …
Transferring file '_checksum.dmp' …
Restarting GRS …
Transferring file 'RECIPE.tab' …
Disconnecting from host '10.130.23.12'…
Disconnected
Download of project finished
```

Batch file for building and deploying without overwriting all files and then starting the GALILEO application

```
cd C:\Program Files (x86)\Eaton\Galileo 10.2
galileo -proj=C:\Users\name\Documents\GALILEO\GALILEO_
BEISPIELE\G10_1\menubutton\menubutton.prj -nogui -builddeploy
-deploytarget=ftp://192.168.200.99/InternalStorage -startgrs
pause
```

18.2 Deploying a project for use in serial production operations

Serial production operations often require for the exact same GALILEO project to be loaded onto multiple devices. However, it is possible for different devices to have to communicate with different PLCs on the network.
In other words, it would be necessary to build the exact same project individually for each device, except the relevant IP address would need to be changed in *Device\PLC Communication\PLC Parameters\IP Address or Hostname*.

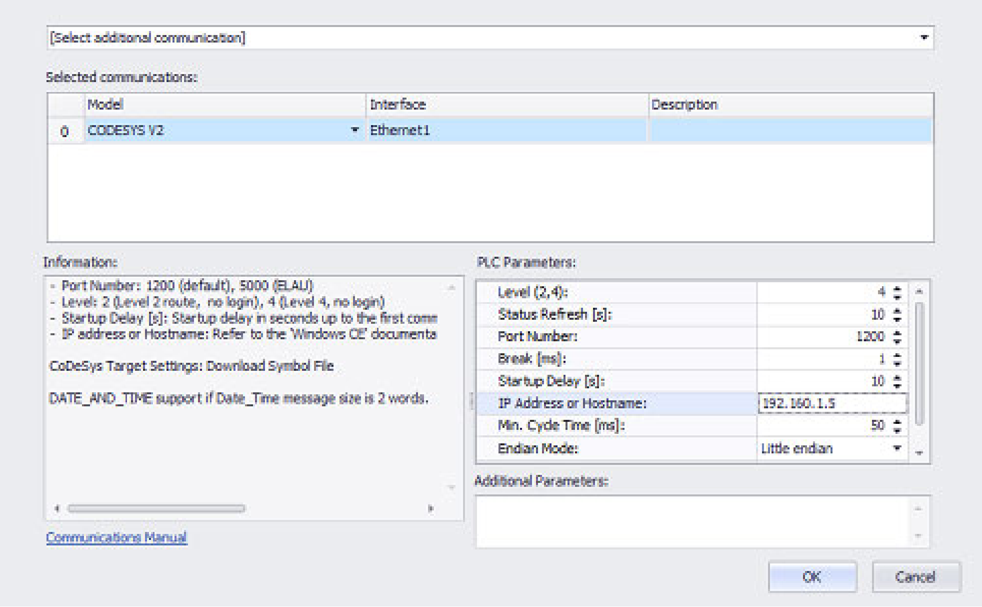
The following method would be appropriate for this specific serial production scenario:

You would build the GALILEO application and then save it on the PC in an output directory. During the compilation process, the <comm.ini><comm.ini> file would be generated in the [\<Name of the current project>\APPL output directory. The contents of this file would correspond to the configuration of the PLC parameters for a communication system. They would also include the device's IP address or hostname. The GALILEO application would finally be loaded onto each device together with the <comm.ini> file after changing the IP address or hostname as appropriate for the relevant device.

Sample contents of <comm.ini> file for CODESYS V2:

```
; CODESYS V2
[Communication0]
Type =16
Group=1
Model=1
Board=0
Endianness=0
TwistedOrder=0
Alignment=3
SlotOrCom=0
LifeTimeCycle=10
Parameter1=0 ; Port Number
Parameter4=2 ; Level (2,4)
String1=192.168.1.5 ; IP Address or Hostname
Break=1
StartupDelay=10
MinCycleTime=50
```

Device\PLC Communication



[Select additional communication]

Selected communications:

	Model	Interface	Description
0	CODESYS V2	Ethernet1	

Information:

- Port Number: 1200 (default), 5000 (ELAU)
- Level: 2 (Level 2 route, no login), 4 (Level 4, no login)
- Startup Delay [s]: Startup delay in seconds up to the first comm
- IP address or Hostname: Refer to the 'Windows CE' documenta

CoDeSys Target Settings: Download Symbol File

DATE_AND_TIME support if Date_Time message size is 2 words.

PLC Parameters:

Level (2,4):	4
Status Refresh [s]:	10
Port Number:	1200
Break [ms]:	1
Startup Delay [s]:	10
IP Address or Hostname:	192.160.1.5
Min. Cycle Time [ms]:	50
Endian Mode:	Little endian

Additional Parameters:

Figure 617: PLC Selection & Configuration dialog box - parameters for the CODESYS V2 communication system

To use this method for serial production operations, follow the steps below:

- In GALILEO, go to *Home Device\PLC Communication\PLC Parameters*.
- Enter the IP address for the first device into IP Address or Hostname.
- In GALILEO, go to *Home Start\Build and Deploy*.
- There, select a destination directory on your PC.
- Build and deploy the GALILEO application by clicking on the ▶ button.

Both the compiled GALILEO application and the <comm.ini> file will now be found in the destination directory.

- Make sure that there is a connection between the PC and the device.
- Make sure that the FTP server is running on the device.
- Open Windows Explorer on the PC.
- In the address bar, enter the service and the IP address for the device. For example: <FTP:\\192.168.200.99>.
- The window will display the device's directory structure and all its contents.
- Copy the folders and contents that were generated from the destination directory on the PC to the device's boot device. You can use the Config Tool on the device to set the boot device to either <InternalStorage> or <StorageCard>.
- Use a normal text editor to open the <comm.ini> file in the destination directory and change the IP address so that it matches the IP address for the next device.
- Establish a connection between the PC and the next device and transfer the compiled GALILEO 10 application together with the <comm.ini> file.
- Repeat for all other devices.

This procedure makes it possible to take a project that has been compiled once and deploy it to multiple devices by only making changes to the <comm.ini> file in order to reflect the necessary changes to the PLC parameters for the relevant communication system.

There are also other options for transferring the compiled project and the <comm.ini> file to the devices. One would be to store both on an SD memory card. Another would be to use the <autolaunch.inf> file to run a batch file in order to copy the files. In addition, it would be possible to use the <Gui4Batch.exe> file to implement a dialog box and a decision option at runtime for GALILEO. The <autolaunch.inf> and <Gui4Batch.exe> files are described in detail in a different manual; please refer to Windows Embedded Compact 7 System Description.

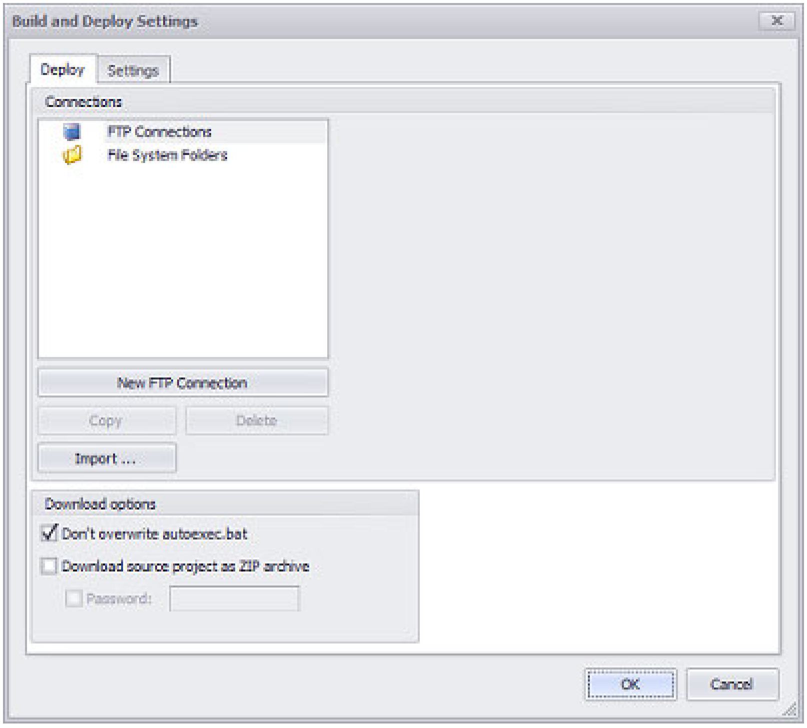
18.3 Configuration for Build and Deploy

The Build and Deploy Settings dialog box is used to define the project's destination directory, as well as how and where the project will be exported as a GRS.

Tab Deploy

This tab can be used to save and view all the connections you have created together with their properties.

Home\Simulator\Build and Deploy settings



☑ Don't overwrite autoexec.bat

☐ Download source project as ZIP archive

Figure 618: Dialog box Build and Deploy Settings

Connections

New FTP Connections / New File System Folder

Clicking on this button will create a new connection. The type of connection created will depend on the option you select in the "Connections" pane, i.e., if you want to create a new FTP connection, you will need to select the FTP Connections node, whereas you will need to select the File System Folders node if you want to create a new filesystem directory.

Copy

Copies an existing connection or directory.

Delete

Deletes an existing connection or directory.

Import...

The Import... button can be used to import existing connections/directories from other projects. After clicking on the button, you will need to select the Galileo project file (.prj) for the project containing the connections/directories you want.

Home\Simulator\Build and Deploy settings

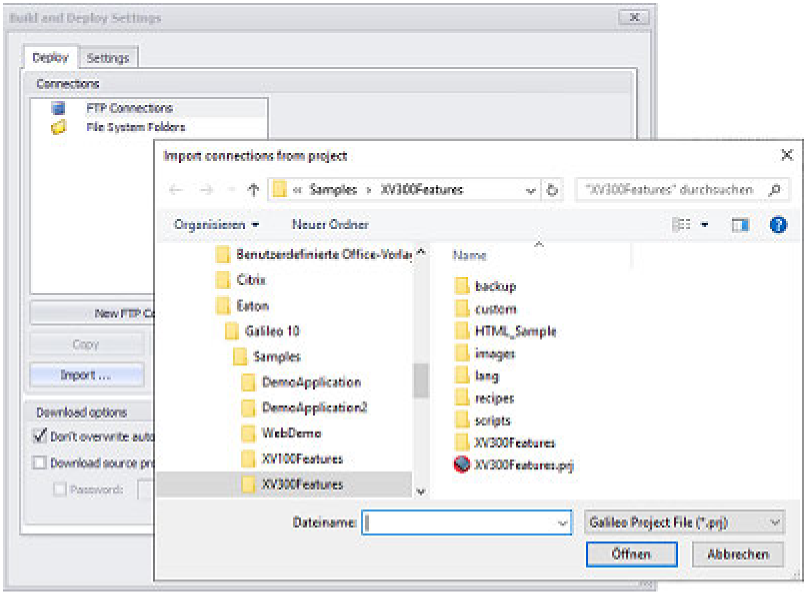
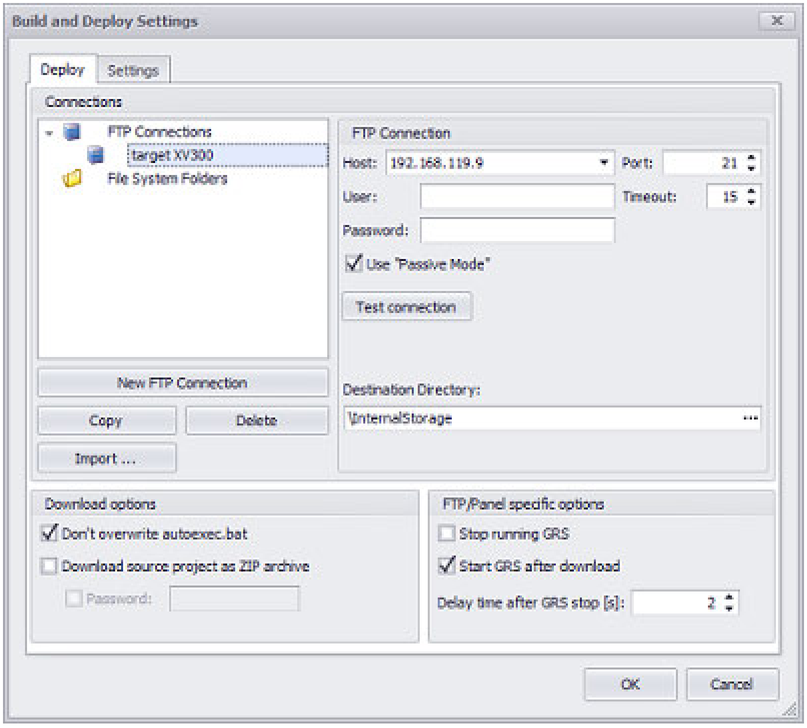


Figure 619: Dialog box Build and Deploy Settings

FTP Connection

As soon as you select an FTP connection in the Connections pane, a section called FTP Connection will appear in the top right area of the dialog box. This section will contain the settings for the FTP connection you selected.

Home\Simulator\Build and Deploy settings



Host: 192.168.119.9

Port: 21

User:

Timeout: 15

Password:

☑ Use "Passive Mode"

Destination Directory: \InternalStorage

☑ Don't overwrite autoexec.bat

☐ Download source project as ZIP archive

☐ Password:

☐ Stop running GRS

☑ Start GRS after download

Delay time after GRS stop [s]: 2

Figure 620: FTP Connection section that appears when a folder is selected under FTP Connections

Host

The target system's (touch panel or PC) IP address
This IP address can be shown or changed on the touch panel by going to:
"Start/Programs/Control Panel/Network/ONBOARD1" or
"Start/Programs/Config Tool/Network/Edit Configuration/ENET1|ENET2"

User

The touch panel's FTP username
The FTP username can be shown on the touch panel, provided it has been configured for the FTP server, by going to:
"*Start/Programs/Control Panel/Owner“* or
"Start/Programs/Config Tool/Advanced Settings/Users“

Password

The touch panel's password
The password can be shown on the touch panel, provided it has been configured for the FTP server, by going to:
"*Start/Programs/Control Panel/Owner“* or
"Start/Programs/Config Tool/Advanced Settings/Users“
The touch panel does not necessarily need to have a password.
If a password has not been set on the touch panel, leave this field blank.

Use "Passive Mode" By default, "Passive Mode" will be enabled for the FTP connection so that data can be transmitted faster. If the FTP connection is blocked by security IT components such as antivirus software or firewalls, this mode can be disabled.

Port The port used for FTP communications.
Default setting: 21

Timeout [s] If the timeout set here elapses before a connection to the touch panel can be established, the system will stop trying to establish a connection to the touch panel.

Test Connection

Clicking on this button will initialize the FTP connection to test whether a connection to the touch panel can be established with the configured settings.

The test is finalized with a test report.

> *CAUTION*
>
> There must be an active FTP server on the touch panel. Otherwise, it will not be possible to establish an FTP connection between your computer and the touch panel.
>
> → The FTP server can normally be shown on the touch panel by going to: *"Start/All Programs/Communication/FTP Server"*

Destination Directory:panelDestination Directory

Use this field to enter the path on the touch panel where you want to load the project as a GRS. The following directory structure will be created in the destination directory:

\APPL	Project file
\BACKUP	Source files of the project
\RUNTIME	GRS files (are automatically updated if necessary, e.g. with new GRS versions)
\DATA	Files are not affected by the transfer of a project (e.g. recipe data)
\OS	Operating System

Directory on the panel

- */StorageCard* if you want the runtime program to be run from an SD card.
- */InternalStorage* if you want the runtime program to be run from internal (flash) memory.

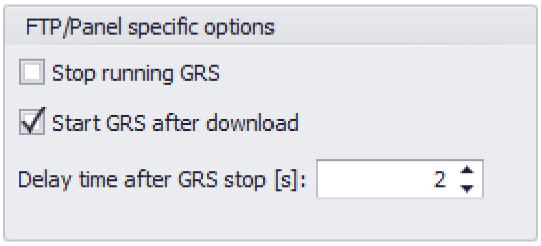
Button ...

Clicking this button establishes the FTP connection with the above settings.. The directory structure of the device is displayed and the destination folder can be selected if the connection can be created.

FTP/Panel specific options

These options will only be shown if an FTP connection is selected.

Home\... button in Build and Deploy group\Deploy tab\FTP/Panel specific options section



FTP/Panel specific options

□ Stop running GRS

☑ Start GRS after download

Delay time after GRS stop [s]: 2

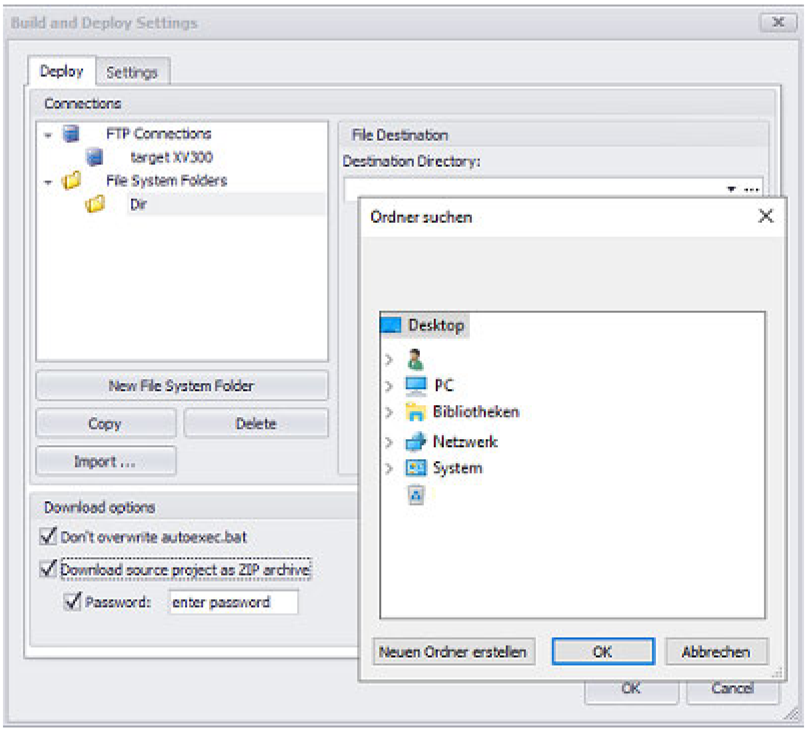
Figure 621: FTP/Panel specific options section

Stop running GRS	The running GALILEO application will be stopped before the download.
Start GRS after download	If the download is completed successfully, the current GALILEO application will start automatically.
Delay after GRS stop [s]	Additional wait time during which the download will be continued after the GRS is terminated on the panel. In certain circumstances, it is possible for the GRS window to already be closed but for required work not to be done yet in the background (e.g., releasing memory). If problems keep happening with downloads around this moment, you can try to slightly increase this wait time.

File System

If the File System Folders node is selected in the Connections pane, the New File System Folder button will appear underneath.

Home\Simulator\Build and Deploy settings



Download options

☑ Don't overwrite autoexec.bat

☑ Download source project as ZIP archive

☑ Password: enter password

Figure 622: File Destination section

As soon as you create a new File Destination folder or select an existing one, a new section will appear to the right.
This section will contain the settings for the filesystem directory you selected.

Destination Directory

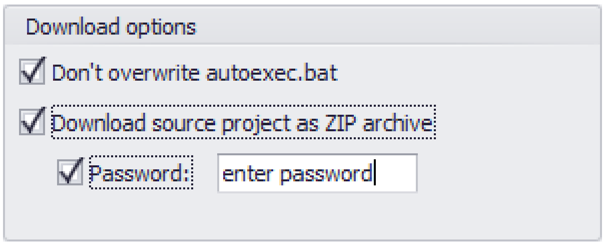
Used to enter the destination directory on the PC. You can use this field to enter the file path you want directly or click on the ... button to open the Browse For Folder dialog box.
The GRS will then be saved in this destination directory when you use the Build and Deploy command.

Download options

The various download options are found in the bottom left section of the Build and Deploy Settings dialog box. These options affect the way GRS data will be downloaded.

The options in this section will apply to all existing connections, i.e., both to all FTP connections and local connections.

Home\ Simulator\Build and Deploy settings



Download options

☑ Don't overwrite autoexec.bat

☑ Download source project as ZIP archive

☑ Password: enter password

Figure 623: Window Download options

Don't overwrite autoexec.bat

When the project is transferred, all files in the "appl\", "backup\", "custom\", "data\", and "runtime\" paths on the touch panel will be deleted. The 'backup\' directory will only exist if the source project is found on the touch panel.
This is only necessary if a new project is transferred. In this way files that are old or no longer required are removed, thus avoiding the possibility of any conflicting data.

Download source project as ZIP archive

All of the project's source files will be packed into a ZIP file that will then be transferred to the touch panel. This ZIP file will be found in the device's destination directory, in the \BACKUP subdirectory, please refer to → section "Directory on the panel", page 1059 as well.

Password

You can protect this file from access and alteration by third parties by using a password.
The dialog to input the password is shown at start the download.
Enter the required password into the field.

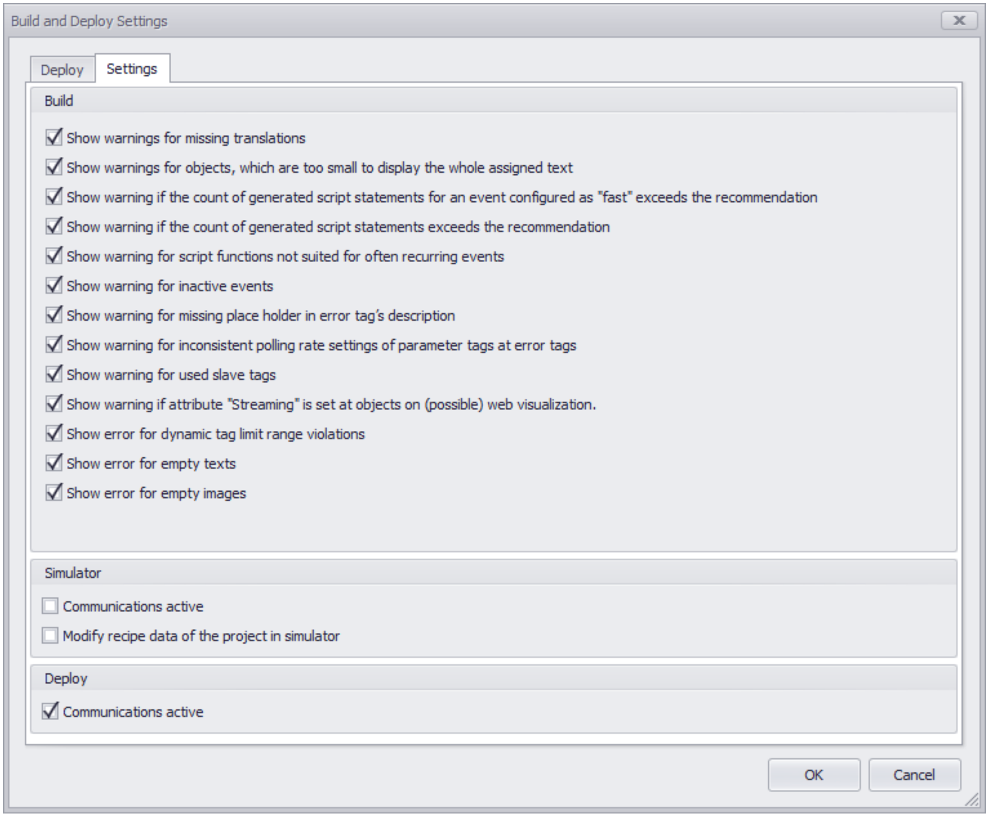
If the following discrepancies are found between the files on the PC and the files on the device, a prompt will appear asking whether you want the corresponding files to be copied to the device:

- Recipe files (.TAB, .ACT) from the "DATA" subdirectory
  These recipe files will only be found on the PC if they have been created with the "Recipe Editor" or transferred to the PC with Upload Data (Local/FTP).
- User management data (.PWM) from the "DATA" subdirectory
  This data will overwrite the password settings on the touch panel.
- The operating system version on the PC and the operating system version on the device are different from each other.
  If the operating system is loaded onto the device, the device will be automatically rebooted.
- The device's destination directory does not contain the same GALILEO project.

Tab Settings

This tab contains options for compiling, simulating and deploying the project.

Home\Simulator\Build and Deploy settings



Build and Deploy Settings

Deploy | Settings

Build

☑ Show warnings for missing translations

☑ Show warnings for objects, which are too small to display the whole assigned text

☑ Show warning if the count of generated script statements for an event configured as "fast" exceeds the recommendation

☑ Show warning if the count of generated script statements exceeds the recommendation

☑ Show warning for script functions not suited for often recurring events

☑ Show warning for inactive events

☑ Show warning for missing place holder in error tag's description

☑ Show warning for inconsistent polling rate settings of parameter tags at error tags

☑ Show warning for used slave tags

☑ Show warning if attribute "Streaming" is set at objects on (possible) web visualization.

☑ Show error for dynamic tag limit range violations

☑ Show error for empty texts

☑ Show error for empty images

Simulator

□ Communications active

□ Modify recipe data of the project in simulator

Deploy

☑ Communications active

Figure 624: Settings tab

Build

These options are used to manage the events that trigger a warning.

Show warnings for missing translations

If this option is enabled and the project is compiled, warnings will be generated if there are any texts that are not available in all the defined languages.

Show warnings for objects, which are too small to display the whole assigned text

If this option is enabled and the project is compiled, warnings will be generated if there are any texts, in objects, that are too long to be displayed fully. All texts in all the defined languages will be checked.

Show warnings if the count of generated script statements defined as "cyclical, fast" exceeds the recommendation

An excessive number of statements in a script that is run cyclically and fast can result in a drop in performance. If this option is enabled, warnings will be generated during compilation if the number of statements exceeds the recommended number for the relevant device model.

Show warnings if the count of generated script statements exceeds the recommendation

While a script is being executed, Galileo will remain locked (not operable).
In order to avoid excessively long lock times, every panel has a recommended maximum number of script statements. If this limit is exceeded, a compiler warning will be shown. This option can be used to disable this compiler warning.
Disabling this warning is a good idea when, for example, a single

	script is used for multiple use cases and, accordingly, the statements are subdivided into multiple conditional statements ("if – else").
Show warnings if script functions that are not suitable for frequently recurring events are used	In the case of certain script functions, quickly running them again is not recommended, as this may result in drops in performance in certain circumstances. If this option is enabled, warnings will be generated during compilation if this type of function is used in events that are configured accordingly.
Show warning for missing placeholders in description texts for error tags.	If a parameter tag is defined when defining an alarm, but the alarm text does not have a placeholder for the value, a warning will be output if this option is enabled.
Show warnings for inconsistent polling rate settings for parameter tags of error tags.	If this option is enabled, warnings will be generated when compiling the project if the polling rate settings for the error tags and the corresponding parameter tags are inconsistent. In order to ensure data integrity, the polling rate for the parameter tags should always be set to "as fast as possible" if possible. For more information, please refer to: → section " Error Tags", page 877.
Show warnings if attribute "Streaming" is set at objects on (possible) web visualization.	The attribute/behavior "Streaming" (e.g. at the Button control) should not be used on a web visualization. Because of the nature of blocking, synchronous behavior of streaming, it is not suited to be used in a web visualization and can lead to serious problems or unexpected behavior. If this options is set, a compiler warning will be generated, if a control has the attribute "Streaming" set and it is likely that the appropriate screen is part of a web visualization (e.g. part of the web visualization group "Local"). For pure web visualizations groups, the warning will always be generated and can't be disabled.
Show error if the value ranges of dynamic limits are exceeded or fallen below.	As described in the section on error tags, → section " Alarms", page 859, a placeholder can be inserted into any text so that it will be dynamically filled out (with a tag value or text) when the corresponding error occurs. In the case of projects with multiple languages, the corresponding text must contain the placeholder in all languages. If it does not, a compiler warning will be output. This option can be used to disable this compiler warning. Disabling this warning is a good idea while developing a new project for which all languages have been created but not all text translations have been completed.
Show error for missing texts.	If this compiler option is enabled, a compiler error will be shown if text objects have been placed on screens but no text has been assigned to them.
Show error for missing images.	If this compiler option is enabled, a compiler error will be shown if image objects have been placed on screens but no image has been assigned to them.

Simulator settings

Communications active	If this option is enabled, the simulation will attempt to communicate

with the PLC.

Modify recipe data of the project in simulator

If this option is enabled, changes to recipes made in the simulator will be automatically applied to the project after the simulator is closed. Accordingly, this option can come in handy in order to test recipe data under simulation conditions and then apply it to the project.
This option should remain disabled if you are using the simulator solely for testing purposes and do not want for the project data to change as a result of this testing.

Deploy

Communication active

If this option is enabled, the Galileo project will attempt to communicate with the configured PLC after being transferred to the target device and starting.
Disabling this option makes sense if there is no PLC / a PLC is not connected or if you want to test the project on the target device without live PLC data.
This ensures that - if the 'Start GRS after download' option is enabled in the Deploy menu - the runtime on the panel starts the first time after the project transfer with disabled communication.
In this case, error messages related to there not being any PLC communications will not be displayed.
As soon as a runtime restart, panel reboot or other runtime start is executed, the runtime starts again with activated communication.
The deactivated option therefore only has an influence on the first automatic start of the runtime after project transfer.

Build and Deploy drop-down menu

You can use the Build and Deploy drop-down menu to select and run individual actions. The actions that will be available for selection will depend on whether an FTP connection or a folder has been selected as the target. All actions will refer to the selected target, e.g., Demo Download local.

Home\Build and Deploy

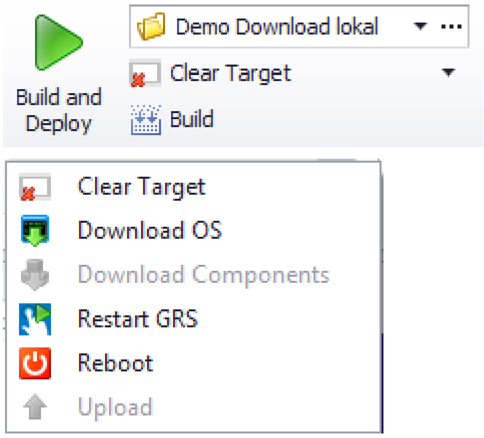


Figure 625: Build and Deploy with the Clear Target option and the full drop-down menu.

Clear target

The following GALILEO directories will be cleared:

- \APPL
- \CUSTOM
- \DATA
- \RUNTIME

→ If the recipe data and user management data for the GALILEO project is found in a directory other than DATA, you will need to delete this data manually.

Download OS

This option will only be available for devices with the WinCE5 or WEC7 operating system.

The current operating system will be loaded onto the device and stored in the selected boot device's \OS directory.

Download components

This option will only be available for devices with the WinCE5 or WEC7 operating system.

The components of the current operating system specified in *Home\CE Configuration* will be loaded onto the device. For a description of the CE Configuration dialog box, please refer to the→ " CE Configuration", page 228 section

GRS Restart

This option will only be available for devices on which the FTP server is running so that they are connected to the PC or to another device with the FTP protocol.

If the GALILEO application is already running, it will be terminated and started again without any user prompts.

Reboot

This option will only be available for devices with the WinCE5 or WEC7 operating system.

This option will only be available for devices on which the FTP server is running so that they are connected to the PC or to another device with the FTP protocol.

The operating system will be rebooted

Upload

This option will only be available if the GALILEO project contains at least one recipe or user management data.

By default, the device will upload the recipe data and the user management data from the \DATA directory into the GALILEO project. For a detailed description of the location where the \DATA directory is found, please refer to → section "Directory on the panel", page 1059

18.4 GALILEO Open

Galileo Open is a PC-based visualization program consisting of the GALILEO Runtime System (GRS).

To select Galileo Open as a communication system, select *Home\Panel Type\Series\Galileo Open*.

When using Galileo Open, you can select any screen size you want. Moreover, you can create projects that will take up either the entire screen or only part of it.

The following are the specifications regarding the maximum possible resolution:

X Resolution: 10 – 6000 pixels

Y Resolution: 10 – 6000 pixels

The runtime program is used by the GALILEO Runtime System (GRS), while the Automatization Software GALILEO program uses the source project.

After being built with GALILEO, these files need to be transferred to the device. To achieve this, the program generates an executable file after a project is successfully compiled so that this file can be transferred to the touch panel. → section " CE Configuration", page 228

If Galileo Open is selected as a communication system and the compilation process is run, GALILEO will generate a directory with all the files required in order to be able to start the project on a Windows computer.

If, for example, a project called **TEST_GalileoOpen.PRJ** is generated in the **D:\DATA\GALILEO\TEST_GalileoOpen** directory, the following directory structure will be found after the compilation process is complete:

```
D:\DATA\GALILEO\TEST_GalileoOpen
           \TEST_GalileoOpen
                      \appl
                      \custom
                      \data
                      \runtime
```

The *\runtime* directory will contain the **GRS_WIN32.exe** file, which needs to be started on the target system.

GALILEO has several options for transferring your project files: FTP, locally, or using a network drive.

If you decide to transfer the files to the target system manually, make sure to only copy the following directory structure:

```
    \TEST_GalileoOpen
             \appl
             \custom
```

\data
\runtime

Once the files are on the target system, you can run the **GRS_WIN32.exe** file by double-clicking on it, by using a shortcut on the desktop, or by entering it into the list of startup programs.

→ Please run the runtime system as an administrator; otherwise, the license will not be stored in Windows 7 or higher.

GALILEO application autostart

If you want the GALILEO application to automatically run at startup on the PC, create a shortcut to the **GRS_WIN32.exe** file in the Windows startup (autostart) folder.

Starting in a window without a frame

When the GALILEO application is running, a window with a visible frame will appear on the screen by default. If you want the application to start in a window without a frame instead, follow the steps below:

- Go to *Project Tree\Scripts\Script* and use the context menu to create a new script, e.g., <StartwithoutFrame>.
- Enter the following line:
  `System.ShowFrameOnOff(OFF);`
- Go to *Home\Settings\Runtime tab* and, in the Startup Script drop-down menu, select the <StartwithoutFrame> script.
- Click on OK.

The next time the GALILEO application is run, it will start in a window without a frame.

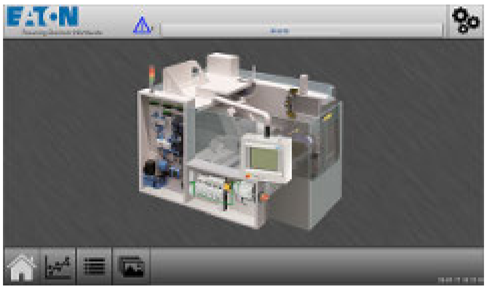


Figure 626: Application in window without a frame

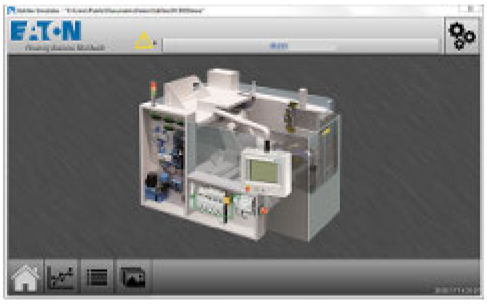
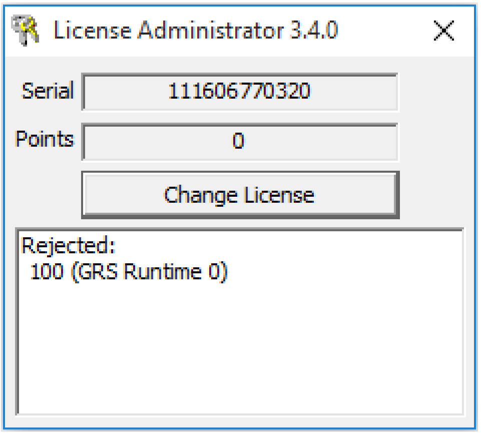


Figure 627: Application in window with frame

18.4.1 Activating the GALILEO OPEN communication system with a license

Once you have saved the GALILEO OPEN Galileo Open directory structure on a PC and run the **GRS_WIN32.exe** file for the first time, the License Administrator dialog box will appear.



Serial 111606770320

Points 0

Rejected:
100 (GRS Runtime 0)

The Serial field will show a 12-digit serial code. This serial code is tied to the PC on which you just started the runtime system.

→ When you enter the activation key for GALILEO OPEN, the computer must have the exact same hardware it will have when it is later used to run GALILEO OPEN. Adding or removing any hardware (docking station, hard drive, etc.) may result in the activation being deactivated, after which GALILEO OPEN will only run in DEMO mode.

To activate GALILEO OPEN with a license, follow the steps below:

- Order an **LIC-GALILEO-OPEN-PC** license product certificate for GALILEO OPEN. If you want to run GALILEO OPEN on an Eaton XP device, you will need to order an **LIC-GALILEO-OPEN-XP** license instead.

The license product certificate will be sent to you by e-mail/postal mail.

- Run the **GRS_WIN32.exe** file as an administrator.
- Write down the 12-digit serial code shown in the License Administrator dialog box. This code is PC-specific.
- Enter the serial code into the ID number field on the license product certificate.
- Fill out the license product certificate completely and send it to the specified address by e-mail.
- You will receive an e-mail containing an activation code.
- Click on the Change License button.
- Use the keyboard that appears to enter the activation code into the field.
- Start the **GRS_WIN32.exe** file again so that the licensed GALILEO application will be active.

GALILEO OPEN is now activated with a license for this device.

19. Web visualization

GALILEO features a service that is automatically loaded onto the touch panel together with the GALILEO application.

Device tab Web visualization

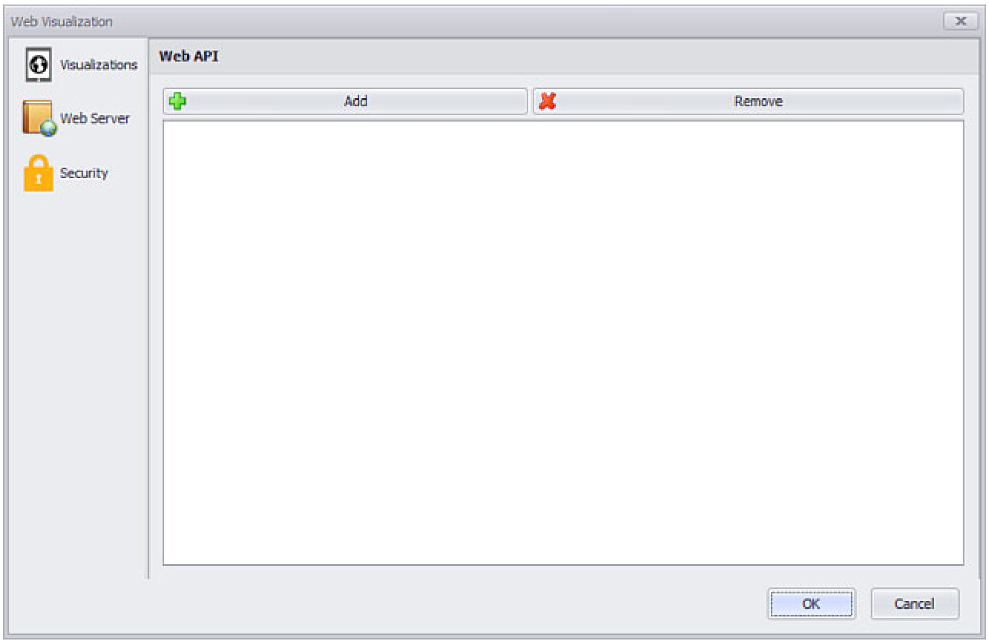


Figure 628: Pane Web visualization

Visualization	Option for the touch panel in the XV300 series, the XP500 industrial PCs and GALILEO OPEN applications Can be used to access a web service that makes it possible to make an existing project available not only on a local device, but on other devices on the Web as well Described in section: → section "Creating a visualization for devices", page 1080
WebServer	
Security	
Connections	Option for the touch panel in the XV300 series, the XP500 industrial PCs, and GALILEO OPEN applications Configuration settings for establishing a connection to the web service through the user program / project Opens the PLC selection & Configuration dialog box Described in section: → section "Settings for connection", page 1089

This web service makes it possible to display and control the GALILEO application not just locally on the device, but also on other devices or PCs with the use of web

clients.
A maximum of five web visualizations can be active simultaneously in addition to the local visualization.

→ Please note that this web server is only available for the following operating systems:

- WEC 7.0 (XV300)
- Win32 (XP500)
- Windows7, Windows10 GALILEO OPEN

This service cannot be used with CE 5.0 (XV100/400).

Browser

In order to display the web clients, the browser used must support HTML5. The web clients support the following browsers:

- Safari desktop and mobile
- Chrome desktop and mobile, as well as mobile version for Android devices, iPhone and iPad

→ It is not guaranteed that the web clients will work correctly with other browsers such as Opera, Edge, Mozilla Firefox and Internet Explorer.

PLC tags are managed by the relevant PLC or with the *Home\Gateway* functionality. Meanwhile, internal tags can be shared. For a more detailed description, please refer to → section "Shared", page 574.

In the case of system tags, each individual system tag comes with web visualization sharing enabled or disabled already.

19.1 Use cases

The various use cases can be selected by selecting the appropriate visualization mode.
The type of visualization can already be selected when creating a new project and can be adjusted later by going to *Device\Visualization*.

The following groups are available:

1. ▬ Local;
   GALILEO application used only locally on the device
   This <Local> group cannot be removed or renamed, but it can be disabled
2. 🌐 Visu;
   GALILEO application for web client
   The web visualization from this <Visu> group serves multiple web clients

Figure 629: Local: Application on the touch panel

The <Local> group cannot be removed or renamed, but it can be disabled as soon as a <Visu> group for the web visualization is added and configured.

Various visualizations can be created and managed for each group.

Visualizations can be added in two ways for each group of visualizations (1 x <Local> or n x <Visu>):

- Linked visualizations
- Unlinked visualizations

Linked visualizations

Linked visualizations are created by copying from an existing visualization group. When you do this, the following will be copied:

Visualization group name
URL suffix for access to the visualization
Assigned bit tags
Password (optional)
Configured start screen for the visualization
Visualization status: Enabled / Disabled

Copied visualizations will always be linked to each other and will share the following:

- Resolution
  All linked visualizations will have the exact same resolution.

- All screens
  If screens or screen contents are added / deleted, the change will apply to all linked visualizations.

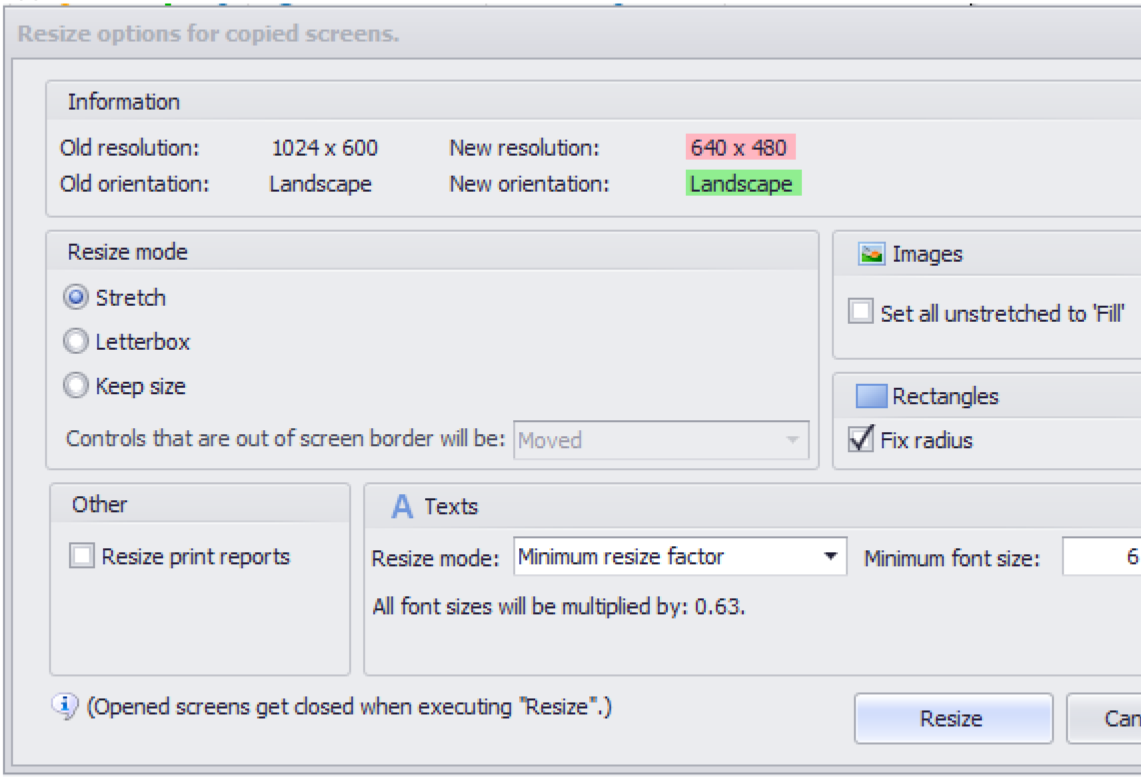
If a user changes the resolution in a visualization, for example, the following will happen:
- The size of all screens will be changed
- The size of all shared screens will be changed
- Other linked visualizations will also have this new resolution

The start screen is a completely independent property. In fact, you can set any start screen you want for the visualization that is currently active.
To do so, use the "Start screen" setting for the visualization or right-click on a screen inside the project tree to open the context menu and click on Set as startup screen.

Unlinked visualizations

Unlinked visualizations are created by simply adding them with the ✚ Add Group button.
When you do this, an empty standard screen that only belongs to the new group will be created.
If you want to add screens from other visualizations, you can do so by opening the context menu in the project tree and using the Copy and Paste options.

If the resolution is different, a dialog box for making the necessary adjustments will appear.



Resize options for copied screens.

Information

Old resolution: 1024 x 600 New resolution: 640 x 480
Old orientation: Landscape New orientation: Landscape

Resize mode
● Stretch
○ Letterbox
○ Keep size
Controls that are out of screen border will be: Moved

Images
□ Set all unstretched to 'Fill'

Rectangles
☑ Fix radius

Other
□ Resize print reports

Texts
Resize mode: Minimum resize factor Minimum font size: 6
All font sizes will be multiplied by: 0.63.

(Opened screens get closed when executing "Resize".)

Figure 630: Resize options for copied screens dialog box

If you need to make size changes for individually differing resolutions, you will only be able to do so with the visualization group. This means that, if necessary, you will need to create a separate group for the display size / resolution.

The unlinked group of visualizations requires the following in Device\Visualization:

Definition of resolution
Name of group

And for each individual visualization within the group:

Name of visualization
URL suffix for access to this visualization
Assigned bit tags (optional)
Password (optional)
Configured start screen
Status: Enabled / Disabled

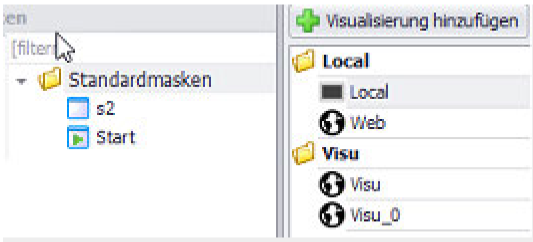
Since the screens in unlinked groups of visualizations are independent,
the names and IDs of the screens can be identical.
In other words, the "Start" screen in the local visualization can have an <ID 0>, and it can have the same ID
in a web visualization.
This makes it possible to use a single script or PLC program for multiple groups of visualizations.

Linked / unlinked within a group

Within a group of visualizations, all visualizations are linked to each other and are managed in this group.
Screens are managed within the group, i.e., even if you only need one visualization, a group needs to be created.

Each group saves its screens in an unlinked area.

Structure example



There are two groups: The <Local> group and the <Visu> group

Standard screens <s2> and <Start> can be used in the <Local> group.

If you wanted to, you could create a <Start> or <s2> screen for the <Visu> group, and these screens would be completely independent from their counterparts in the <Local> group.

Summary

- The name of each group and visualization must be unique, meaning it can only be used once.
  However, a group can have the same name as a visualization.
- All the visualizations in a group have the same priority, i.e., there is no "master" visualization.
- Visualizations cannot be moved between groups – they can only be copied from one to the other.
- You can make as many copies of a screen between the groups as you want with copy & paste (Ctrl+C/Ctrl+V).
  If the resolutions differ, the corresponding resize options will be displayed.
- You can create new visualizations by copying other ones.
- A new visualization can place limits on a new group, e.g., if there are varying resolutions.
- The resolution is set when the group is created, and can be changed for the entire group later on in the Visualization dialog box.
- The resolution for the Local group and the visualizations linked to it can only be changed by selecting a different Panel Type in the Device tab, as this resolution depends on the target device for the project.

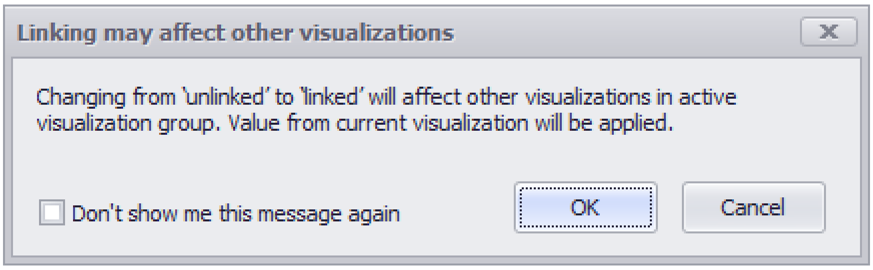
19.2 Global user management settings

In GALILEO, global settings affect all visualizations in a number of ways. You can configure these global settings for the various visualizations independently of each other by going to Home\Project Configuration\User Management, opening the General tab, and then going to the Login/Logout section.
The corresponding behavior will be slightly different for linked and unlinked visualizations.

🔗 Linked
If you create a new visualization, it can be linked to other linked visualizations. This will be indicated in the Editor with the 🔗 icon.

If you make changes to a linked property, the changes will be applied to all the visualizations in the active visualization group.
If the property is not linked, you can make changes to it independently of other visualizations.

The change from "unlinked" to "linked" will affect other visualizations - the value from the current visualization will be applied. Because of this, a confirmation prompt will be displayed.



Linking may affect other visualizations

Changing from 'unlinked' to 'linked' will affect other visualizations in active visualization group. Value from current visualization will be applied.

□ Don't show me this message again

C Ↄ Unlinked
If a visualization is unlinked, this will be shown with the C Ↄ icon in the Editor.

If you create a new visualization, the relevant settings will be set to the default values. This applies to all the application behavior settings configured in the project configuration for the project.

These properties, changes to these settings are completely independent from the settings in other visualizations. When this property is unlinked from the visualization, this will be shown with the C Ↄ icon, i.e., disabled and with the link "broken."

In order for the 🔗 and C Ↄ chain link icons to be displayed, there must be more than one visualization in a group.

This means that the 🔗 and C Ↄ icons make it possible to directly see and edit the status.

The following project tree objects support the linked / unlinked property:

- All confirmation screens (Graph, Recipe)
  Recipe Editor dialog box\Confirmation tab

- Alarm List -> Screen change in the event of Error/Warning/Message
  Alarms dialog box\Handling tab
- User Management:
  - Switch to screen after logoff
  - Changing keyboards for password, username and password
- Tag Help
  Tag Help dialog box

The currently active visualization will be displayed in the quick access toolbar at all times, where it can be selected directly.

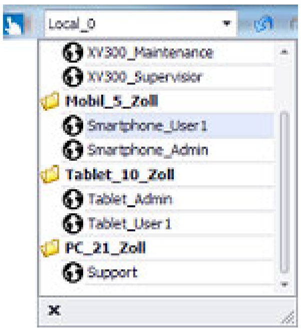


Figure 631: Quick Access Toolbar

If a visualization from the same group is selected, the project view will not change.

If a visualization from a different group is selected, all open screens will be closed and the screens from the selected visualization that were open the last time the visualization – or linked visualizations from the same group – was open will be opened again.

Additional settings need to be configured in order for the visualization to work:

- Visualizations – Creating groups, adding / deleting / enabling / disabling visualizations and defining their properties
- Connection - Used to configure the port, the maximum number of connections, and the encryption settings

19.3 Different screen navigation configurations for web clients and for the local application on the device

The user interfaces for the GALILEO application used locally on the device and the application used in web clients can be implemented and displayed differently within the same project.

For starters, different user interfaces can be implemented by using different startup screens (▶). In other words, a specific startup screen would be set up for the GALILEO application used locally on the device and another one would be set up for the application used in web clients.

In addition, the "IsWeb" system tag can be associated with the visibility of visualization objects and screen changes. By having the accessibility of screen changes depend, for example, on the "IsWeb" system tag, fully separate screen trees can be put together for the local visualization on the device and for the web visualization.
Bit tags can also be used for this purpose.

It is recommended to have different screen navigation configurations for web clients and for the GALILEO application used locally on the device. This way, you can ensure that, for example, users will only be able to trigger process-relevant actions on the actual device.

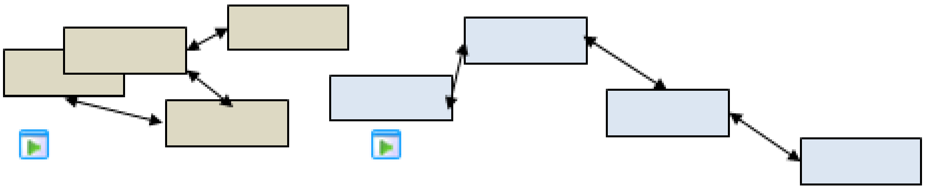


Figure 632: Recommended: Completely different screen navigation configurations for the GALILEO local and web client applications

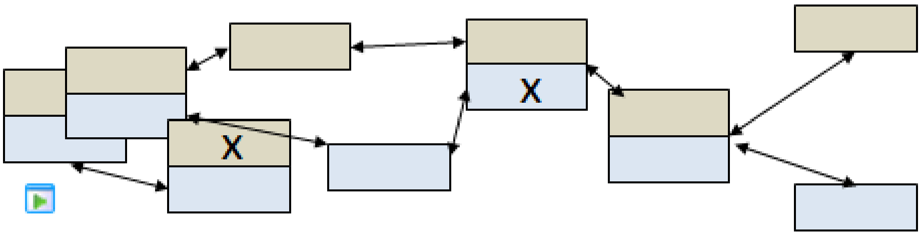


Figure 633: Mixed screen navigation; certain screens are different for the local GALILEO application and for the GALILEO web client application, while others are the same for both

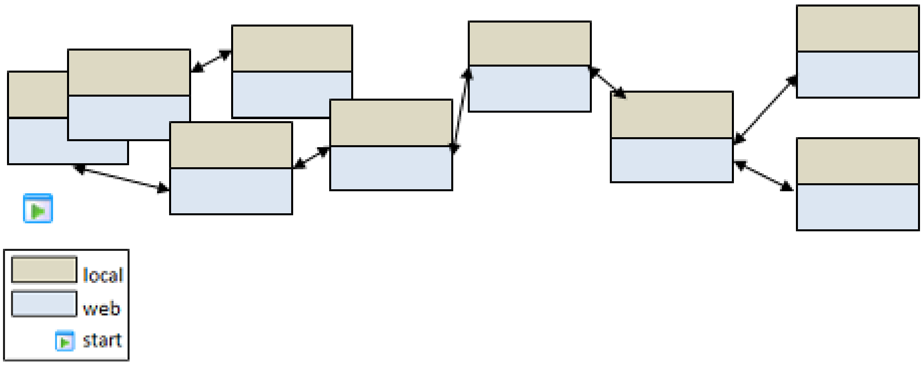


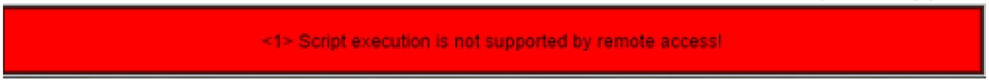
Figure 634: Same screen navigation for the GALILEO application used locally and in web clients

19.4 Web visualization restrictions

Please note that there will be limitations related to functions and tags when the web visualization functionality is enabled.

Available functions

If you configure your GALILEO application with the <Web and local> visualization mode, you will be able to use all functions without limitations in your project and for the local application on the device, but not for the web clients. In other words, you will need to make sure that the screen navigation configurations for the web client and the local application on the device are set up in such a way that the web client will not be able to call the aforementioned functions. If it does call one of these functions at runtime, the call will not be executed and an error message will appear.



<1> Script execution is not supported by remote access!

Figure 635: Example of an error message due to an illegal runtime script call by the web client

Web clients are not allowed to run the following functions:

- Video player
- Webcam
- HTML

Displaying the graph visualization object is permitted, as is running other graph functions such as Scroll to Start.

All recipe functions, as well displaying the recipe visualization object, are permitted.

User management tags

If you want the web client to be able to access the user management settings, you will need to create the user management tags as internal tags. In addition, the checkbox for the <Shared> option must NOT be enabled. Example: Tag used to save the current username.

19.5 Creating a visualization for devices

Option for the touch panel in the XV300 series, the XP500 industrial PCs and GALILEO OPEN applications.

The following sections go over the procedure for setting up a web visualization for the two use cases.
For a description of the use cases themselves, please refer to → section " Web visualization", page 1070.

1. ▬Local;
   GALILEO application used only locally on the device
   This <Local> group cannot be removed or renamed, but it can be disabled
2. 🌐Visu;
   GALILEO application for web client
   The web visualization from this <Visu> group serves multiple web clients

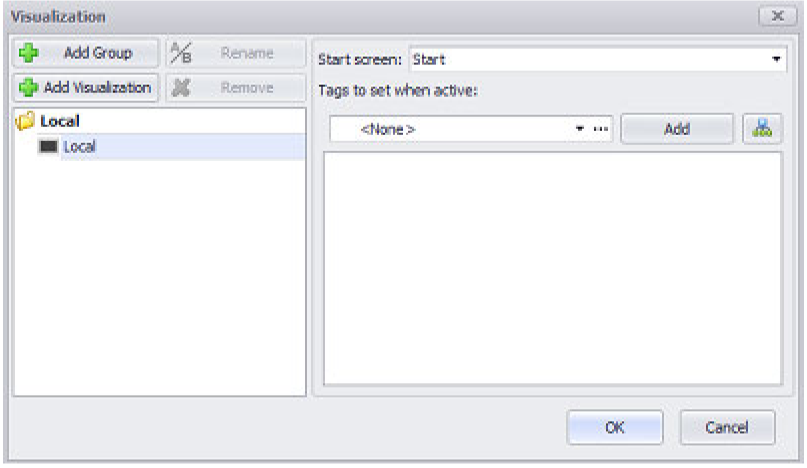
19.5.1 Local

The Local group is a required component of every single application and cannot be renamed or removed. Local visualizations (for a specific group of people, for instance) can be added to this group.

The resolution for the Local group is the display size of the panel model selected for the project and, accordingly, the size of the standard screens.

All visualizations created underneath it will be sized for this screen size.

- Go to *Device\Visualization* and select <Local> as the visualization mode.



Start screen: Start

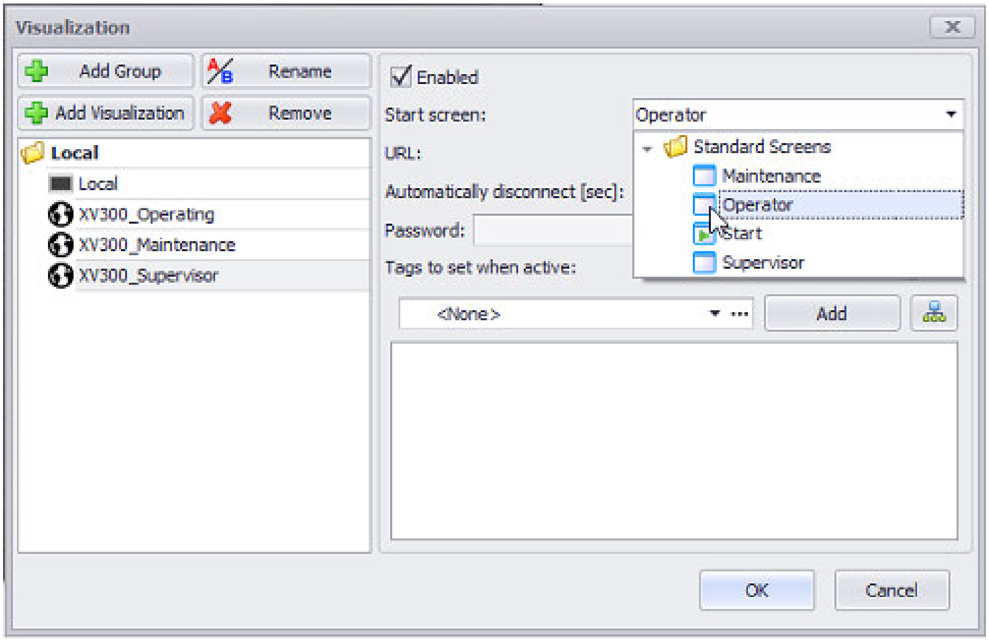
<None>

Figure 636: Visualization dialog box

- In the "Start screen:" field, select one of the existing screens as a start screen.
- Add a Tag to set when active:, such as:< #incFlow>
- Confirm your selection by clicking on OK
- UseBuild and Deploy, for example, to run the Galileo Simulator.

In this case, a web server will not be loaded onto and started on the device. The GALILEO application will start on the device. It will not be possible to access the GALILEO application with web clients.

- Use the Add Visualization button to add a visualization.
- Rename the <Local_0> visualization in line with the intended use, for example.



☑ Enabled
Start screen: Operator
URL:
Automatically disconnect [sec]:
Password:
Tags to set when active:
<None>

Figure 637: WebVisu tab with visualization mode

- You can select the existing visualizations directly in the quick access toolbar.

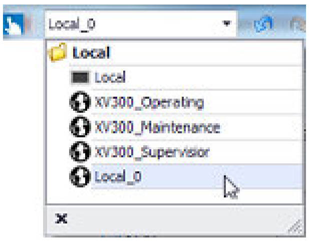


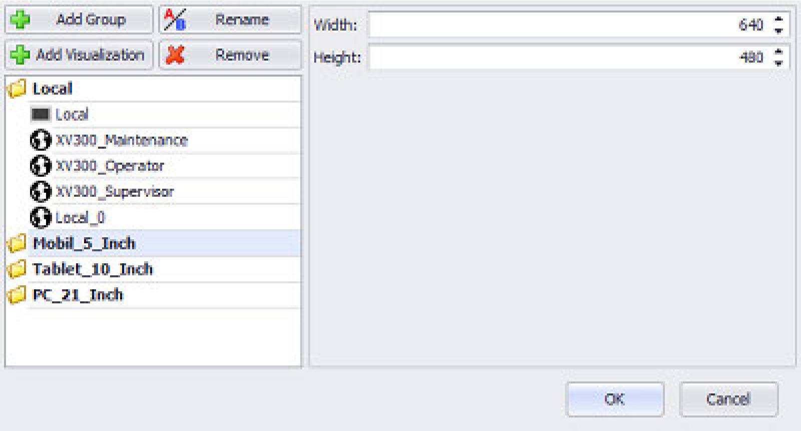
Figure 638: Quick Access Toolbar

- Assign a start screen to each visualization.
  If you do no assign a specific start screen to each visualization, the start screen for the <Local> group will be automatically used for the visualization.
- In the URL: field, enter a URL suffix for access to the visualization.
- If necessary, assign a password for later access to the visualization.
- Add a Tag to set when active:, such as < #incFlow>, in order to be able to control the visualization later on.
- Confirm your selection by clicking on OK
- UseBuild and Deploy, for example, to run the Galileo Simulator.

19.5.2 Visu

- Go to *Device\Visualization* and click on Add Group to add a <Visu> web visualization.

  Different resolutions will require their own groups.
- Rename each of these groups according to the corresponding display device, e.g.: <Smartphone>, <Tablet>, and <PC>.
- Enter the resolution as the width and height in pixels for each of these groups. The size specified here will be applied to the standard screen size.

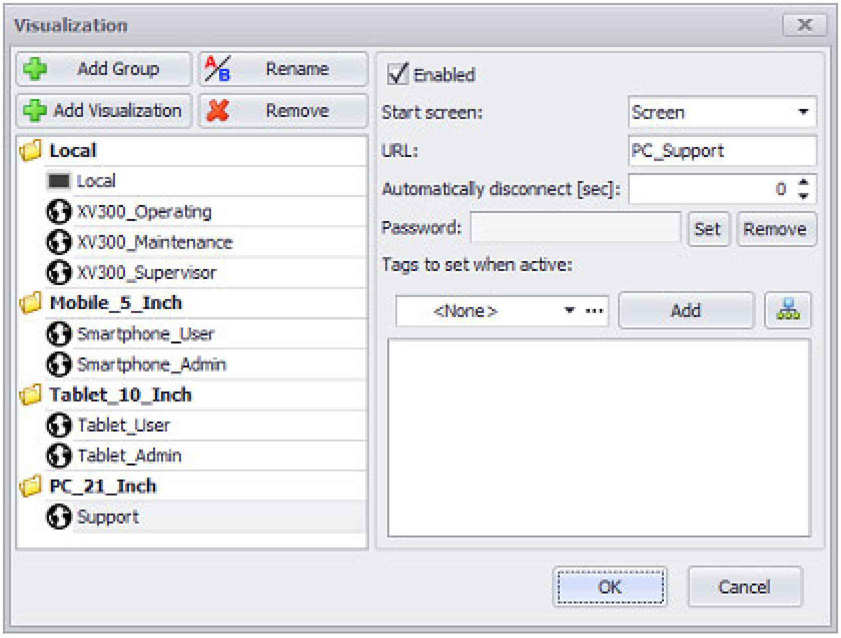


Width: 640

Height: 480

Figure 639: Visualization dialog box with three web visualization groups

- Select a group.
- Use the Add Visualization button to add a visualization.
- Rename the <Visu> visualization in line with the intended use, for example.



☑ Enabled

Start screen: Screen

URL: PC_Support

Automatically disconnect [sec]: 0

Password:

Tags to set when active: <None>

Figure 640: WebVisu tab with visualization mode

- ▶ You can select the existing visualizations directly in the quick access toolbar.

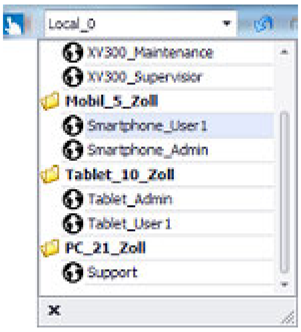


Figure 641: Quick Access Toolbar

- ▶ Assign a start screen to each visualization. If you want the visualization to start with a screen other than that used by the local application on the device, use the Start screen to enter an existing standard screen as the start screen for the web application.
  If you do not assign a different start screen, the group's start screen will be automatically used for the visualization.
- ▶ In the URL: field, enter a URL suffix for access to the visualization.
- ▶ If necessary, assign a password for later access to the visualization.
- ▶ Add a Tag to set when active:, such as < #incFlow>, in order to be able to control the visualization later on.
- ▶ Confirm your selection by clicking on OK
- ▶ UseBuild and Deploy, for example, to run the Galileo Simulator.

You will need to separately create a connection to the device where you want the visualization to be shown. Please refer to → section "Settings for connection", page 1089

As soon as the GALILEO application has been built and deployed onto the device, the application will be started locally on the device and the web server will be automatically started with the corresponding port. The GALILEO icon in the lower right status bar will indicate that the application is running. You will not need to configure any additional settings on the device. Please note that the number of web clients that can be started will be limited to the number specified in Maximum connections (the local application will not count towards this limit).

→ Make sure that the web server is not started twice with the same port!

19.5.3 Start web client

- ▶ To start the web client, open the browser you will be using; please refer to → section "Browser", page 1071

- Enter the device's IP address, followed by the port, into the browser's address bar.
  If you are using the default port, you do not need to add it.
  Example:192.168.200.99:1234

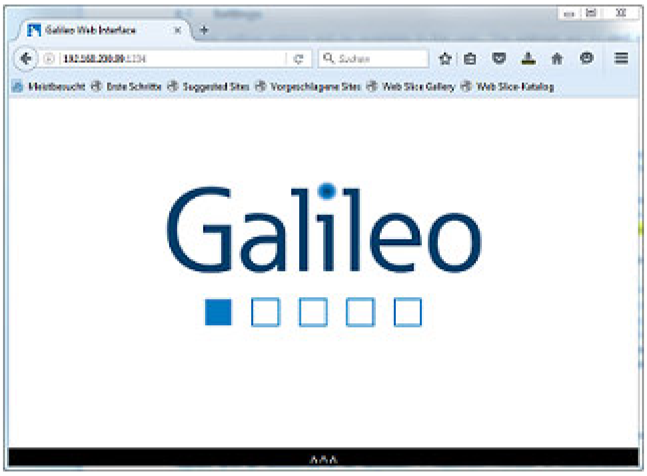


Figure 642: Web client started

If the maximum number of connections is reached, an error message will be shown every time an attempt is made to start a new web client.

Alternatively, you can start the web client as follows:

- Make sure that the GALILEO application is running.
- Make sure that you have set the browser you want in *Home\Options\Web browser*.
- In the quick access toolbar, click on the icon or run the Show in browser command.

The GALILEO application will be shown in the web client.

19.5.4 Simulator and visualization

How will the simulator behave differently for the different visualization modes?

Local – The GALILEO Runtime System will be started and show the user interface for the GALILEO application on the PC. The web server will not be started.

Visu – The GALILEO Runtime System will be started, but the user interface for the GALILEO application will not be shown on the PC. The web server will be started. It will be possible to access the GALILEO web visualization with a browser and localhost.

To start a browser with the web client and access the simulator's web visualization, follow the steps below:

- Select a visualization in the quick access toolbar
- Go to *Device\Connection*
- Enter a port, e.g., <1500>.
- Confirm by pressing OK.
- Start the GALILEO application in the simulator by clicking on ▶.
- Open the browser.
- Enter "localhost", followed by ":" and the port, into the address bar.
  e.g. http://localhost:1500

→ If the connection fails, use a different port.
Make sure that the web server is not started twice with the same port.

19.5.5 Settings Visualization

A visualization makes it possible to display and control the GALILEO application not just locally on the touch panel, but also with a custom configuration (e.g., customized for specific application users) via web clients on other, different devices or PCs as well.
For a detailed description of how this works, please refer to → page 1070.

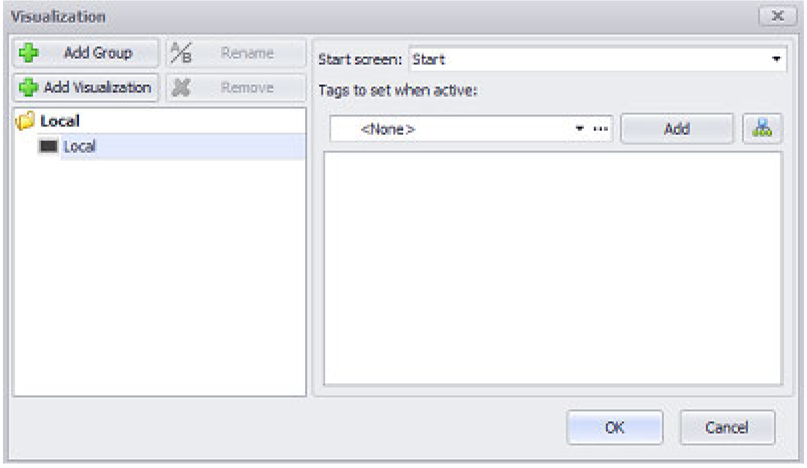
Option for the touch panel in the XV300 series, the XP500 industrial PCs, and GALILEO OPEN applications.

You can use the Visualization dialog box to set up the visualizations for your GALILEO project.

As long as you have not configured a visualization, Local will be set as the default.

A visualization is a group of specific properties such as a start screen, password, tags, and the shared resolution property.

Device tab\ Visualization



Start screen: Start

Tags to set when active: <None>

Figure 643: Visualization dialog box used to configure the visualization mode

Add Group	Adds a visualization
Add Visualization	Adds a visualization to the selected group
Rename	Can be used to rename the selected, existing groups and/or visualizations.
Remove	Deletes an existing group or visualization

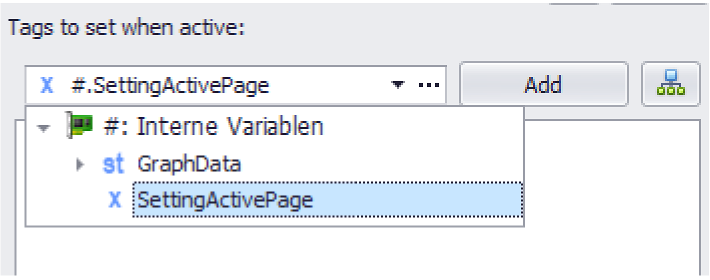
The way individual visualizations are grouped and individual visualizations are added to these groups determines the view for the user in the application.
This means that this feature can be used to distinguish the resolution that should be used, i.e., which device should be used for display. A separate visualization can be / needs to be created for each device and each resolution so that the content will be displayed correctly and access to specific data and views can be managed.

Start screen

Used to select the screen with which the visualization will start.
This setting will not affect the local start screen, i.e., the screen that the GALILEO application will start with when run locally on the touch panel. This local start screen can be set by using the context menu for the *Project Tree/Screens tab*.

Tags to set when active:

Used to assign bit tags
These tags will be set to TRUE (1) if the visualization is active.
These tags must not share local tags.



Add

Once you make a selection from the existing bit tags in the project with ⋯, you can add it with this button.
You can adjust the limits for the tag.

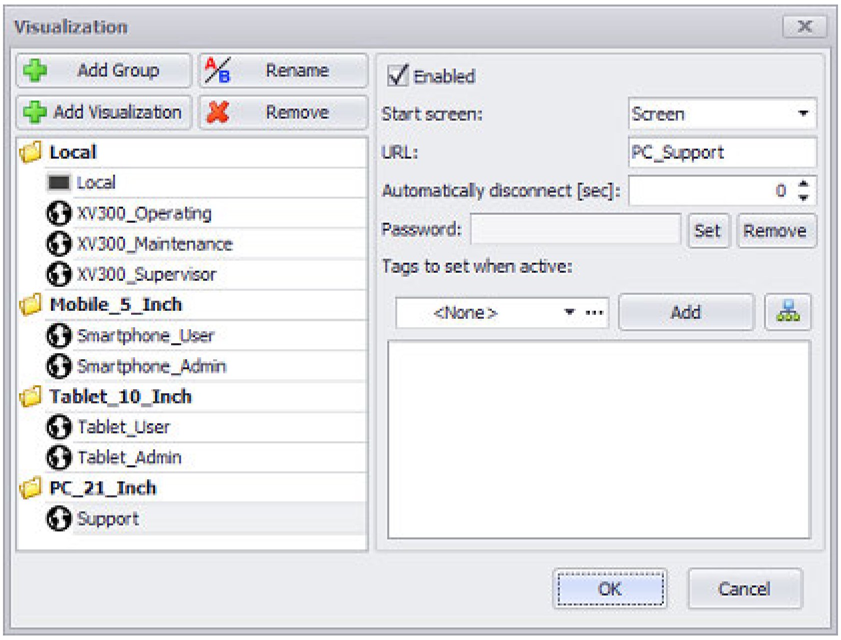
🖧

Used to select tags with the Controls Manager structure

19.5.5.1 Support for multiple resolutions

A different resolution can be defined for each group or each visualization in a group. This ensures that you will always be able to create a screen that matches the end device's resolution.
For each visualization for the various devices, these standard screens can also fulfill various functions within the same project.
These screens need to be created for each visualization, as the screens are completely independent from each other.

Device tab\ Visualization



- Local
  - Local
  - XV300_Operating
  - XV300_Maintenance
  - XV300_Supervisor
- Mobile_5_Inch
  - Smartphone_User
  - Smartphone_Admin
- Tablet_10_Inch
  - Tablet_User
  - Tablet_Admin
- PC_21_Inch
  - Support

☑ Enabled

Start screen: Screen

URL: PC_Support

Automatically disconnect [sec]: 0

Password:

Tags to set when active:

<None>

Figure 644: Visualization dialog box with visualization examples

When ☑ Enabled, the selected visualization will be visible in the application.

Start screen

Each visualization can be assigned its own standard start screen.

URL

URL suffix for establishing a connection. For example: with <PC_Support>, the visualization will be accessible at http://192.168.119.2/PC_Support.

Automatically disconnect [sec]

If the specified time elapses without any user interaction, the connection will be terminated.
0 = No time limit.

Password

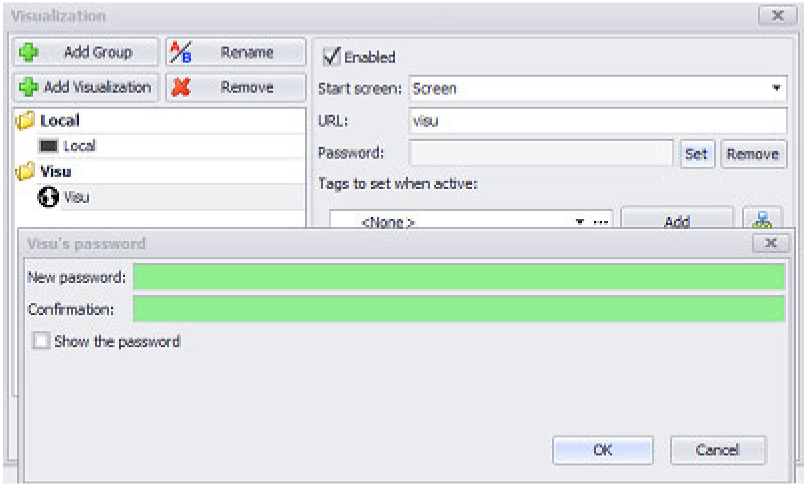
Used to secure data transfers
This password will apply to web visualizations only and, if set, will need to be entered in order for a web client to be able to display the visualization.

Set

Clicking on this button will open a dialog box that you can use to set a new password. If the two fields match, they will be highlighted in green. Otherwise, they will be highlighted in red.

Remove

Deletes the set password



☑ Enabled

Start screen: Screen

URL: visu

Password:

Tags to set when active:

<None>

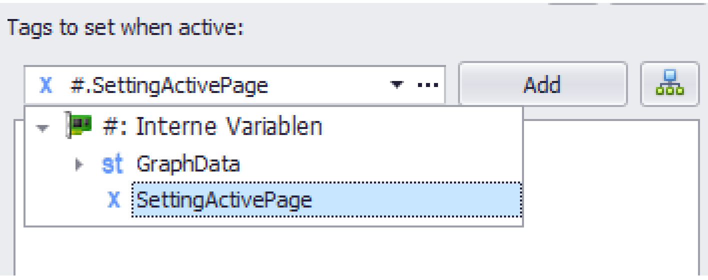
New password:

Confirmation:

□ Show the password

Tags to set when active:

List of the tags selected for the visualization.
Events in the visualization can be controlled later on with the use of this tag.



Tags to set when active:

X #.SettingActivePage

#: Interne Variablen

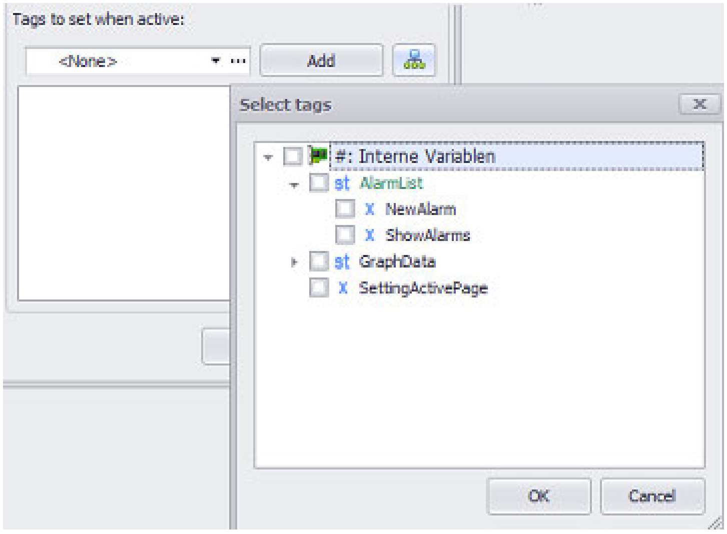
st GraphData

X SettingActivePage

Add

Once you make a selection from the existing bit tags in the project with , you can add it with this button.
You can adjust the limits for the tag.

Used to select tags with the Controls Manager structure



Tags to set when active:

<None>

□ #: Interne Variablen

□ st AlarmList

□ X NewAlarm

□ X ShowAlarms

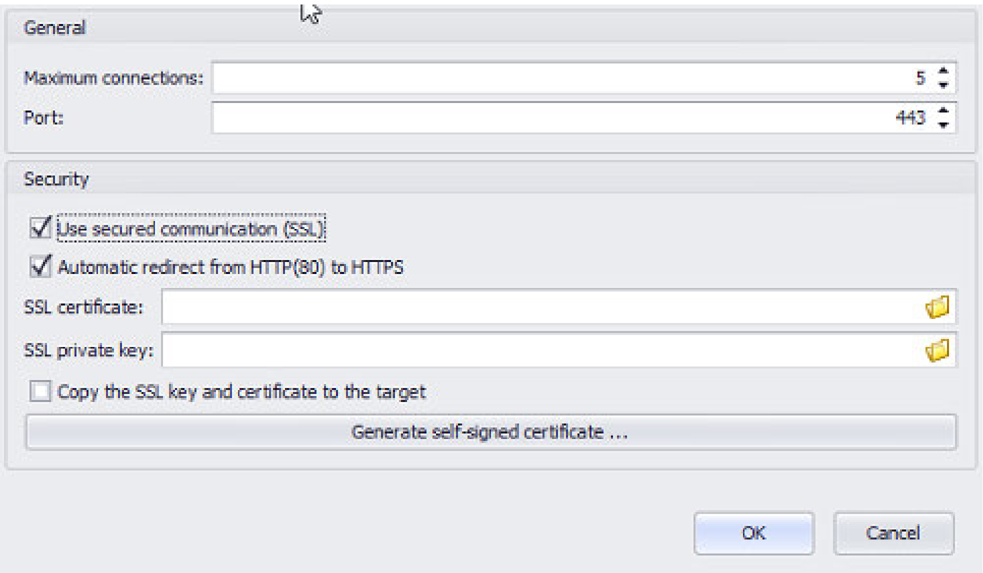
□ st GraphData

□ X SettingActivePage

19.5.6 Settings for connection

For the web visualization, it is necessary to configure a connection through which the web visualization can be reached.

Device tab\ Connection



General

Maximum connections: 5

Port: 443

Security

☑ Use secured communication (SSL)

☑ Automatic redirect from HTTP(80) to HTTPS

SSL certificate:

SSL private key:

□ Copy the SSL key and certificate to the target

Figure 645: Connection dialog box

General

Maximum connections: Used to specify how may web clients are allowed to simultaneously connect to the GALILEO web visualization.
If you enter a value of 2, for example, only 2 different web clients will be able to access the visualization simultaneously.
If an access attempt is made with another web client, an error message will be shown on that client.
The other web clients, as well as the local visualization, will continue to run normally.
The maximum number that can be configured is 5.
The local visualization on the device does not count towards this limit.

Port

Used to specify the network port for data transmission.
The following are the default ports:

- Port 80 for transmission via HTTP
- Port 443 for encrypted transmission via HTTPS

These ports can be changed at any time. When doing so, make sure that the port you select is not already being used. The browser's web client will connect to the web service when the following address is entered into the address bar:
<Protocol>://<IP address of device with web ser-vice>:<Port>
e.g. “http://192.168.119.2:80“
If you are using the default port, you will not need to specify the port in the web browser's address bar. In this case, the request is made, e.g. with
http://192.168.119.2.

Security

Data transfers between the web service and web clients can be encrypted with SSL. SSL is implemented on the device for this purpose. In order to avoid warnings in the browser, a certificate needs to be generated and signed by a CA. This certificate must be stored in a secure certificate repository on the device.

Use secured communication (SSL)

If the ☑ checkbox is enabled, an encrypted connection will be used and the drop-down menu for the certificate will be enabled.

Automatic redirect from HTTP(80) to HTTPS

Can only be enabled if an encrypted connection has been selected.
If the ☑ checkbox is enabled, there will be automatic redirecting from HTTP to HTTPS if the user does not enter "https://" at the start of the address in their web client.

Example

"192.168.200.99" is entered into the web client's address bar.

The web service listens on default port 80. Since no port was specified in the address, default port 80 is used automatically. The web service will now redirect the connection request to HTTPS automatically.

Input for the web browser address bar: 192.168.200.99
Interpretation: http://192.168.200.99:80
Redirect: https://192.168.200.99:443

SSL certificate
Used to select the SSL certificate.
The certificate manager is used to manage the selections available here.

19.5.7 Certificate manager

The certificate manager helps securely manage the certificates in the GALILEO project.

For encrypted connections via SSL, it is necessary for there to be corresponding certificates.

Encrypted connections can be needed for the web visualization, for an OPC UA connection, etc.

Device tab\ Certificates

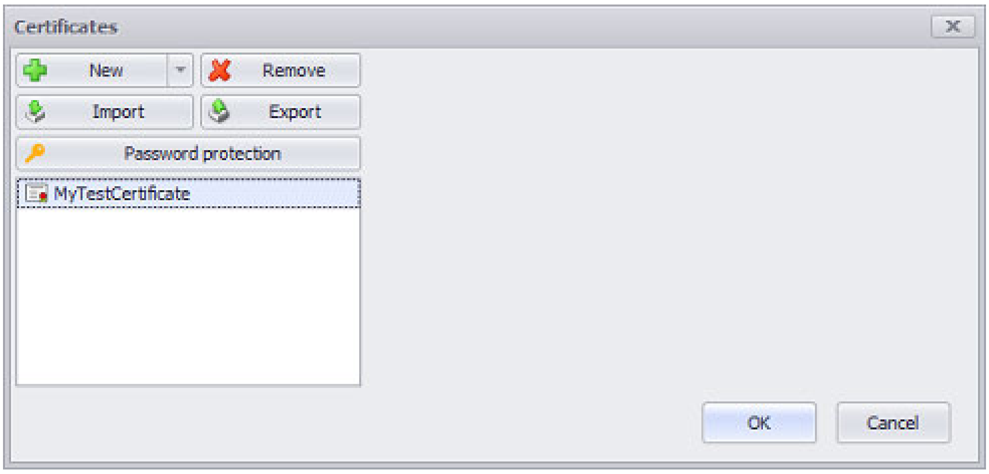


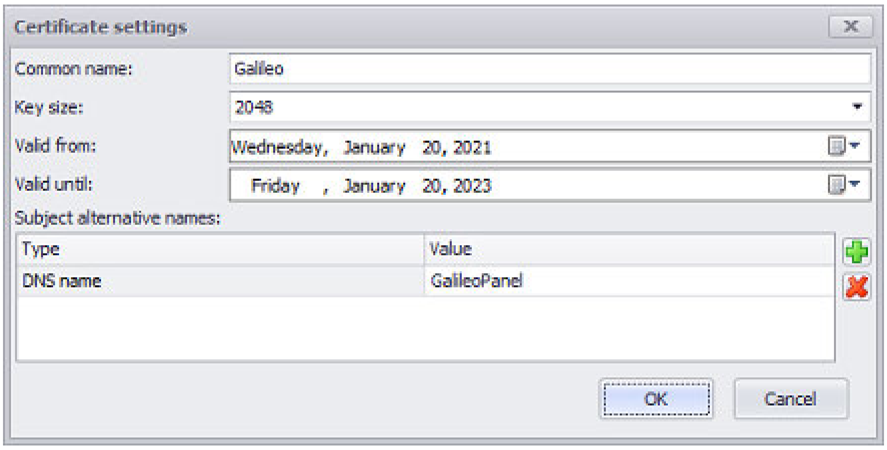
Figure 646: Certificates dialog box

19.5.7.1 ✚ New

Can be used to create a self-signed certificate.

The following dialog box helps when defining the certificate:

Device tab\ \✚



Common name: Galileo

Key size: 2048

Valid from: Wednesday, January 20, 2021

Valid until: Friday , January 20, 2023

Subject alternative names:

Type	Value
DNS name	GalileoPanel

Figure 647: Certificate Settings dialog box

Name: Name of certificate.
This information will be used in the Issuer and Subject certificate fields.

Key size: Key length for the public key (2048 bits or 4096 bits)

Valid from/until The period during which the certificate will be valid

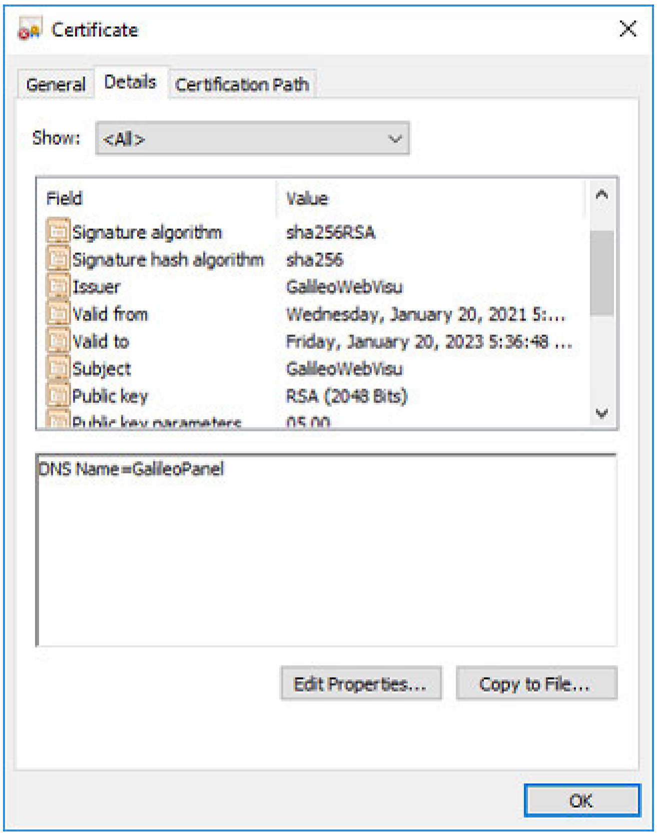
Subject Alternative Names:

Used to add additional fields for the certificate (IP, DNS Name, URI)

✚ Used to add an entry

✖ Delete

The settings of the screenshot above will result in following certificate (excerpt from the Windows detail view of the certificate).



Show: <All>

Field	Value
Signature algorithm	sha256RSA
Signature hash algorithm	sha256
Issuer	GalileoWebVisu
Valid from	Wednesday, January 20, 2021 5:...
Valid to	Friday, January 20, 2023 5:36:48 ...
Subject	GalileoWebVisu
Public key	RSA (2048 Bits)
Public key parameters	05 00

DNS Name=GalileoPanel

Figure 648: Microsoft Windows certificate view

When a connection is established between a web client and the web service with the use of a self-signed certificate, the browser will generate a warning telling the user that the site is a non-trusted site. This can be bypassed by installing the certificate on the web client.

19.5.7.2 ➕ New ▾ Add external certificate

Can be used to create a self-signed certificate.

The following dialog box helps when defining the certificate:

Device tab 📜 \ ➕ ▾

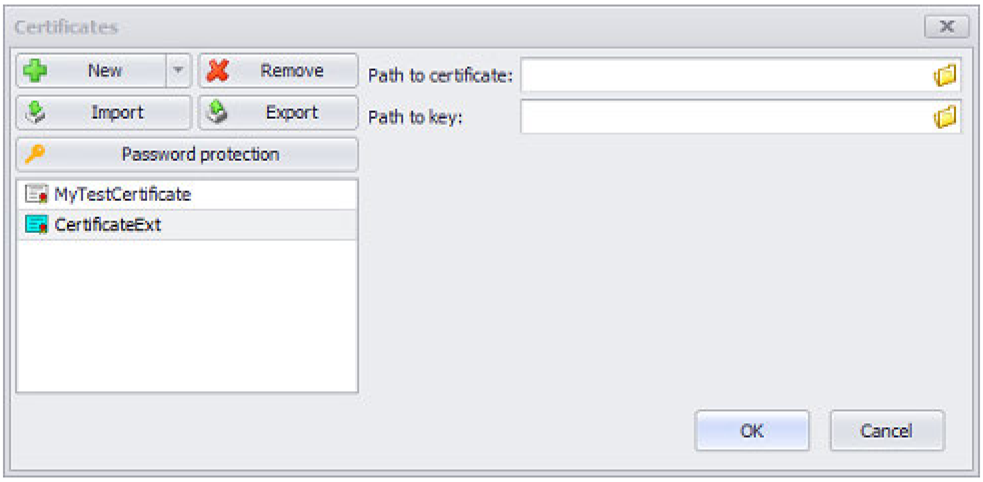
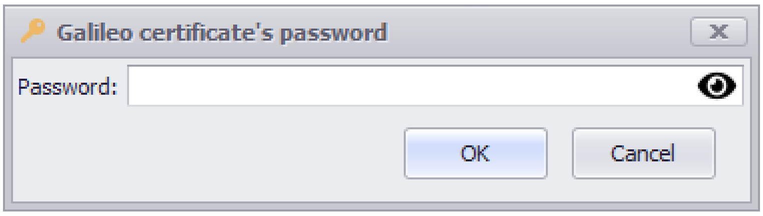


Figure 649: Certificate Settings dialog box

Certificate path:	Path to security certificate (CRT file)
Key path:	Path to private key (KEY file)
Remove	Used to delete existing certificates from the GALILEO project
Import	Can be used to import certificates and certificate containers. Supported formats: PKCS#12, CER, DER
Export	Used to export a certificate to a PKCS#12 file.

19.5.7.3 Password protection

To protect the certificates used in GALILEO from misuse, you can protect them with a password. When certificate information is required (e.g., during compilation or when using the export function), the password prompt will appear for the corresponding certificate:

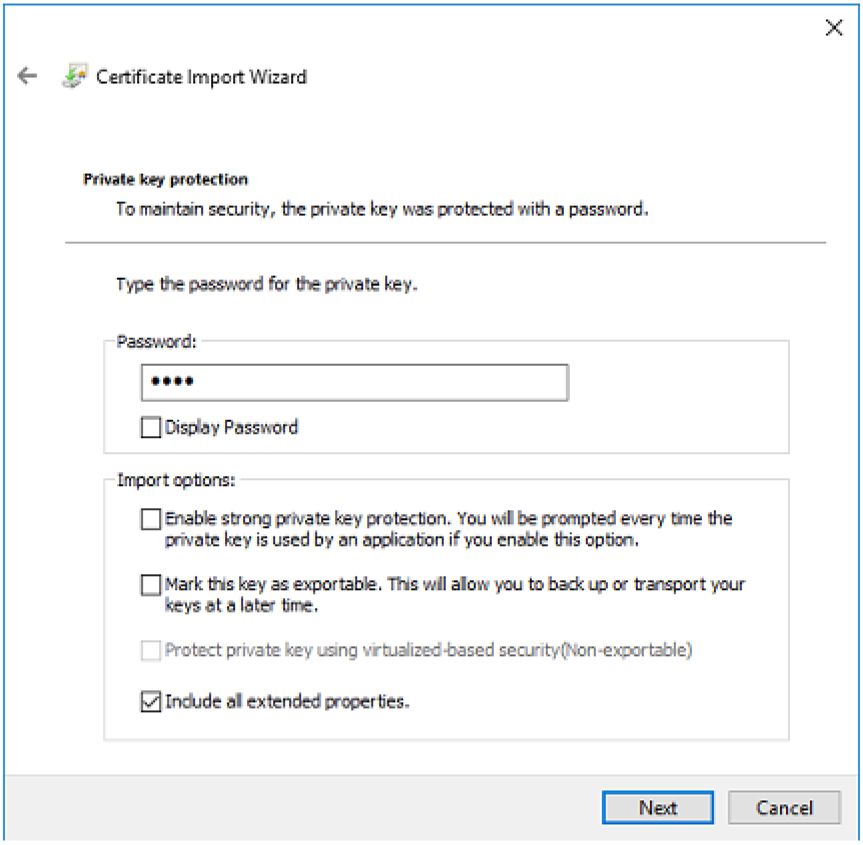


GALILEO will store the entered password (if correct) as long as the GALILEO project is open. This means that you will not have to enter the password again every time you run a compilation operation.

The password is stored in an encrypted PKCS#12 container.

WARNING
There is no way to reset this password if you lose it.
The certificates will become unusable and you will have to create new certificates.

The password will also be requested, for example, for imports into another system:



Certificate Import Wizard

Private key protection

To maintain security, the private key was protected with a password.

Type the password for the private key.

Password:

••••

□ Display Password

Import options:

□ Enable strong private key protection. You will be prompted every time the private key is used by an application if you enable this option.

□ Mark this key as exportable. This will allow you to back up or transport your keys at a later time.

□ Protect private key using virtualized-based security(Non-exportable)

☑ Include all extended properties.

Figure 650: Microsoft Windows Certificate Import Wizard

20. Overviews

20.1 Explanation of table format display for functions

GALILEO features a series of functions that perform actions on various objects and visualization objects.

Functions can basically be triggered in one of two ways:

1. When a device operator taps on a button (on the device screen) that is linked to a function.
2. When the PLC triggers a function using the control structure.

Not all functions can be called with function key visualization objects and not all functions can be called by the PLC. Functions that are not intended to be called by a PLC will be ignored, or GRS will show the following error message:
This function is not supported <%1>
(where %1 will be replaced with the function name).

You can find all the functions with a clear description, assigned to the relevant topic or collected there, in the Topics tab of GALILEO User Help. Each description will also include a table with the following information:

Number of the Function	Every available function has a unique function number. Specifying this function number in the 4th data word of the → section "Control structure", page 603 will cause GRS to carry out the corresponding action if control bit 15 in the control structure is set to 1.
Executable from the PLC	Not all functions can be triggered by the PLC, e.g., user management functions. Some can only be triggered by tapping on the button for a function.
Function with ON/OFF state	Some functions work like switches. An ON state and an OFF state can be defined for the function key visualization object. If the function is triggered via the PLC, the desired target state will be specified in bit 14 of the control structure's 3rd data word. OFF = Bit is 0, ON = Bit is 1. The → "Change to language", page 660 and → "Start/Stop Graph Block", page 732 functions are examples of functions with an ON/OFF state. Bit 14 is ignored in the case of functions that do not have an ON/OFF state (e.g., → "Print History", page 1033).
Parameter	If the function also needs a parameter, it can be configured with the function key visualization object. If the function is triggered via the PLC, the parameter must be entered in the 5th data word. Functions in the Recipe group will also require the ID for the recipe object and the index for the recipe entry, both of which must be entered in the 6th data word
Function available in standard screens	There are functions that are not available in standard screens.
Function available with keyboards	Only functions from the Characters, Recipes, and Info Displays groups are available on keyboard screens.

Function available in script	Whether it is allowed to use the function in a script.
Function available in application entry script	Whether it is allowed to use the function in a script used as an application entry script.

For instance, language changes can be implemented using the → "Change to language", page 660 function, which device operators can run by tapping on the corresponding button on the device screen and which the PLC can run by calling the function. To call the function, the PLC needs to set bit 15 in the control structure after the number for the desired language is entered into the control structure's 5th data word and function number <612 (264h)> is entered into the control structure's 4th data word.

For a more detailed description of the correct procedure, see chapter → section "Controlling functions with the PLC", page 602.

20.2 Visualization object functions

20.2.1 Container

- → "First Entry", page 305
- → "Last Entry", page 305
- → "Move Next Entry", page 305
- → "Move Previous Entry", page 306
- → "Set Active Entry", page 306

20.2.2 Controls

20.2.2.1 Static Controls

HTML-Viewer

- → "Load URL", page 437
- → "Load URL Var", page 437

Image

- → "Load Image", page 443
- → "Load Image Var", page 443
- → "Select Image ", page 444
- → "Select Image Var", page 445

20.2.2.2 Dynamic Controls

Option list

- → "List box: First Line", page 404
- → "List box: Last Line", page 405
- → "List box: Line down", page 405
- → "List box: Line up", page 406
- → "List box: Page up", page 407
- → "List box: Page down", page 406
- → "List box: Select", page 407
- → "List box: SelectVar", page 408

Dialog button

Can be called with a function key:

→ "CloseDialog", page 378
→ "CloseAllDialogs", page 377

Can be called as functions in scripts:

→ "OpenDialog", page 378
→ "CloseDialog", page 378
→ "CloseAllDialogs", page 377

Graph Block

→ "Delete Archive File", page 722
→ "Load Archive File", page 724
→ "Load Archive File Var", page 725
→ "Load Archive File Full Path As Var", page 724
→ "Save Archive File", page 725
→ "Save Archive File As", page 726
→ "Save Archive File As Var", page 727
→ "Save Archive File Full Path As Var", page 727

→ "Start/Stop Graph Block", page 732

→ "Display Time Graph End", page 723
→ "Display Time Graph Ruler", page 723
→ "Display Time Graph Start", page 723
→ "Scroll Left", page 728
→ "Scroll Left (Ruler)", page 728
→ "Scroll Left 1 (Ruler)", page 728
→ "Scroll Right", page 729
→ "Scroll Right (Ruler)", page 729
→ "Scroll Right 1 (Ruler)", page 730
→ "Scroll to End", page 730
→ "Scroll to End (Ruler)", page 731
→ "Scroll to Start", page 731
→ "Scroll to Start (Ruler)", page 731
→ "Start/Stop Autoscroll", page 732
→ "Zoom In", page 734
→ "Zoom Out", page 735

→ "Delete Graph Buffer", page 722
→ "Delete Viewer Buffer", page 722
→ "Delete Archive File", page 722

→ "View Archive File", page 733
→ "View Archive File Var", page 733
→ "View Archive File Full Path As Var", page 734

Parameter list

→ "First Line (Parameter list)", page 828
→ "Last Line", page 830
→ "Line down", page 832
→ "Line up", page 834
→ "Page down", page 835
→ "Page up", page 837
→ "Show Tag Help ", page 941
→ "Sort ascend/descend", page 839
→ "Sort by Description", page 841
→ "Sort by Number", page 843

Recipes

→ "First Line (Recipe)", page 772
→ "Last Line", page 773
→ "Line Down", page 773
→ "Line Up", page 773
→ "Page Down", page 776
→ "Page Up", page 776

Change Recipe Entry Name→ "Change Recipe Entry Name", page 766
Change Recipe Entry Name→ "Change Recipe Entry Name userdef.", page 766

→ "Load Recipe Entry", page 774
→ "LoadRecipeEntryVar", page 775
→ "LoadSelectedRecipeEntry", page 775
→ "LoadLastRecipeEntry", page 774

→ "Save Recipe Entry", page 782
→ "Save Recipe Entry As", page 783
→ "Save Recipe Entry As userdef.", page 781
→ "Save Recipe Entry Var", page 784
→ "Save Recipe Entry As Var", page 783
→ "Save As/Load Recipe Entry", page 782
→ "Save As/Load Recipe Entry userdef.", page 779
→ "SaveActualRecipeEntry", page 782
→ "Save As to next", page 779

→ "Save As to next userdef", page 780
→ "Save As to next Var", page 780

→ "Clear Recipe Entry", page 767

→ "Display: Copy Entry", page 768
→ "Display: Loaded Entry", page 769
→ "Display: New Entry", page 769
→ "Display: Selected Entry", page 769

→ "Copy Recipe Entry", page 768
→ "Copy to next", page 767
→ "Paste Recipe Entry", page 777

→ "Protect Recipe Entry", page 777

→ "Find Recipe Entry", page 770
→ "Find Recipe Entry (start)", page 771
→ "Find Recipe Entry userdef.", page 771
→ "Find Rcp Entry userdef. (start) ", page 771
→ "Find Next Recipe Entry", page 772
→ "Find Next Recipe Entry (start)", page 770

→ "New to next", page 775
→ "New to next userdef.", page 776
→ "Restore Default Values", page 778
→ "Select Loaded Recipe Entry", page 784
→ "Refresh Labels", page 778

Video Player

→ "Video:Load", page 502
→ "Video:LoadVar", page 503
→ "Video:Pause", page 503
→ "Video:Select", page 504
→ "Video:SelectVar", page 504
→ "Video:Start", page 505
→ "Video:Stop", page 505

Webcam

→ "WebCam:Capture Image", page 519
→ "WebCam:Capture Image As", page 520
→ "WebCam:Capture Image Var", page 520
→ "WebCam:Capture Image Var As", page 521
→ "WebCam:Start", page 522
→ "WebCam:Stop", page 522

20.2.3 Value Conversion Manager

→ "Conversion of Group", page 925
→ "Next Conversion of Group", page 926
→ " Value Display/Entry", page 506

20.2.4 Text management

→ "Change to language", page 660

20.2.5 Help management

→ "Show Tag Help ", page 941

20.2.6 Alarm-management

20.2.6.1 Display, navigation, and sorting functions for alarm lists:

→ "Error first line", page 884
→ "Error last line", page 885
→ "Error line down", page 885
→ "Error line up", page 886
→ "Error line extend", page 885
→ "Error page down", page 886
→ "Error page up", page 886
→ "Error Priority ON/OFF", page 887
→ "Error Autoscroll On/Off", page 883

20.2.6.2 Error help functions:

→ "Error help", page 885

20.2.6.3 Error confirmation functions:

→ "Error last line", page 885
→ "Error confirm", page 883
→ "Error confirm (actual window)", page 884
→ "Error confirm all", page 884

20.2.6.4 Functions used to display and save the error history:

→ "Clear System History", page 882
→ "Clear History in Screen", page 882
→ "Clear all Histories", page 882
→ "Save Error History as File", page 888
→ "Save Error History as File Var", page 888
→ "Save System History as File", page 888
→ "Save System History as File Var", page 889
→ "Show History On/Off", page 889
→ "Show Sys. History On/Off", page 889
→ "Save History to CSV", page 994
→ "Save History to CSV Var", page 994
→ "Save History of previous day to CSV", page 993
→ "Save History of previous day to CSV Var", page 994

20.2.6.5 Refresh functions:

→ "Error Refresh", page 887

20.2.7 User Management

→ "Change Group", page 902
→ "Change Group -userdef.", page 903
→ "Change PW (User x)", page 903
→ "Change PW (User x) -userdef.", page 904
→ "Change PW (other User) -userdef.", page 904
→ "Change PW (Own) -userdef.", page 904
→ "Change PW (other User)/ ChangePWOther", page 905
→ "Change PW (own)", page 906
→ "Delete User -userdef.", page 907
→ "Delete User", page 908
→ "Lock/Unlock User", page 909
→ "Lock/Unlock User -userdef.", page 910
→ "Login User", page 910
→ "Login User -userdef.", page 911
→ "Login User x", page 911

→ "Login User x -userdef.", page 912
→ "Logout User", page 912
→ "New User -userdef.", page 912
→ "New User", page 913
→ "RefreshAutoLogoutTime", page 914
→ "Select Group", page 914
→ "Select Group -userdef.", page 914
→ "Select User", page 915
→ "Select User -userdef.", page 915
→ "Show_all_Groups", page 915
→ "Show all Groups -userdef.", page 916
→ "Show all Users of Group", page 916
→ "Show all Users of Group -userdef.", page 916
→ "Show All Users -userdef.", page 917
→ "Show Groups of User", page 917
→ "Show Groups of User -userdef.", page 917
→ "Show all Users", page 917
→ "Unlock User", page 918
→ "Unlock User -userdef.", page 919
→ "Unlock user x", page 919
→ "UserLogin", page 919

20.2.8 User Logger

→ "Add String Tag Log", page 950
→ "Add Text Log", page 950
→ "Add Value Tag Log", page 950
→ "Logger ON-OFF", page 951
→ "Flush Logger", page 950
→ "View Current Log File", page 951

20.2.9 Communication

→ "Comm. on Connection x ON/OFF", page 625
→ "Connection x persistent ON/OFF", page 625
→ "GPI - Communic. Dialog", page 625
→ "Quiet on Connection x ON/OFF", page 626
→ "Start Sync to Connection", page 626
→ "DisconnectAllClients", page 627
→ "Flush persistent tag storage values", page 628

→ "CAN SDO Node x ON/OFF", page 628

→ "CAN SDO Node x Persistent ON/OFF", page 628

→ "CAN PDO ON/OFF", page 629

→ "CAN PDO Persistent ON/OFF", page 629

→ "Modbus Station No ON/OFF", page 629

20.2.10 Printer functions (printer)

→ "Change to Printer", page 1033
→ "Print History", page 1033
→ "Print Parameter List", page 1033
→ "Print Parameter List (no Header)", page 1034
→ "Print Report", page 1034
→ "Print Screen", page 1034

20.2.11 Keyboards

20.2.11.1 Keyboard

→ "Alpha numeric Keyboard", page 332
→ "Call user defined Keyboard", page 334
→ "Inc/Dec 1", page 334
→ "Inc/Dec 10", page 335
→ "None", page 337
→ "Numeric keyboards", page 337
→ "Numeric Password", page 339
→ "Select Numeric, Select Numeric Digit", page 339

20.2.11.2 Characters/Stream Characters

→ "Dec 1 (assigned)", page 321
→ "Dec 10 (assigned)", page 321
→ "Display 'Scroll Window'", page 321
→ "Display 'Maximum'", page 322
→ "Display 'Minimum'", page 322
→ "Display 'Value/Text'", page 322
→ "Inc 1 (assigned)", page 322
→ "Inc 10 (assigned)", page 323
→ "Key "Alt Layout" User", page 323
→ "Key "AltGr" User", page 323
→ "Key "CAPS" User", page 323
→ "Key "Ctrl" User (Terminal)", page 324
→ "Key "Scroll cursor left"", page 325
→ "Key "Scroll cursor right"", page 325
→ "Key "Scroll cursor to begin"", page 325
→ "Key "Scroll cursor to end"", page 326
→ "Key "number"", page 324
→ "Key "number" with "RET"", page 324

→ "Key "RET" and "Call Script Function"", page 325
→ "Key "string"", page 326
→ "Key "string" with "RET"", page 326
→ "Key "UNICODE"", page 326
→ "Key 'Window Move'", page 327
→ "Keyb. Mask CANCEL + Call Script", page 327
→ "Keyboard Mask OK + Call Script", page 327
→ "Keyboard Mask YES", page 328
→ "Keyboard Mask YES + Call Script", page 328
→ "Keyboard Mask CANCEL", page 328
→ "Keyboard Mask Display", page 329
→ "Keyboard Mask OK", page 329
→ "UserM: "Group Sel. -userdef"", page 329
→ "UserM: "Group Selection"", page 329
→ "UserM: "Lock"", page 330
→ "UserM: "Password"", page 330
→ "UserM: "Password confirm"", page 330
→ "UserM: "Password new"", page 330
→ "UserM: "UnLock"", page 331
→ "UserM: "User"", page 331
→ "UserM: "Sel. User -userdef"", page 331
→ "UserM: "Select User"", page 332

20.3 Internal functions (Internals)

Internal functions are functions that trigger or change something internally in the device.

20.3.1 Backlight: Learn Dim Value

The screensaver reduces the brightness level in order to extend the backlight's life. This function sets the current backlight brightness as a reduction value for the screensaver. By default, the screensaver will reduce the brightness to the lowest level, where the actual percentage corresponding to this level will depend on the specific device model being used. This setting will be retained even after the device is switched off.

→ Reducing brightness levels will always contribute to extending the backlight's life (please refer to theBrightness 40% function, for example).

Number of the Function	134 (86h)
Executable from the PLC	No
Function with ON/OFF state	No
Function available in standard screens	Yes
Function available with keyboards	No
Function available in script	Yes
Function available in application entry script	Yes

20.3.2 Backlight Off

This function can be used to switch off the display's backlight. The backlight will be switched back on as soon as the touchscreen is touched again. However, please note that this touch will not be able to trigger any functions, i.e., in order to trigger a function after the backlight is switched off, the operator will need to touch the screen once so that the backlight is switched back on and then touch it again as necessary in order to run the function.

→ This function is only supported by XV100, XV150, XV300, XV400 devices.

→ Switching the backlight on and off too often may result in a shorter backlight life. Accordingly, this function should only be used when the backlight will remain off for an extended period of time so that switching it on and off will not have a negative impact on its life. Reducing brightness levels will also contribute to extending the backlight's life – please refer to the Brightness 40% function as well.

Number of the Function	133 (85h)
Executable from the PLC	Yes

Function with ON/OFF state	No
Function available in standard screens	Yes
Function available with keyboards	No
Function available in script	Yes
Function available in application entry script	Yes

20.3.3 Beep

This function can be used to start an audible signal from the PLC. The frequency and duration of the signal is not adjustable (500 Hz, 100ms).

→ This function is only supported by XP500 devices and GALILEO Open.

Number of the Function	180 (B4h)
Executable from the PLC	Yes
Function with ON/OFF state	No
Function available in standard screens	Yes
Function available with keyboards	No
Function available in script	Yes
Function available in application entry script	Yes

20.3.4 Boot System

This function shuts down the device. This means that communication will be terminated, the project will be closed, and the device will be restarted.

Number of the Function	99 (63h)
Executable from the PLC	Yes
Function with ON/OFF state	No
Function available in Standard screens	Yes
Function available in keyboard masks	No
Function available in script	Yes
Function available in application entry script	No

20.3.5 Brightness Dynamic

Changes the display's brightness level. The current for the backlight will be reduced to the value in the tag. The tag must contain a value between 1 and 100. This will significantly extend the backlight's life. This setting will be retained even after the device is switched off. This function can only be used on the device.

Number of the Function	127 (7Fh)
Executable from the PLC	No
Function with ON/OFF state	No
Parameter 1	Tag with the percentage value of the backlight setting
Function available in standard screens	Yes
Function available with keyboards	No
Function available in script	Yes

Function available in application entry script	Yes

20.3.6 Brightness Static

Changes the display's brightness level. The current for the backlight will be reduced to the configured value in the tag. This value is specified as a percentage and can be any number between 1 and 100. This will significantly extend the backlight's life. This setting will be retained even after the device is switched off. This function can only be used on the device.

Number of the Function	126 (7Eh)
Executable from the PLC	Yes
Function with ON/OFF state	No
Parameter 1	Percentage value of the backlight setting
Function available in standard screens	Yes
Function available with keyboards	No
Function available in script	Yes
Function available in application entry script	Yes

20.3.7 Brightness 40%

Reduces the display's brightness level. The current for the backlight will be reduced to 40%. This will significantly extend the backlight's life. This setting will be retained even after the device is switched off. This function can only be used on the device.

Number of the Function	131 (83h)
Executable from the PLC	Yes
Function with ON/OFF state	No
Function available in standard screens	Yes
Function available with keyboards	No
Function available in script	Yes
Function available in application entry script	Yes

20.3.8 Brightness 60%

Reduces the display's brightness level. The current for the backlight will be reduced to 60%. This will significantly extend the backlight's life. This setting will be retained even after the device is switched off. This function can only be used on the device.

Number of the Function	130 (82h)
Executable from the PLC	Yes
Function with ON/OFF state	No
Function available in standard screens	Yes
Function available with keyboards	No
Function available in script	Yes
Function available in application entry script	Yes

20.3.9 Brightness 80%

Reduces the display's brightness level. The current for the backlight will be reduced to 80%. This will significantly extend the backlight's life. This setting will be retained even after the device is switched off. This function can only be used on the device.

Number of the Function	129 (81h)
Executable from the PLC	Yes
Function with ON/OFF state	No
Function available in standard screens	Yes
Function available with keyboards	No
Function available in script	Yes
Function available in application entry script	Yes

20.3.10 Brightness 100%

Sets the display's brightness to its maximum level. The current for the backlight will be set to 100%. This setting will be retained even after the device is switched off. This function can only be used on the device.

Number of the Function	128 (80h)
Executable from the PLC	Yes
Function with ON/OFF state	No
Function available in standard screens	Yes
Function available with keyboards	No
Function available in script	Yes
Function available in application entry script	Yes

20.3.11 Call 2 Script Functions

Pressing the button calls, once, the event script that is defined as the first parameter. Releasing the button runs the event script that is defined as the second parameter.

Number of the Function	182 (B6h)
Executable from the PLC	No
Parameter 1	Name of an event script (this script will be called when the button is tapped on)
Parameter 2	Name of an event script (this script will be called when the button is released)
Function with ON/OFF state	No
Function available in Standard screens	Yes
Function available in keyboard masks	No
Function available in script	No
Function available in application entry script	No

20.3.12 Call Script Function

Starts an event script.
The script and its subfunctions are run once.

Various functions can be run within the script, e.g., mask changes, setting tag values, making calculations.

Number of the Function	181 (B5h)
Executable from the PLC	Yes
Parameter 1	Name of the event script (for special function and in the script) or ID number of the event script if run from the PLC
Function with ON/OFF state	No
Function available in Standard screens	Yes
Function available in keyboards	No
Function available in script	No
Function available in application entry script	No

20.3.13 Call DLL Function

This function can be used to call an external function within a DLL. Example using tag <tagX> as the input argument and tag <tagY> as the input/output argument:

Numbers of the Functions	213 (D5h)
Executable from the PLC	No
Function with ON/OFF state	No
Parameter 1	The filename of the DLL and the name of the function being called as a text with the following format: <DLLName, FunctionName>
Parameter 2	Option Tags can be passed either as input arguments ("call by value") or as input/output arguments ("call by reference"). Input arguments: Keyword "in" The tag's value will be passed to the DLL function. Operations within the DLL will not affect the tag value. Input/output arguments: The tag's value will be passed to the DLL function. It will be possible to change the value within the DLL, and at the end of the DLL function, this value will be written to the tag in GALILEO. <in tagX,ref tagY > Arrays that are passed as parameters are subdivided into the individual elements individually. This means that a call with "in MyArray", where "MyArray" is an array with 2 elements, will be converted in the following DLL call: "in MyArray[0], in MyArray[1]". In other words, the DLL function must receive two parameter pointers and handle them separately.
Function available in Standard screens	Yes
Function available in keyboard masks	No

Function available in script	Yes
Function available in application entry script	Yes

20.3.14 CloseMenu

This function from the "Screen" group can be used to hide a menu button visualization object with the use of an event script.

Function available in script	Yes
Function available in application entry script	Yes

20.3.15 Execute

This function starts a program with the name passed as the argument for parameter 1. A text that will be passed as the start parameter for the program being run can optionally be passed as an argument for parameter 2.

Execute batch files

The Execute function can be used to run EXE files. If you want to run a batch file, you will need the command shell. The first argument must call the command shell (cmd). The second argument starts with </C>, which will close the command shell after the batch file is run. The batch file is defined next, including the path. The parameters for the batch file are listed at the end.

Example

Parameter 1: cmd

Parameter 2: /C \StorageCard\xy.bat 1

Number of the Function	197 (c5h)
Executable from the PLC	No
Function with ON/OFF state	No
Parameter 1	Filename of the program that should be run.
Parameter 2	The optional parameter that will be passed to the program being started.
Function available in Standard screens	Yes
Function available in keyboard masks	No
Function available in script	Yes
Function available in application entry script	No

20.3.16 Execute (dynamic)

This function starts a program with the name passed as the argument for parameter 1. A text that will be passed as the start parameter for the program being run can optionally be passed as an argument for parameter 2.

Number of the Function	209 (d1h)
Executable from the PLC	No
Function with ON/OFF state	No
Parameter 1	String tag for the filename of the program that should

	be run.
Parameter 2	Optional string tag with the content that will be passed to the program being started.
Function available in Standard screens	Yes
Function available in keyboard masks	No
Function available in script	Yes
Function available in application entry script	Yes

20.3.17 GetActScreenNo

Returns the number of the screen currently being shown.

Return parameter	Actual mask number
Function available in script	Yes
Function available in application entry script	No

20.3.18 GetTickCount

This function returns the time since system start in milliseconds.

Number of the Function	–
Executable from the PLC	No
Function with ON/OFF state	No
Parameter 1	–
Return parameter	Time in milliseconds
Function available in standard screens	No
Function available with keyboards	No
Function available in script	Yes
Function available in application entry script	No

20.3.19 Last Screen

This function makes the device switch to the screen that was last displayed regarding the screen call hierarchy.

Screen call hierarchy means following:

- The control "Screen Change" as well as the script function "Screen Change" add an entry into the screen calling hierarchy.
- After 3 consecutive screen changes (Start -> ScreenA -> ScreenB -> ScreenC) there are 3 screens in the hierarchy (ScreenA, ScreenB, ScreenC).
- By executing this function, everytime there will be a navigation by one step upwards. After the first execution from ScreenC to ScreenB. Afterwards to ScreenA. At the end to Start.

The function Last Screen itself doesn't add another entry into this hierarchy: Otherwise, two screens - both with this function -, would simply only cause a switching between these two screens. It would not be possible to navigate a level upward.

The most recent 100 screens will be stored.

→ The system tag LastScreenID works differently.
This system tag always stores the screen ID of the previously displayed screen. It doesn't considers any call hierarchies.

Number of the Function	32 (20h)
Executable from the PLC	Yes
Function with ON/OFF state	No
Function available in standard screens	Yes
Function available with keyboards	No
Function available in script	Yes
Function available in application entry script	No

20.3.20 ReadPLCDateTime

The device will read the Date_Time system tag from the communication system defined using the argument.

Number of the Function	150 (96h)
Executable from the PLC	No
Function with ON/OFF state	No
Parameter	Communication number (0–7) used to specify the PLC from which the Date_Time system tag should be read
Function available in Standard screens	Yes
Function available in keyboards	No
Function available in script	Yes
Function available in application entry script	Yes

20.3.21 ReReadTag

This function makes the device explicitly read this tag from the communication interface.

Parameter entered	Name of a tag
Function available in script	Yes
Function available in application entry script	No

20.3.22 Restart Runtime

Terminates GRS and restarts it.

Number of the Function	98 (62h)
Executable from the PLC	Yes
Function with ON/OFF state	No
Function available in Standard screens	Yes
Function available in keyboard masks	No
Function available in script	Yes
Function available in application entry script	No

20.3.23 Save Screen as Bitmap

The function stores the current mask as a bitmap file on the Flash card.

The filenames are 0.bmp, 1.bmp, 2.bmp, etc., with the count starting from 0.bmp every time after the device is switched off.
The filename will be incremented every time this special function is run.

Number of the Function	255 (0FFh)
Executable from the PLC	Yes
Function with ON/OFF state	No
Function available in standard screens	Yes
Function available with keyboards	No
Function available in script	Yes
Function available in application entry script	No

20.3.24 Save Screen as Bitmap Var

The function stores the current mask as a bitmap file on the Flash card.

→ Important:
The file name must be entered with the path, the file name and the suffix.

Number of the Function	212 (D4h)
Executable from the PLC	No
Function with ON/OFF state	No
Parameter 1	String tag with file name
Function available in standard screens	Yes
Function available with keyboards	No
Function available in script	Yes
Function available in application entry script	No

20.3.25 ScreenChange

This function makes the device switch from the current screen to the selected target screen. XV300 and XP500 devices support target screen animations, as does GALILEO Open. For these devices, the type of animation can be specified with Parameter2.

There are three basic types of animation:

1. Cover: The current screen will remain in place and the target screen will fly in
2. Uncover: The target screen will remain in place and the current screen will fly out
3. Push: The target screen will move in, "pushing" the current screen out

Parameter1	Name of screen
Parameter2	Animation selection <none> - No animation Cover FromLeft - The target screen will fly in from the left

	CoverFromRight - The target screen will fly in from the right CoverFromTop - The target screen will fly in from above CoverFromBottom - The target screen will fly in from below UncoverLeft - The current screen will fly out towards the left UncoverRight- The current screen will fly out towards the right The current screen will fly out upwards The current screen will fly out downwards PushLeft - The target screen will move in, pushing the current screen out to the left PushRight - The target screen will move in, pushing the current screen out to the right PushUp - The target screen will move in, pushing the current screen up and out PushDown - The target screen will move in, pushing the current screen down and out
Function available in script	Yes
Function available in application entry script	No

20.3.26 Script

Call of a different script. Other Event Scripts can be called inside an event script. Only Sub functions of a loop script can be called inside loop scripts.

Parameter entered	Name of a script
Function available in script	Yes
Function available in application entry script	No

20.3.27 Select ASCII Tag

Calling this function will select the first visualization object of type PLC text within the screen. This means that input can be entered directly for this object.

Tapping on the Return key will confirm the input, and the object will be deselected.

The Select ASCII Tag function sets the focus on the first PLC text visualization object. The <Select Alphanumeric> keyboard must be selected as the keyboard in the properties for the PLC text visualization object.

Numbers of the Functions	202 (CAh)
Executable from the PLC	No
Parameter 1	The string tag that should be changed.
Function with ON/OFF state	No
Function available in Standard screens	Yes
Function available in keyboard masks	No
Function available in script	Yes
Function available in application entry script	No

20.3.28 Select Numeric Tag

When this function is called, the current screen on the device screen must feature at least one visualization object of type Value Display/Entry that uses the <Parameter 1> tag for the Select Numeric Tag function. <Select Numeric> must be selected as the keyboard in the visualization object's properties.

→ If there is no value display/entry visualization object, the Select Numeric Tag function will not do anything.

If there is exactly one Value Display/Entry visualization object on the screen, the object will be selected and it will be possible to enter input on this visualization object directly, e.g., with an external device such as a barcode scanner or keyboard. Pressing the Enter key will complete the input sequence, and the visualization object will be deselected.

If a screen features several Value Display/Entry visualization objects that use the <Parameter 1> tag for the Select Numeric Tag function, then the following rules will be used to determine which object should be selected:

1. If one of the visualization objects is set up with a keyboard of type Select Numeric, that object will be selected.
2. If there are multiple visualization objects with a keyboard of type Select Numeric set up, the first along the Z sequence will be selected. The Z order of visualization objects can be determined using the Controls Manager.

Numbers of the Functions	203 (CBh)
Executable from the PLC	No
Parameter 1	The numeric tag that should be changed.
Function with ON/OFF state	No
Function available in Standard screens	Yes
Function available in keyboard masks	No
Function available in script	Yes
Function available in application entry script	No

This function can come in handy when using a barcode scanner, for example. The function can be called in a standard script, and when the barcode scanner is activated, the barcode can be displayed directly in the value display/entry visualization object. This will make the work easier for device operators, who would otherwise have to tap on the Value Display/Entry visualization object every time before scanning.

20.3.29 Set day

This function shows an Increment/Decrement keyboard (Inc/Dec) for setting the day on the integrated clock (value between 1 and 31).

Number of the Function	146 (92h)
Executable from the PLC	No
Function with ON/OFF state	No

Function available in Standard screens	Yes
Function available in keyboard masks	No
Function available in script	No
Function available in application entry script	No

20.3.30 Set day -userdef

This function can be used to open a custom keyboard for setting the day on the integrated clock
(value between 1 and 31).

Number of the Function	154 (9Ah)
Executable from the PLC	No
Function with ON/OFF state	No
Function available in Standard screens	Yes
Parameter 1	Selected User Defined Keyboards
Function available in keyboard masks	No
Function available in script	No
Function available in application entry script	No

20.3.31 Set hour

This function shows an Increment/Decrement keyboard (Inc/Dec) for setting the hours time of the integrated clock (value between 0 and 23).

This setting is always in 24-hour format, irrespective of whether the time/date of the current project is set in US format.

Number of the Function	147 (93h)
Executable from the PLC	No
Function with ON/OFF state	No
Function available in Standard screens	Yes
Function available in keyboard masks	No
Function available in script	No
Function available in application entry script	No

20.3.32 Set hour -userdef

This function can be used to open a custom keyboard for setting the hour on the integrated clock
(value between 0 and 23).

This setting is always in 24-hour format, irrespective of whether the time/date of the current project is set in US format.

Number of the Function	155 (9Bh)
Executable from the PLC	No
Function with ON/OFF state	No
Parameter 1	Selected User Defined Keyboards

Function available in Standard screens	Yes
Function available in keyboard masks	No
Function available in script	No
Function available in application entry script	No

20.3.33 Set minute

This function shows an Increment/Decrement keyboard (Inc/Dec) for setting the minutes time of the integrated clock (value between 0 and 59).

Number of the Function	148 (94h)
Executable from the PLC	No
Function with ON/OFF state	No
Function available in Standard screens	Yes
Function available in keyboard masks	No
Function available in script	No
Function available in application entry script	No

20.3.34 Set minute -userdef

This function can be used to open a custom keyboard for setting the minutes on the integrated clock
(value between 0 and 59).

Number of the Function	156 (9ch)
Executable from the PLC	No
Function with ON/OFF state	No
Parameter 1	Selected User Defined Keyboards
Function available in Standard screens	Yes
Function available in keyboard masks	No
Function available in script	No
Function available in application entry script	No

20.3.35 Set month

This function shows an Increment/Decrement keyboard (Inc/Dec) for setting the month of the integrated clock (value between 1 and 12).

Number of the Function	145 (91h)
Executable from the PLC	No
Function with ON/OFF state	No
Function available in Standard screens	Yes
Function available in keyboard masks	No
Function available in script	No
Function available in application entry script	No

20.3.36 Set month -userdef

This function shows an Increment/Decrement keyboard (Inc/Dec) for setting the month of the integrated clock (value between 1 and 12).

Number of the Function	153 (99h)
Executable from the PLC	No
Function with ON/OFF state	No
Parameter 1	Selected User Defined Keyboards
Function available in Standard screens	Yes
Function available in keyboard masks	No
Function available in script	No
Function available in application entry script	No

20.3.37 Set second

This function shows an Increment/Decrement keyboard (Inc/Dec) for setting the seconds time of the integrated clock (value between 0 and 59).

Number of the Function	149 (95h)
Executable from the PLC	No
Function with ON/OFF state	No
Function available in Standard screens	Yes
Function available in keyboard masks	No
Function available in script	No
Function available in application entry script	No

20.3.38 Set second -userdef

This function can be used to open a custom keyboard for setting the seconds on the integrated clock (value between 0 and 59).

Number of the Function	157 (9dh)
Executable from the PLC	No
Function with ON/OFF state	No
Parameter 1	Selected User Defined Keyboards
Function available in Standard screens	Yes
Function available in keyboard masks	No
Function available in script	No
Function available in application entry script	No

20.3.39 Set Touch Lock

→ Please note that this function cannot be used for Galileo Open projects. The GRS runtime program will trigger a controlled error if the device does not support this function.

Calling this function will disable the touch zones. While the touch zones are disabled, an image will cyclically appear and disappear at random points on the device screen.

To reactivate the touch display (TouchEnable), one of two different procedures must be followed depending on the device model in question.

- IR touchscreen
  Touching the display at two points that are diagonal relative to each other (two coordinate pairs) and are at least 5 cm from each other will reactivate the touch zones.
- Resistive touchscreen
  Touching the display al all four corners of the screen clockwise or counterclockwise will reactivate the touch zones.

This function can also be triggered with the Control data block, 3rd data word, bit 1. If the touch zones are disabled with this bit in the PLC, it will only be possible to activate them again using the PLC. If the GALILEO Runtime System is closed, the touch zones will be re-enabled the next time it is started.

Please note that the screen settings also feature the option of disabling touch zones after x minutes without any activity. This property is configured using the Touch disable with timeout[min] field.

If the touch zones are disabled with the option in the screen settings or by directly calling this function, they can be activated again as indicated above (as appropriate for the device model being used).

Number of the Function	103 (67h)
Executable from the PLC	Yes
Function with ON/OFF state	No
Function available in standard screens	Yes
Function available with keyboards	No
Function available in script	Yes
Function available in application entry script	Yes

20.3.40 Set year

This function shows an Increment/Decrement keyboard (Inc/Dec) for setting the year of the integrated clock (value between 1997 and 2032).

Number of the Function	144 (90h)
Executable from the PLC	No
Function with ON/OFF state	No
Function available in Standard screens	Yes
Function available in keyboard masks	No
Function available in script	No
Function available in application entry script	No

20.3.41 Set year -userdef

This function can be used to open a custom keyboard for setting the year on the integrated clock
(value between 1997 and 2032).

Number of the Function	152 (98h)
Executable from the PLC	No
Function with ON/OFF state	No
Parameter 1	Selected User Defined Keyboards
Function available in Standard screens	Yes
Function available in script	No
Function available in application entry script	No

20.3.42 Show Frame On/Off

This function shows and hides the frame, incl. the window title, for the GRS sequential program. The last setting will be retained even after the GRS sequential program ends.

Number of the Function	164 (A4h)
Executable from the PLC	No
Function with ON/OFF state	Yes
Function available in Standard screens	Yes
Function available in keyboard masks	No
Function available in script	Yes
Function available in application entry script	Yes
Remark	Only the GPI supports this function.

20.3.43 ShowMenu

This function from the "Screen" group can be used to slide in a menu button visualization object from the edge of the standard screen with the use of an event script.

Parameter 1	Name of user control
Parameter 2	Menu alignment; the UserControl will slide in from the edge of the screen. Left - The UserControl will slide in from the left Right - The UserControl will slide in from the right Top - The UserControl will slide in from above Bottom - The UserControl will slide in from below
Parameter 3	Menu behavior; the UserControl will either slide in so that it overlaps the standard screen or the standard screen will be pushed to the side as far as necessary for the UserControl to be displayed. Overlap PushAway
Parameter 4	Turns the menu behavior defined in Parameter 3 on and off.

	ON OFF
Function available in script	Yes
Function available in application entry script	No

Sample script:

Say you want the ShowMenu function to be called when a button on the startup screen is tapped on. This function should make UserControl <SubScreen1> slide in from the right over the startup screen.

▶ Create a new user control named <SubScreen1> by clicking in *Project Tree\Screens\UserControls*, opening the context menu, and clicking on *Add new UserControl* .

▶ Now create an event script by clicking on *Project Tree\Scripts\Script* and then on *Context menu\Add script*, e.g., <ShowThatMenu>.

▶ Add the following lines to the script:

```
Screen.ShowMenu(SubMask1, Right, Overlap, ON);
```

▶ Open the startup screen, e.g., <Start>.

▶ Add a Function Key visualization object to the startup screen.

▶ In the properties for the function key, select the Call Script Function from the System group in the Function field.

▶ In the Script field, select the <ShowThatMenu> event script.

▶ Start the Galileo Simulator with function key F5 or by clicking on Home\ . Build.

20.3.44 Shut down

The GRS runtime program is terminated.

Number of the Function	100 (64h)
Executable from the PLC	Yes
Function with ON/OFF state	No
Function available in Standard screens	Yes
Function available in keyboard masks	No
Function available in script	Yes
Function available in application entry script	Yes

20.3.45 Summer time On/Off

This function allows the internal clock to be advanced or delayed by one hour.

This setting will be retained even after the device is switched off.

Number of the Function	102 (66h)

Executable from the PLC	Yes
Function with ON/OFF state	Yes
Function available in Standard screens	Yes
Function available in keyboard masks	No
Function available in script	Yes
Function available in application entry script	Yes

20.3.46 SyncWriteTag

This function makes the device write this tag synchronously to the communication interface. The script will be aborted and will not be resumed until the tag has been written.

Parameter entered	Name of a tag
Function available in script	Yes
Function available in application entry script	Yes

20.3.47 US date and time On/Off

This function is used to toggle between the US and 24 hour time/date display format.

This setting will be retained even after the device is switched off.

	24 hour format	US format
Date	Day.Month.Year[1]	Month.Day.Year
Time	Hours:Minutes.Seconds[2] (hours between 1 and 23)	Hours:Minutes.Seconds am/pm (hours between 1 and 12)

[1] The year can be in two-digit or four-digit format
[2] Seconds are optional

Number of the Function	101 (65h)
Executable from the PLC	Yes
Function with ON/OFF state	Yes
Function available in Standard screens	Yes
Function available in keyboard masks	No
Function available in script	Yes
Function available in application entry script	Yes

20.3.48 WriteCalendarWeekToValue

Writes the current calendar week to a tag defined as a parameter

Number of the Function	541 (21dh)
Executable from the PLC	No
Function with ON/OFF state	No
1. Parameter	Calendar week: Tag used to store the current calendar week
Function available in Standard screens	No
Function available in keyboards	No

Function available in script	Yes
Function available in application entry script	Yes

20.3.49 WriteDateFromValues

Sets the current system date based on tags defined as parameters.

Number of the Function	543 (21Fh)
Executable from the PLC	No
Function with ON/OFF state	No
1. Parameter	Tag: Tag with day information
2. Parameter	Month: Tag with month information
3. Parameter	Year: Tag with year information
Function available in Standard screens	No
Function available in keyboards	No
Function available in script	Yes
Function available in application entry script	Yes

20.3.50 WriteDateTimeFromValues

Sets the current system date and time based on tags defined as parameters.

Number of the Function	542 (21Eh)
Executable from the PLC	No
Function with ON/OFF state	No
1. Parameter	Hour: Tag with hour information
2. Parameter	Minutes: Tag with minutes information
3. Parameter	Seconds: Tag with seconds information
4. Parameter	Tag: Tag with day information
5. Parameter	Month: Tag with month information
6. Parameter	Year: Tag with year information
Function available in Standard screens	No
Function available in keyboards	No
Function available in script	Yes
Function available in application entry script	Yes

20.3.51 WriteDateTimeToValues

Writes the current system time to tags defined as parameters.

Number of the Function	540 (21ch)

Executable from the PLC	No
Function with ON/OFF state	No
1. Parameter	Hour: Tag for saving the current hour
2. Parameter	Minutes: Tag for saving the current minutes
3. Parameter	Seconds: Tag for saving the current seconds
4. Parameter	Day: Tag for saving the current day
5. Parameter	Month: Tag for saving the current month
6. Parameter	Year: Tag for saving the current year
Function available in Standard screens	No
Function available in keyboards	No
Function available in script	Yes
Function available in application entry script	Yes

20.3.52 WritePLCDateTime

The device will write the Date_Time system tag to the communication system specified with the argument.

→ The system structure is not written synchronously but with the time interval of the Status refresh time.

Number of the Function	151 (97h)
Executable from the PLC	No
Function with ON/OFF state	No
Parameter	Communication number (0...7) specifying the PLC to which the Date_Time system tag should be written.
Function available in Standard screens	Yes
Function available in keyboards	No
Function available in script	Yes
Function available in application entry script	Yes

20.3.53 WriteTimeFromValues

Sets the current system time based on tags defined as parameters.

Number of the Function	544 (220h)
Executable from the PLC	No
Function with ON/OFF state	No
1. Parameter	Hour: Tag with hour information
2. Parameter	Minutes: Tag with minutes information
3. Parameter	Seconds: Tag with seconds information

Function available in Standard screens	No
Function available in keyboards	No
Function available in script	Yes
Function available in application entry script	Yes

20.3.54 WriteWeekdayToValue

Writes the current weekday to a tag defined as parameter.

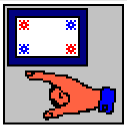
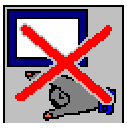
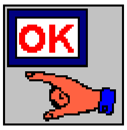
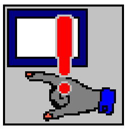
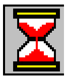
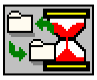
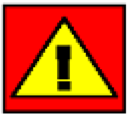
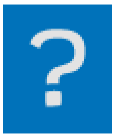
The weekdays are numbered according ISO 8601:

Monday:	1
Tuesday:	2
Wednesday:	3
Thursday:	4
Friday:	5
Saturday:	6
Sunday:	7

Number of the Function	545 (221h)
Executable from the PLC	No
Function with ON/OFF state	No
1. Parameter	Weekday: Tag used to store the current weekday
Function available in Standard screens	No
Function available in keyboards	No
Function available in script	Yes
Function available in application entry script	Yes

20.4 System Bitmaps

The system comes with the following system bitmaps for XV100 and XV400 touch panel:

  	These three bitmaps are used for Touch Disable/Enable. See also:→ section "Select Touch mode", page 1130
	This icon appears if the system cannot detect the touch operation correctly. This can be due to an operating error, a technical fault, or a soiled display. The display will return to normal 1-2 seconds afterwards.
	This bitmap will appear when the following functions are executed: - Special Function: Save History to File - Special Function: Clear History - Special Function: Save Graph Archive - Special Function: Load Graph Archive
	This bitmap appears during the execution of functions load recipe and save recipe.
	This bitmap appears, if the state of the object cannot be indicated correctly. For example, if the tag is outside of the permissible range.
	This bitmap is used only in connection with the ruler in Graph objects. This bitmap indicates a non valid curve value at the current ruler position.

20.4.1 Select Touch mode

There are three touch modes that can be used in order to prevent undesirable touch manipulations on the device screen:

- TouchEnable
  The device operator will be able to operate the touch display.
- TouchLock
  The touch display will not accept any more touch operations.

- TouchDisable
  In this touch mode, no touch operations will be accepted on the touch display.

20.4.1.1 TouchLock

The Set Touch Lock special function can be used to make the device screen switch to this touch mode.

Alternatively, the device screen can be made to switch to this mode by going to *Project Tree\Screens\Standard screens\Properties '<name of the screen>'* and enabling the Touch lock with timeout checkbox.

20.4.1.2 TouchEnable after TouchLock

A distinction must be drawn between different device touchscreen technologies when it comes to activating the touch display again (TouchEnable):

Infra-red touch

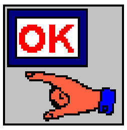
XV100 devices

The touchscreen needs to be touched in two places that are diagonal relative to each other (double finger touch) in order to be activated again. The minimum distance between these two places is approx. 5 cm.

Resistive touch

XV100 devices

The touchscreen needs to be touched at all 4 corners of the screen, clockwise or counterclockwise, in order to be activated again.



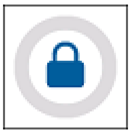
If the touch locked (TouchLock), then this bitmap appears during the "double finger touch" on IR touch or the "touch in all 4 corners" on Resistive touch devices.

Capacitive touch

XV300 touch panels

To activate XV300 touch displays again (TouchEnable), follow the steps below:

▶ Touch the device screen anywhere until the icon below appears:



▶ Then touch the icon with your finger and hold until the gray circle becomes completely blue and the icon disappears.

The touchscreen will be reactivated.

→ Touching this icon will not do anything

.

20.4.1.3 TouchDisable / TouchEnable

The touch display can be deactivated (TouchDisable)
by setting bit 1 of the 3rd data word in the Control structure.

To reactivate the touch display (TouchEnable), set bit 1 back to 0.

→ The bit 1 of the 3rd data word can be set statically.
While the bit is set, the touch display is in the "TouchDisable" mode.

→ "Set Touch Lock", page 1122
→ "Controlling functions with the PLC", page 602

20.5 About the application

20.5.1 Limits per project

Languages

Max. quantity of Languages	100

Screens

Max. quantity of Screens	65'000
Max. number of alarm list visualization objects per screen	1
Max. number of Extended Error Information Window objects per mask	1

Container

Max. size of all contained User Controls in pixels	32768 px
Horizontal orientation = Width of all User Controls Vertical orientation = Height of all User Controls	

Error Messages

Max. number of error message in the history	4'096
Max. number of error entries and alarm list entries	65'535

Texts

UNICODE supported	Yes
Interchange format for texts	*.csv

Communication

Max. number of communication interfaces	8

Variables

Max. number of tags (Bit, Byte, Word, Double Word, Float, Error Bit)	4'294'967'295
Available tag formats	binary decimal decimal +/- hexadecimal octal BCD hh:mm / mm:ss

Scripts

Maximum number of compiled loop scripts (all loop scripts)	Depending on device
Maximum number of compiled script lines of an event script	Depending on device

User Management

Max. number of groups	200
Max. number of users	500

Recipe limits

The following limits apply to recipes:

Objects Recipe	Max. amount per project	32
Recipe entries	Max. amount per recipe object	9'999

Variables	Max. amount per recipe entry	2'000

Graphs

Max. number of graph blocks	100
Max. number of tags per graph visualization object	255
Smallest scan rate	1 second
Max. number of measured values (enhanced mode)	RAM memory size in bytes[1]
Max. number of archive files	65,535[2]

[1] The maximum number of measured values depends primarily on the RAM required. The size of the RAM memory depends on the device model and device configuration being used. The amount of RAM memory available for graph blocks will depend on the project size, texts, recipes, screen files, and screen size. The RAM memory required for each graph block is calculated as follows: RAM memory (in bytes) = (22 + number of measured values * (8 + number of curves * 4))

[2] The memory required for archive files must be available.

20.5.2 Shortcuts - Project commands and key combinations

Project commands

Project commands will be available as long as no other dialog box is open.

F5	Create
F6	Building and deploying
CTRL + S	Save project
CTRL + Z	Undo last operation
CTRL + Y	Redo
CTRL + F1	Minimizes/restores the ribbon
ALT	Shows alternative shortcuts for the ribbon

General key combinations

General key combinations (available at different places):

CTRL + A	Select all
CTRL + C	Copies the selected object(s)
CTRL + Insert	Copies the selected object(s)
CTRL + V	Pastes the selected object(s)
CTRL + X	Cuts the selected object(s)
SHIFT + Insert	Pastes the selected object(s)
DELETE	Deletes the selected object(s)
F1	Opens context-sensitive help

Key combinations

Key combinations for working with visualization objects on screens:

Mouse + Ctrl key held down	Used to select multiple visualization objects
CURSOR buttons	Used to move the visualization object on a grid
ALT + CURSOR button	Used to move the visualization object with pixel accuracy
SHIFT + CURSOR button	Used to stretch / shrink the visualization object with pixel accuracy
ALT + SHIFT + CURSOR button	Used to stretch / shrink the visualization object in line with the grid
CTRL + G	Groups the selected visualization objects together
CTRL + U	Ungroups the selected visualization object group
CTRL + L	Aligns all selected visualization objects with the left edge of the visualization object that was first selected (blue corners)
CTRL + R	Aligns all selected visualization objects with the right edge of the visualization object that was first selected (blue corners)
CTRL + T	Aligns all selected visualization objects with the top edge of the visualization object that was first selected (blue corners)
CTRL + B	Aligns all selected visualization objects with the bottom edge of the visualization object that was first selected (blue corners)

CTRL + W	Adjusts the width of all selected visualization objects to match the width of the visualization object that was first selected (blue corners)
CTRL + H	Adjusts the height of all selected visualization objects to match the height of the visualization object that was first selected (blue corners)
+	Moves the visualization object to the front one level (Z order)
-	Moves the visualization object to the back one level (Z order)
CTRL + +	Moves the visualization object to the foreground (Z order)
CTRL + -	Moves the visualization object to the background (Z order)
TAB	Moves the selection between visualization objects (down the Z order)
SHIFT + TAB	Moves the selection between visualization objects (up the Z order)
CTRL + F6	Switches to the next open screen (or script)
CTRL + SHIFT + F6	Switches to the previous open screen (or script)
CTRL + F4	Closes the currently open screen

Script Editor

Keyboard shortcuts for working with the Script Editor

CTRL + SPACE	Shows suggestions for the current cursor position
CTRL + F	Opens the search dialog box

Project tree

Key combinations for working in the project tree (Screens, Tags, etc.):

F2	Rename
+	Expands the content of a folder or tag structure
-	Collapses the content of a folder or tag structure

Tables

Key combinations for working in tables:

Mouse + Ctrl key held down	Can be used to select multiple individual cells
Mouse + Shift key held down	Can be used to select a continuous range of cells

20.5.3 Elements encompassed by styles

The properties that can be saved in a style will depend on the specific type of visualization object to which the style applies.

Button, Screen Change, Function Key

- 3D/2D/No Frame display style
- Appearance at lower limit
- Appearance within range
- Appearance at upper limit

Value display

- 3D/2D/No Frame display style
- Appearance in ON state
- Appearance in OFF state

PLC Text

- Font
- Font Size
- Text color
- Normal/Bold/Italic format
- Text Alignment
- 3D/2D/No Frame display style
- Word Wrap
- Blinking Text
- Background color
- Blinking Background

Slider

- 3D/2D/No Frame display style
- Background color
- Fader size
- Fader color

Bargraph

- 3D/2D/No Frame display style
- Upper fill color
- In range fill color
- Lower fill color
- Flashing/Non-flashing fill color
- Tick color
- Background color

Date/Time Display

- Font
- Font Size
- Text color
- Normal/Bold/Italic format
- Text Alignment
- 3D/2D/No Frame display style
- Word Wrap
- Blinking Text
- Background color

Style Limitations

When you create a style, you will be able to apply it to any visualization object belonging to the corresponding type. This also means that styles, as a general rule, cannot be applied to more than one visualization object type. For example, you can assign the style for a button to another button, but not to a PLC text.

As an exception to the general rule, the following visualization object types can share styles:

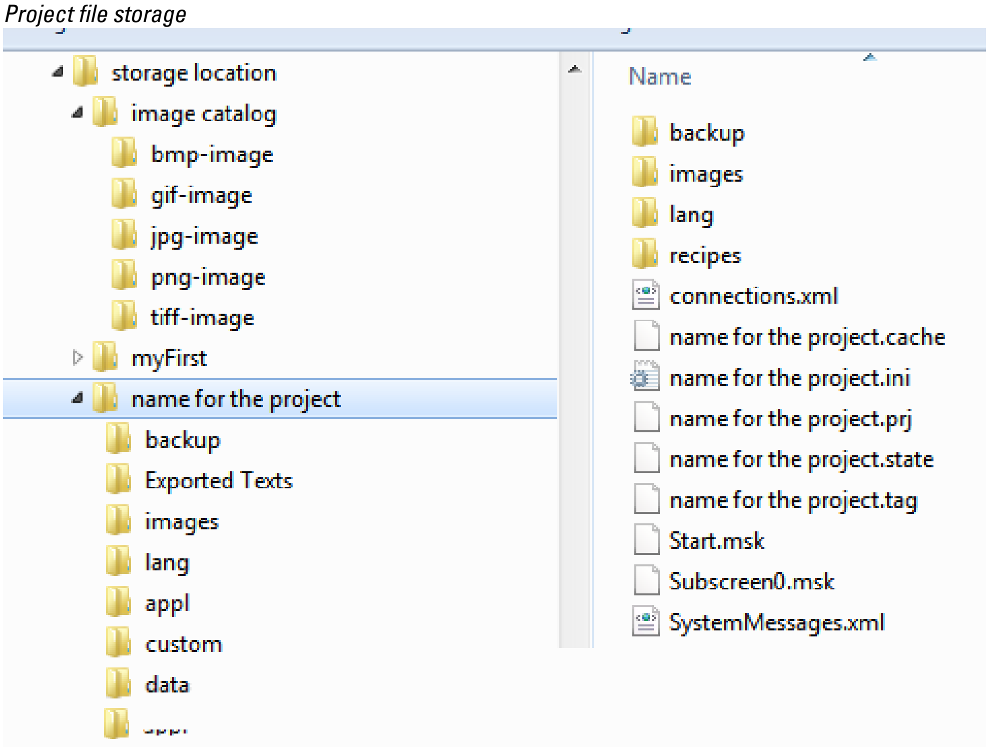
- Button
- Screen Change
- Function Key

20.5.4 File and directory structure

Project generation

A project consists of the following files, which will be stored in the project's storage location for the project:

Project file storage



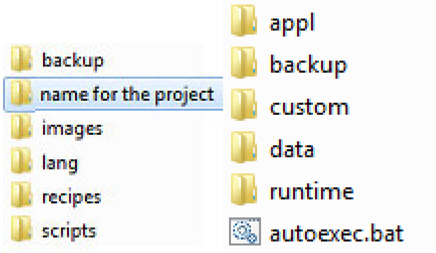
storage location
image catalog
bmp-image
gif-image
jpg-image
png-image
tiff-image
myFirst
name for the project
backup
Exported Texts
images
lang
appl
custom
data

Name
backup
images
lang
recipes
connections.xml
name for the project.cache
name for the project.ini
name for the project.prj
name for the project.state
name for the project.tag
Start.msk
Subscreen0.msk
SystemMessages.xml

Figure 651: View in the computer's file browser

A subdirectory with the project's name will be created in this directory and will be used to store the data used for transfers to the touch panel.

Location of created project files for deployment to the panel



backup
name for the project
images
lang
recipes
scripts

appl
backup
custom
data
runtime
autoexec.bat

Figure 652: View in the computer's file browser

20.5.4.1 Project files

A project in GALILEO consists of the following files:

.PRJ	Project file (in the compilation: Suffix *.GFX)
	- PLC definitions
	- Project settings
.cache .PRH	Help file
LANG\.XML	Text files
FONTS\.TTF	Fonts
IMAGE\.PNG	Image files
.msk	Screen files, each screen consists of one file (in the compilation: suffix *.MGF)
	- Objects and their settings
	- Mask settings
.tag	Tag file
	- Configurations of all tags (in the compilation integrated in file *.GFX)
.PWM	Password file
	- all user and password settings
.REF .ACT .TAB	Data for the recipe object (this file only exists in the compilation)
.PCL	Printer settings for the PCL driver on the Device
.INI	List with files of the CE components.
script/.scrp	Script file
	- Contents of all scripts

	- Script settings (in the compilation integrated in file *.GFX)
SystemMessages.xml	System Messages in all System message languages
connections.xml	Used to manage FTP and folder settings

Appendix

A.1 Further usage information

In order to be able to take full advantage of the online documentation's functionality, your computer's download manager must be configured in such a way that clicking on links will automatically open the corresponding Internet pages and PDF documents via FTP.

Hardware

For information on the devices on which GALILEO can be run, please refer to the following manuals:

User manual Engineering for XVM400	MN04802027Z
Operation manual HMI-PLC XV200	MN04802007Z
Operation manual HMI-PLC XV-102	MN04802004Z
Operation manual HMI-PLC XV-112	MN04802005Z
Operation manual HMI-PLC XV-152	MN04802006Z
Operation manual HMI-PLC XV200	MN04802007Z
Manual XV300	MN048017
Manual XV360 Single-Touchdisplay	MN048025
XP-503 Panel PC manual	MN048014
Manual for device series Industrial PC XP-504	MN048028

Software

For information on the device operating systems on which GALILEO can be run, please refer to the following manuals:

System description Windows CE3/CE5 for Touch Panel XV100, XV400, XC150	MN05010007Z
Windows Embedded Compact 7 System Description for Touch Panel XV300	MN050004

Communication

After being installed, the GALILEO application will be run on a device that can communicate with various PLCs.
In order for the application to work properly, you will need to configure additional settings based on the controller(s) being used.

The following documents, together with other documentation, explain what needs to be taken into account and configured:

Networks in Brief	MN05010009Z

You can open the Communications User Manual directly with ▤ \ ? Help and Tools as a .PDF document in English.

For further usage information on the various communication systems, please refer to the system descriptions provided by the relevant manufacturer.

Tutorials

You can directly access helpful videos that can help you better understand how to use specific functions by clicking on ▤ \ ? Help and Tools.

These tutorials will be opened outside of GALILEO.
You will need to have a compatible viewer installed on your operating system and/or Internet access.

Information for experts

For information on migrating existing GALILEO 8.1 projects to GALILEO, as well as on supported protocols, please refer to the Galileo Migration Guide PDF document that was stored in the selected folder on your computer when GALILEO was installed.

PDF Galileo Migration Guide.pdf
C:\Program Files\Eaton\Galileo 10\doc (default installation path)

The Galileo Migration Guide provides help when porting projects from GALILEO 8.1 to GALILEO. GALILEO will automatically perform and configure many of the required conversions and applications during the import process. However, you may need to make certain changes by hand in certain cases. The Galileo Migration Guide is designed to provide assistance in these cases.

Download Center

Click on the Customer Support tab to get to the Download Center - Documentation page

🌐 Eaton.com/documentation

For more information, please consult the corresponding product page on the Internet.

🌐 Automatization Software GALILEO
Eaton.com/GALILEO

A.2 What happens If ...

This section provides troubleshooting information for GALILEO in case it does not behave as expected.

Generally speaking, it is necessary to distinguish between cases in which the problem generates a message and cases in which it does not.

As long as the Automatization Software GALILEO program is working, a message will be displayed for assistance purposes on the user interface.

If no message is generated, it is possible that the error ("freeze," "crash") has to do with your computer's operating system.

The system states described here refer to situations in which the Automatization Software GALILEO program is not working.

If a message is output, it will be necessary to determine which system has generated the message – the computer's operating system, the Automatization Software GALILEO program itself, the project being worked on, or, if applicable, the device connected via the PLC.

→ "Messages from GALILEO", page 1145
→ "System Messages", page 1147

A.2.1 Messages from GALILEO

Messages during installation

Messages generated by the Automatization Software GALILEO program during installation will only consist of prompts concerning:

- An update
- Installation as a DEMO version
- The storage location

The → section "Installation instructions", page 42 explain when these prompts will appear.
Other messages will be generated by the PC's operating system.

Messages after transferring projects

If the Automatization Software GALILEO program is not licensed, the word DEMO will be shown on the application.

In order to be able to use the product, a current product license must be found on your system. If there is no current license on your system, you will only be able to run the program in demo mode or you will be prompted to enter a license.

GALILEO behavior

Symptom

Pane missing, properties have disappeared, view layout different from usual layout

Cause

Switching between two monitors
Closing panes by accident

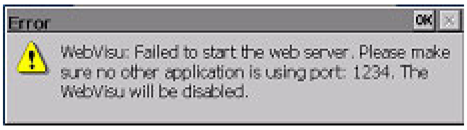
Remedy

Use the following command:
Home\Reset Layout
And
→ section "Customizing the application's user interface", page 131

Behavior in conjunction with WebVisu

Error message on the PC

Symptom



Error

WebVisu: Failed to start the web server. Please make sure no other application is using port: 1234. The WebVisu will be disabled.

Cause

You are attempting to start two web-server-based services on the same port.
For example: If <Web only> or <Web and local> is selected in *Home\Settings\WebVisu tab* in the GALILEO application and port <1234> is selected, the GALILEO web service will be active on this port.
If a web server is already enabled in the "Webserver" Config Tool tab on the touch panel and port "1234" is also being used there, there will be two web servers attempting to communicate on the same port, which is not possible.

Remedy

Disable the web server in the touch panel Config Tool or use a different port.

A.3 System Messages

System messages are output on the device at runtime. The reason behind these messages can be an incorrect configuration, e.g., when the Set Touch Lock function is added to a screen and run at runtime even though the device does not feature any touch zones.

The reason for these messages can also be the device's limits or its status, e.g., when the <Printing failed (not enough memory!)> system message appears.

System message languages

System messages are available in the five system message languages:

- German, English, Italian, Spanish, and French

Additional system message languages can be defined as necessary. In order for the system messages to appear in a newly added system message language at runtime, the language must be selected for the project language in the System messages language field.

System message languages can be added and edited in *Home\Settings\System Messages tab*, and the process is described in → Chapter "A Project Configuration", page 196.

To select the system message language that will be used by the project language, click on *Home\Texts\Languages*.

If you implement a function for changing languages, device operators will be able to switch between project languages and, accordingly, between system message languages at runtime.

All the system messages in the various system message languages are saved as an XML file when the project is saved. If you want to make overall changes, please make them in GALILEO and not in the file.

Types of system messages

System messages are always divided up into error messages and information messages.

About	Information messages are shown in green. These messages provide the operator with information. For example: 60) Successfully created user '%1'
Error	Errors are displayed in red. These system messages appear in the event of a malfunction on the panel or an operator error. For example:4) Project entry screen (%1) not found !
Fatal error	Fatal errors are shown in red A fatal error will be generated in the event of touch panel or project states that prevent the program from running correctly. In most cases, an error message of type error will appear first, after which the GRS will be closed. After this, the designer can reload faulty project data onto the device. For example: x) Touch is dirty or defect!

No more memory available

Under certain circumstances, "**No more memory available**" may also be shown in addition to various system messages. This means that there is not enough memory available for the application and that the GRS cannot be run correctly. To reduce the amount of memory required, check the following in the project:

- Graph messages: Reduce the number of measurements
- Reduce the number of fonts per project language

Unsupported DPI settings

GALILEO has been optimized for a default DPI setting of 96 (pixels per inch). The visualization interface may be distorted if a different display resolution is used.

→ Set the screen resolution on your PC to 96 DPI (100%). E.g., in the Display settings.

Table 6: List of system messages

ID	Catalog Number	English	German	Comment/Solution
0	About	1%	1%	The texts for the system messages with ID 0 and ID 1 will be generated only when an information message or fault is generated from the system. These texts are available in addition to the set system messages for IDs 2 to 66.
1	Error	1%	1%	
2	Error	Can't find or read from file '%1'	Kann Datei '%1' nicht finden oder lesen	The stated file has been destroyed or is not present. Reload the current project to the device. The SD-/CF card may have to be reformatted.
3	Error	Can't read from text number <%1!d!> from file '%2' !	Kann ab Text Nummer <%1!d!> in Datei '%2' nicht lesen !	The language file does not have the correct XML format. Solution: Open the file with a text editor (e.g. Notepad) and check the entry of the stated text number. Example of a correct text entry: `-<standard textID="15" Guid="50823b28-349d-4398-a45d-9e3371acb793"> <![CDATA [This text is in Courier New.]]> </standard>`
4	Error	Project entry screen (%1) not found !	Projekt-Startmaske (%1) nicht gefunden !	Check in *Project Tree\Screens* to see which standard screen has been set as the startup screen and check that the files are being successfully transferred to the device (do not forget to compile the project!).

ID	Catalog Number	English	German	Comment/Solution
5	Error	No matching file found (%1) !	Keine entsprechende Datei gefunden (%1) !	The stated file has been destroyed or is not present. Solution: Reload the current project to the device. The SD-/CF card may have to be reformatted.
6	Error	No more memory available for function %1 !	Kein Speicher für Funktion %1 verfügbar !	The global system memory has no more space. Note "%1" and contact Support.
7	Error	Some texts in text file '%1' are missing !	Es fehlen einige Texte in der Textdatei '%1' !	The text file on the target system does not match the active application. Solution: Load the project onto the device again by using the "Clear destination directory before download" download option.
8	Error	Accessing '%1' failed with error '%2' !	Zugriff auf '%1' fehlgeschlagen mit Fehlermeldung '%2' !	Certain functions cannot be run in conjunction with the operating system. Note "%1"/"%2" and contact Support.
9	Error	The function < %1 > is not supported !	Die Funktion < %1 > wird nicht unterstützt !	The parameter can be used to restrict the source of the error. Solution: If a decimal number (incl. hex number in parentheses) is specified as the parameter, this means that an unknown function was triggered. Not all functions are supported by all hardware models. Only selected functions can be executed by the PLC.
10	Error	Touch is dirty or defect!	Touch ist verschmutzt oder defekt!	During the device's initialization phase, the system detected that the touch filter was excessively soiled or that too many IR channels (infrared light barriers for touch interpretation) were being disrupted. This means that the device's normal operation is no longer guaranteed, as large areas cannot be operated. Solution: 1. On IR devices: • Clean Touch filter and restart Touch Panel. • If the Touch filter is not dirty or covered, the Touch Panel must be sent for repair. 2. In the case of resistive and capacitive devices: • Recalibrate touch.

ID	Catalog Number	English	German	Comment/Solution
11	Error	Can't create or write to file '%1' !	Kann Datei '%1' nicht erstellen oder beschreiben !	The specified file could not be saved due to one of the following reasons: • Not enough memory available • No write permissions • No SD/CF card plugged in Solution: Insert the SD-/CF card correctly and restart GRS. The project may require a larger-capacity SD/CD card or hard disk, or an additional memory module. The memory expansions possible depend on the device type concerned.
12	Error	Header information in file is invalid (%1) !	Header-Information in Datei ist fehlerhaft (%1) !	The stated file is damaged or it was not generated by the GALILEO compiler. Solution: Recompile the project and load it once more onto the device. The PCMCIA card may have to be reformatted.
13	Error	Recipe with id < %1 > not found	Rezept mit der ID < %1 > nicht gefunden	There is no recipe with this ID number in your project. Solution: Check whether the right recipe ID was passed in a function call. If necessary, compile the project again and load it onto the device again.
14	Error	Can't save recipe with empty name !	Kann Rezept mit leerem Namen nicht speichern !	The message will appear only if you enabled the "No empty names are allowed" option under the "Rules for entry names" section in the Recipe Editor dialog box. A name was not assigned to the recipe during saving.
15	Error	The recipe name '%1' already exist ! Choose another name.	Der Rezeptname '%1' besteht bereits ! Wählen Sie einen anderen Namen.	The message will appear only if you enabled the "No identical names are allowed" option under the "Rules for entry names" section in the Recipe Editor dialog box. The recipe was assigned a recipe name that already existed during saving.
16	Error	Execution of function %1 has failed (%2) !	Ausführung der Funktion %1 ist fehlgeschlagen (%2) !	A function that was started manually or from a script could not be executed correctly. Solution: Check and correct the function concerned or ensure the necessary conditions; e.g. necessary hardware is connected and operational.

ID	Catalog Number	English	German	Comment/Solution
17	Error	Error in script %1 !	Fehler in Skript %1 !	The specified script has an error that was not detected during compilation. Solution: Check and correct the script concerned.
18	Error	Can't read from file '%1' - structure mismatch!	Datei '%1' kann nicht gelesen werden - Falsche Inhaltsstruktur!	The stated file has been destroyed or is not present. Solution: Reload the current project to the panel. The SD-/CF card may have to be reformatted.
19	Error	No user authorization for %1 !	Keine Zugriffsberechtigung für %1 !	Not used.
20	Error	Communication disturbed: '%1' !	Kommunikation gestört: '%1' !	Data could not be sent to the PLC or read from the PLC. The cause of the error is indicated in the form of a dynamic parameter. Solution: Check the connection cable between device and PLC . Check the parameter configuration in your PLC. You can open the Communications User Manual directly with \ ? Help and Tools as a .PDF document in English. For further usage information on the various communication systems, please refer to the system descriptions provided by the relevant manufacturer.
21	Error	Not enough disc space available (%1 kB)	Nicht genügend Speicher vorhanden (%1 kB)	Insufficient memory available on the flash card or the hard disk to save the specified data size. This error may occur when converting a recipe database since the recipe data must be stored temporarily on the memory medium. Solution: Use a flash card with additional memory capacity larger than the stated value. In GALILEO, the memory size required is determined and shown during downloading.
22	Error	Protected or invalid path defined (%1) !	Geschützter oder ungültiger Pfad definiert (%1) !	The storage location for a file is a forbidden or protected path. Files are not allowed to be generated on the data storage device that the device uses to boot up. Solution: Adapt corresponding storage location in the GALILEO-

ID	Catalog Number	English	German	Comment/Solution
				Project.
23	Error	Printer not ready !	Drucker ist nicht bereit !	A printer function was started, however, the printer is not connected correctly, not switched on or not ready. Solution: Connect the printer to the correct panel port and check the data cable if necessary. It is also possible that there may not be any paper in the printer or that there is a printer error. Check the printer's status.
24	Error	Printer busy !	Drucker ist beschäftigt !	A printer function was started. The printer is still busy with a previous print function. Solution: Wait till the current printer function has been completed and start the printer again, see also → "Status data block", page 607.
25	Error	Incorrect Galileo registration code (%1) - DEMO mode !	Ungültiger Galileo Registrierungscode (%1) - DEMO Mode !	Either no license key, or an incorrect license key, was entered when installing the GALILEO development software program. The Device and PC are running in Demo mode. This message appears on startup. It must be acknowledge. It will appear again one hour later. After this, the Galileo Runtime System will abort with a fatal error. Solution: In GALILEO, go to *Home\License* and check the license key for GALILEO. The license key will be included when you purchase the product; refer to → "LicensingAutomatization Software GALILEO", page 40 as well.
26	Error	No valid runtime license found - DEMO mode !	Keine gültige Runtime Lizenz gefunden - DEMO Mode !	If the project will be running on a panel/PC, there must be a valid Runtime System license. Otherwise, the project will be fully functional and executable, but only for a limited time. Solution: Buy a valid Runtime System license from your Eaton distributor.
27	Error	Demo mode has expired! Runtime will shut down.	Zeit für DEMO Mode ist abgelaufen! Das Programm wird beendet.	If the project is running on a device or PC, the GALILEO license key must be valid and there must be a valid Runtime System license. Otherwise, the project will only be executable for a limited time.

ID	Catalog Number	English	German	Comment/Solution
				Solution: Check to make sure that the license key was entered correctly in GALILEO and on the device/PC.
28	Error	Printer < %1 > not detected !	Drucker < %1 > nicht gefunden !	A printer function was started. However, a printer was not defined or enabled in the project. Solution: Define or enable an existing printer in your project.
29	Error	%1 (%2)	%1 (%2)	A recipe entry or graph with this ID was not found. Solution: Load the project onto the device again. The Conflict Solver dialog box will appear. Select "Transfer to target" and confirm by clicking on OK. Caution! This will delete any recipe data on the device. If you want to keep this data, you will need to select "Ignore" in the "Conflict Solver" dialog box and carry out a recipe upload before loading the project onto the device again.
30	Error	Authentication failed: Username and/or password wrong	Authentifizierung fehlgeschlagen: Benutzername und/oder Passwort ungültig	The user entered an invalid username or password. Solution: Please note that the username and password are case-sensitive.
31	Error	Username is empty	Benutzername ist leer	The user did not enter a username.
32	Error	Password is empty	Passwort ist leer	The user did not enter a password.
33	Error	No group selected	Keine Gruppe ausgewählt	The user did not select a group.
34	Error	Your password has expired and must be changed	Passwort ist abgelaufen und muss geändert werden.	This prompt will appear if a user is member of a group for which the "Password aging" option is enabled.
35	Error	Too many failed login attempts, the account is locked	Zu viele ungültige Login-Versuche, das Benutzerkonto wurde gesperrt	The user is locked since several login attempts with an invalid password have failed. This prompt will appear if a user is member of a group for which the "Bad logins" option is enabled.
36	Error	Cannot create user: Number of users has reached the maximum	Benutzer kann nicht angelegt werden: Die maximale Anzahl der Benutzer ist erreicht	The maximum number of users has been created already. A maximum of 500 users can be created per group.
37	Error	Group '%1' does	Gruppe '%1' existiert nicht	You have entered an invalid group

ID	Catalog Number	English	German	Comment/Solution
		not exist		name. Solution: Use the "Show All Groups" function to see all existing groups. Reenter the group name. These passwords are case-sensitive.
38	Error	User '%1' does not exist	Benutzer '%1' existiert nicht	You entered an invalid username. Solution: Use the "Show All Users" function to see all existing users. Enter the username again. These passwords are case-sensitive.
39	Error	User '%1' already exists	Benutzer '%1' existiert bereits	You attempted to create a user that already exists. Solution: Use the "Show All Users" function to see all existing users. Enter the username again. These passwords are case-sensitive.
40	Error	New Password must differ from current one	Das neue Passwort muss sich vom bisherigen unterscheiden	Solution: Change the password.
41	Error	New Password already used in history	Passwort wurde bereits früher verwendet	The password history is active and checks the last x passwords. In other words, when changing the password, the last x passwords must no longer be used. When changing the password, you have used a password that is already contained in the password history. Value x can be entered in *Home\Settings\User Management\General\Passwords*, in the On change check the last field. Solution: Change the password.
42	Error	Cannot confirm new password: not matching	Neues Passwort kann nicht übernommen werden: keine Übereinstimmung	If a new password is entered, a multiple confirmation is required. For this the same text or number must be entered a second time. Different password entries will cause this error message to appear. Solution: Keep in mind that the password is case-sensitive. Enter the new password once more in the two lower fields.
43	Error	No active user present. Please login to perform action	Kein aktiver Benutzer eingeloggt	The executed function can only be run if a user is active (logged in). Solution: Log in first, then run the function again.
44	Error	User '%1' is locked	Benutzer '%1' wurde gesperrt	The user was locked due to one of the following reasons:

ID	Catalog Number	English	German	Comment/Solution
				• Multiple failed login attempts with an invalid password • The user was locked by a user from a different group with the "Lock/Unlock user" system permission. Solution: The user from a different group with the "Lock/Unlock user" system permission can unlock the user.
45	Error	Password is too short. Type at least %1 characters	Das Passwort ist zu kurz. Mind. %1 Zeichen sind nötig.	When the password is changed, too few characters are used. Solution: Set a new password with at least x characters. Value x can be entered in *Home\Settings\User Management\General\Passwords*, in the Minimum password length field.
46	Error	Cannot lock active user	Der aktive Benutzer kann nicht gesperrt werden.	The user is authorized to lock a certain user on the system, however he cannot lock himself. Solution: A different user from a group with the "Lock/Unlock user" system permission must log in in order to be able to lock this user.
47	Error	Cannot delete active user	Der aktive Benutzer kann nicht gelöscht werden	The user account of the currently logged in user must not be deleted. Solution: A different user from a group with the "Delete user" system permission must log in in order to be able to delete this user.
48	Error	No permission to create user in group '%1'	Keine Berechtigung zum Erstellen eines Benutzer in der Gruppe '%1'	You are not authorized to create a user account, the member of a particular group. Solution: Check whether you defined the right group with the necessary system permissions.
49	Error	No permission to delete user '%1' in group '%2'	Keine Berechtigung zum Löschen des Benutzers '%1' in der Gruppe '%2'	You do not have the permissions required in order to assign the current user to a different group. Solution: Check whether you defined the right group with the necessary system permissions. Check the way the username is written as well. These passwords are case-sensitive.
50	Error	No permission to change password of user '%1'	Keine Berechtigung für das Ändern des Passworts des Benutzers '%1'	You have no authorization to change the password of a user. Solution: Check whether you defined the right group with the

ID	Catalog Number	English	German	Comment/Solution
				necessary system permissions.
51	Error	No permission to change group of user '%1'	Keine Berechtigung für das Ändern der Gruppe des Benutzers '%1'	You have no permission to assign a group to the user. Solution: Check whether you defined the right group with the necessary system permissions.
52	Error	No permission to modify group '%1'	Keine Berechtigung für das Ändern der Gruppe '%1'	You do not have the permissions required in order to assign the current user to a different group Solution: Check whether you defined the right group with the necessary system permissions.
53	Error	No permission to lock user '%1'	Keine Berechtigung für das Sperren des Benutzers '%1'	The active user has no permission to lock the selected user account. Solution: You only have the right to lock user accounts for members of the defined target groups. Check whether you defined the right group with the necessary system permissions.
54	Error	No permission to unlock user '%1'	Keine Berechtigung für das Freischalten des Benutzers '%1'	The active user has no permission to unlock the selected user account. Solution: You only have the right to unlock user accounts for members of the defined target groups. Check whether you defined the right group with the necessary system permissions.
55	Error	No permission to create users on this device	Keine Berechtigung um Benutzer auf diesem Gerä zu erstellen	The active user does not have the required permissions for creating users on this device. Solution: Check whether you defined the right group with the necessary system permissions.
56	Error	No permission to delete users on this device	Keine Berechtigung um Benutzer auf diesem Gerät zu löschen	The active user does not have the required permissions for deleting users on this device. Solution: Check whether you defined the right group with the necessary system permissions.
57	Error	No permission to change passwords	Keine Berechtigung um Passwörter zu ändern	The active user does not have the required permissions for changing passwords. Solution: Check whether you defined the right group with the necessary system permissions.
58	Error	No permission to change group affiliations	Keine Berechtigung um Gruppenzugehörigkeiten zu ändern	The active user does not have the required permissions for changing group affiliations. Solution: Check whether you

ID	Catalog Number	English	German	Comment/Solution
				defined the right group with the necessary system permissions.
59	About	No permission to lock/unlock users	Keine Berechtigung um Benutzer zu sperren/freizuschalten	The active user does not have the required permissions for locking/unlocking users Solution: Check whether you defined the right group with the necessary system permissions.
60	About	Successfully created user '%1'	Benutzer '%1' erfolgreich erstellt	Notification indicating that the user was created.
61	About	Successfully deleted user '%1'	Benutzer '%1' erfolgreich gelöscht	Notification indicating that the user was deleted.
62	About	Successfully locked user '%1'	Benutzer '%1' erfolgreich gesperrt	Notification that the user was locked.
63	About	Successfully unlocked user '%1'	Benutzer '%1' erfolgreich freigeschaltet	Notification indicating that the user was unlocked.
64	About	Successfully changed password of user '%1'	Passwort für Benutzer '%1' erfolgreich geändert	Notification indicating that the user's password was changed.
65	About	Successfully changed group of user '%1'	Gruppe des Benutzers '%1' erfolgreich geändert	Notification indicating that the user's group was changed.
66	About	GraphBlock is running. Please stop to perform file loading.	Graph-Aufzeichnung läuft. Bitte stoppen um das Laden der Datei auszuführen.	An archive file cannot be opened while graph plotting is in progress. Solution: Stop the current graph plotting.

Glossary

Client

The term "client" refers to an application that requests specific services from a server.

*

***.avi**

Audio Video Interleaved

***.bmp**

Pixel-based file format for two-dimensional raster graphics

***.crt**

SSL certificate file

***.csv**

Comma-Separated Values (Character-Separated Values) Data format for text

***.DLL**

Dynamic link library

***.gdb**

GDB

***.itf**

Internal Tag Import Format

***.jpg**

Pixel-based file format for the JPEG (Joint Photographics Expert Group) image file format The JPEG format does not support transparency

***.key**

Private key for the SSL certificate

***.mov**

Movi

***.mp4**

MPEG-4 Codec, Moving Pictures Experts Group IV Video container format

***.png**

PNG (Portable Network Graphics) image file format for graphics and video software, The PNG file format supports transparency with its alpha channel

***.prj**

File format used by GALILEO project files

***.rtf**

Rich Text Format File format for texts

***.svg**

Scalable Vector Graphics

***.tiff**

Vector-based image file format for graphics and video software, The TIFF format supports transparency, as well as images using 8-bit channels (grayscale, RGB, CMYK, etc.)

***.zip**

ZIP file format used to compress and archive files

***xlsx**

Microsoft Excel file format - Excel computation table in XML format

A

Activation code

Device-tied number used to license one of the communication systems.

Address reference

The term "address reference" refers to the data packet's start address.

Alpha channel

Transparency information for PNG images Used to specify the degree of transparency for each pixel

Application

Short for "application software," a computer program that performs a function useful to the user.

B

Background Image

Image that fills up the entire screen

Bitmap

Image file in the BMP raster graphics image file format.

Boot

Booting up, starting (up) - automatic process that takes place after the device is switched on, and in which a simple program in ROM memory starts a more complex program.

C

CBA

Communication Board Adapter

CF-card

CompactFlash card

CIS

Card Information Structure

Color model

All colors must be described in a manner that can be processed. All computer systems work with binary data, i.e., with numbers. Color models that describe all the colors that can be generated by a device and that are based on the specific device's color elements are referred to as "device-dependent."

Comboard Loader

Loading program integrated into GRS and used to update modules with loadable firmware.

Command sequence

Path information List of the commands that the device operator must tap in succession in order to get to the location described; for example: Start\Project Overview\Variables folder.

Communication

The transfer of data between the panel and the PLC, controller, or peripheral connected to it.

Communication Board Adapter

In WinGRS, this adapter is the runtime license.

CompactFlash card

CompactFlash (CF) is an interface standard for non-volatile, rewritable data memory in the form of CF cards, which are used with Eaton touch panels. Data written to these cards is stored in a non-volatile manner that does not require any additional (secondary) power.

Compilation

The process of transforming project data to an executable user interface program for the specific panel model / to a GRS-specific data structure

Compile

The process of transforming project data to an executable user interface program for the specific panel model / to a GRS-specific data structure

Control data block

Together with the status data block, the control data block is used to monitor and display the status of data transfers between the HMI and PLC.

D

Data packet

The term "data packet" refers to a variable, array, or structure that has been addressed in Galileo. These data packets are always transmitted to the PLC as a unit.

Data Set Ready

The transmitter is ready to send data.

Data synchronization

Routine used to ensure that all the modules connected to the HMI have the same data (presetting).

Data Terminal Ready

The receiver is ready to receive data.

Data Types

Depending on device

All colors must be described in a manner that can be processed. All computer systems work with binary data, i.e., with numbers. Color models that describe all the colors that can be generated by a device and that are based on the specific device's color elements are referred to as "device-dependent."

Destination Directory

File path on the computer or touch panel; configured destination for the project

DHCP (used to obtain an IP address automatically)

You can enable this setting if you do not want to configure every single individual computer within a network, provided there is a DHCP server on the network. When this setting is enabled, the computer will get information such as an IP address, subnet mask, gateway, and DNS from the DHCP server. In most cases, the router used on a network will also feature a DHCP server.

Direction URL

Uniform Resource Locator

DNS (Domain Name Server)

When you enter an address such as www.intel.com into a browser or FTP client, your computer will first need to ask a server for the IP address behind the name in order to actually be able to reach the address. The server that provides this information is known as a "domain name server." Every single Internet provider provides this service,and most providers have a secondary DNS in case their primary DNS fails. DNS records are the IP addresses for these servers.

Drag and drop

Windows functionality Used to select and position items by dragging them with the mouse

Drop-down

List with options available for selection

DSR

Data Set Ready

DTR

Data Terminal Ready

Dynamic Control

Elements (objects and tools) that display a change based on an action on the HMI or a change in the data on the PLC (e.g., button, value display or input field, error window, etc.).

E

EATON

Eaton is a power management company that provides energy-efficient solutions that help our customers effectively manage electrical, hydraulic, and mechanical power. A global technology leader, Eaton acquired Cooper Industries plc in 2012. The 2012 revenue of the combined companies was $21.8 billion on a pro forma basis. Eaton has approximately 102,000 employees and sells products to customers in more than 175 countries.For more information, please visit www.eaton.com.

Eaton Automation

The integration in the Eaton group has led to the formation of a new business unit offering power distribution and automation solutions from a single source. Solutions aimed at both the panel building and the mechanical engineering. Eaton Automation offers HMI, PLC, HMI/PLC, Remote IO and SmartWire-DT in the field of industrial automation. Industrial automation is complemented by industrial switchgear from Eaton with contactors, motor-protective circuit-breakers, frequency inverters, soft starters, control circuit devices and circuit-breakers. The product range is designed for machine and panel building solutions. We cater for the demands of our international customers with our systematic product conformity to UL/CSA, CCC and IEC standards. We are present on a global level just like our customers with sales subsidiaries in more than 150 countries and development centers in Asia, America and Europe.

Eaton Electrical

Eaton’s electrical business is a global leader in power distribution, power quality, control and industrial automation, power monitoring, and energy management products and services. Eaton's global electrical product lines, including Cutler-Hammer®, Moeller®, Powerware®, Holec®, MEM®, and Santak® provide customer-driven solutions to serve the power system needs of the data center, industrial, institutional, government, utility, commercial, residential, and OEM markets worldwide.

Error tag

System message

Error texts

Texts shown in the error window in projects. These texts can consist of several lines.

Error variable

Variable with the Error or Error Array data type. When this variable has a value of "1", this means that there is an error.

F

FAT

File Allocation Table

File Allocation Table

FATs are used to define filesystems.

Firewall

Firewalls are used to prevent outside attempts to access IP addresses on a private network. In other words, they are used to protect internal data. When configured correctly, they can also be used to set up rules or lists that prevent specific URLs from being requested, e.g., when they are in violation of company policy. A firewall's main task is to use the information in a packet (the source and destination IP addresses, as well as the port) to decide whether the packet should be rejected or allowed to pass. This also prevents packets not meant for the network from subjecting the network to an unnecessary load, as well as packets meant for the private network from reaching the Internet.

Flash

Non-volatile memory on CompactFlash card or SD card.

FTP

File Transfer Protocol

G

GALILEO OPEN

GALILEO OPEN is a PC-based visualization program consisting of the Galileo Runtime System (WinGRS) and a communication module.

GALILEO Project Inspector (GPI)

Program for testing GALILEO projects directly on a PC. With a few exceptions, all of the HMI's functions can be tested and tried out on a PC.

GALILEO Runtime System (GRS

The term "GRS PLC runtime system" refers to the GRSW sequential programs on HMI devices. These programs interpret (compile) the data structure generated by the GALILEO code generation program. GRS2 for Windows CE devices with an x86 processor. The corresponding panel must have a parallel port and a PCMCIA card. GRSW3 for all Windows CE devices with ARM processors.

Gateway

Gateway When two computers on different networks want to communicate with each other, the networks need to be connected with a router. For example, surfing on the Internet requires for packets to be routed from the Internet to the network and vice versa. By using a subnet mask, a computer can know whether the receiver can be found on its network or whether it is located outside of it. If it is located outside the network, the computer will send a packet to the router specified with the gateway IP address.

GDB

Graph Block tag values in ASCII format

GPI

Galileo Project Inspector

GRS

Short for "Galileo Runtime System"

GRS2

The term "GRS PLC runtime system" refers to the GRSW sequential programs on HMI devices. These programs interpret (compile) the data structure generated by the GALILEO code generation program. GRS2 for Windows CE devices with an x86 processor. The corresponding panel must have a parallel port and a PCMCIA card. GRSW3 for all Windows CE devices with ARM processors.

GRSW3)

The term "GRS PLC runtime system" refers to the GRSW sequential programs on HMI devices. These programs interpret (compile) the data structure generated by the GALILEO code generation program. GRS2 for Windows CE devices with an x86 processor. The corresponding panel must have a parallel port and a PCMCIA card. GRSW3 for all Windows CE devices with ARM processors.

H

Help texts

Texts shown in the help windows in a project. These texts can consist of several lines.

History

Events stored as a result of error and warning changes.

HMI (MMI)

Human Machine Interface An operating terminal on which the project is installed as a user interface.

HSB model

Hue/Saturation/Brightness color model

I

IL

Installation instructions

Image sequence

Multi-frame file in *.gif image file format

Internal Storage

Internal touch panel memory

IP Address

IP addresses are 32 bits (4 bytes) long and are used to uniquely identify networks, subnetworks, and individual computers that work with the TCP/IP protocol. A distinction is drawn between private address spaces for local networks (intranet) and public addresses (Internet).

IR

Infrared

L

Language file

Each project can be translated into a maximum of 100 languages. Each translation will need to be stored in its own file, which is referred to as a "language file."

Log file

Log file used to store the actions carried out by the application program's users in a traceable manner.

M

MDI

Multi Document Interface

MN

Manual - Operation manual

Monospaced fonts

Fonts with a fixed width for all characters

Multi Document Interface

Feature making it possible to open multiple documents (screens, views) simultaneously in a Windows application (GALILEO)

O

Object

Static or dynamic element used for engineering purposes. Static objects are located in the view's background and do not change at runtime. In contrast, dynamic objects are located in the view's foreground, and their appearance can change as a result of data changes.

OOPM

Object-oriented programming and modeling

Operating system

A group of programs that control and manage the processes in a computer and its connected devices.

OS

Operating system

P

PCMCIA

Personal Computer Memory Card International Association (PCMCIA)

PCT

Projected capacitive touch

Peer to Peer (P2P)

Peer-to-peer is a term used for computers that are connected to each other in an architecture in which both computers can assume the role of server and client.

Personal computer

A personal computer is made up of a central processing unit, RAM, external data storage devices, an operating system, and application programs, and is connected to peripheral devices (monitor, printer). PCs can be stationary or portable.

Personal Computer Memory Card International Association (PCMCIA)

PCMCIA Card, also known as PC Card, is a standard for the expansion cards for portable computers used in Eaton touch panels. PCMCIA cards are energy-efficient and support hot plugging, i.e., they can be replaced during ongoing operation. Plug and play is supported thanks to the fact that all the properties required in order to automatically configure the driver are stored on the card's CIS.

PLC

Programmable logic controller The controller or peripheral that is connected to the HMI.

PLC(S)

Programmable logic controller The controller or peripheral that is connected to the HMI.

Polling

Cyclical reading of the PLC's addressed variables

Port

Ports can be seen as virtual mailboxes for data packets. A computer can communicate with other computers on 65536 different ports.

Project

File containing all the screens with configured objects and their assignments so that they can be loaded onto the HMI as a view unit

Project languages

Project languages The texts in a project can be simultaneously stored in more than one language on the HMI. However, only one of these languages can be displayed on the MMI at any one time. Special functions can be used to switch to a different language. For a more detailed description of these special functions, please refer to the GALILEO program.

Projected capacitive touch

A display designed for high precision, user friendliness, and durability. It is designed to bring the controls that have now become prevalent in consumer electronics to machines, with advantages such as a gesture-based user interface, two-finger multi-touch depending on the application software being used, intuitive operation that enables operators to start working right away, and the fact that no calibration is required

R

Recipe entry

Refers to a collection of values stored under a definable name in a recipe object. Multiple recipe entries are combined into a general management structure. Recipe entries can be edited in the *.TAB file. The file's name will be the same as the recipe object's name. The name of each entry will be in the second column of each row. A maximum of 1000 entries can be managed in each recipe object.

Recipe handling

Refers to a collection of recipe entries. The values for the corresponding variables can be stored on the panel using a specific name.

Recipe Object

Refers to a collection of recipe entries. The values for the corresponding variables can be stored on the panel using a specific name.

Registers

Subpages in a dialog box or object

Reset button

RFC 6335

Since 2011, with RFC 6335, a Request For Comments has described the procedures used by the Internet Assigned Numbers Authority (IANA) to assign ports to protocols. These ports are dynamically assigned to client programs by the operating system.

RGB model

Red/Green/Blue color model

Ribbon

Menu ribbon that can be expanded and collapsed and that provides the various available commands

ROM (read-only memory)

Non-volatile read-only memory

Router

Routers are devices used to forward ("route") requests from a network to the Internet (or to another network). Routers provide a measure of security for private networks, as nodes outside of the network will be unable to determine which specific computer requested the data. This is because all the computers on the private network will appear under the same IP address on the Internet.

Runtime

General designation for the GALILEO Runtime System

Runtime code

The compiled project used by GRS.

Runtime data

Data generated by the GRS during ongoing operation.

Runtime software

General designation for the GALILEO Runtime System

RxD

Receive cable for received data

S

Screen

View on the HMI that contains the objects configured in GALILEO (e.g., controls, indicators, function keys, etc.).

SD card

Secure Digital memory cards are non-volatile, rewritable flash data storage devices that are used with Eaton touch panels and are commonly referred to as SD cards. Data written to these cards is stored in a non-volatile manner that does not require any additional (secondary) power.

SD card slot

Secure Socket Layer

encrypted network connection between a server and a client (browser)

Server:

The term "server" is usually used to refer to computers that provide services on a network. Admittedly, however, this definition is not very precise. More specifically, servers are applications on a computer that are responsible for providing or processing data. In fact, every computer can provide such services. Servers are not active in and of themselves. They wait until they are addressed by a client, after which they perform the corresponding tasks. Each server application provides its service on the network via a specific port.

Show System Info

Special function that can be used to show a window with current system data in GRS

Shutdown

Special function that can be used to exit GRS

Sidebar

Controls Manager and Properties panes shown on the right side

Slot

Refers to a slot for a communication module in the HMI

Source project

The project currently open in GALILEO.

Specifications for connection to supply voltage

SSL

Secure Socket Layer

Standard texts

Texts that can be used in a general manner throughout a project, e.g., titles, button labels, etc. These texts can consist of several lines.

Start screen

The start screen that appears when the program starts on the HMI (device). The start screen needs to be defined as one of the standard screens in the project organization structure for each GALILEO project.

Static Control

An element (objects and tools such as lines, circles, rectangles, images, text, etc.) that does not change while the HMI is running.

Status data block

Together with the control data block, the status data block is used to monitor and display the status of data transfers between the HMI and PLC.

Status line

The status bar is located at the bottom of the program window. It is used to show whether the editing mode is in "overwrite" or "insert" mode, etc. In addition to the standard information shown by Windows, the currently selected text number will also be shown.

Stroke

A hub is a device used to connect various network devices together. Hubs broadcast all data to all connected devices (devices connected with a patch cable).

Style

Design template for visualization objects

Subnet mask

A subnet mask is an IP address "filter." It has the same syntax as an IP address. This mask defines which computers can transfer data between themselves within a network. This also means that subnet masks define the maximum size of the corresponding subnetworks.

Switch

Switches are networking devices that are more advanced than hubs. One of the main features that sets them apart from the latter is the fact that they are more "intelligent" and forward data packets much more efficiently by sending them only to the devices that need to receive them. Multiple data packets can pass through a switch at the same time. Among other things, this means that switches have a significantly higher total bandwidth (throughput) than hubs. Moreover, switches learn which stations are connected to which ports, meaning that additional data transfers will not result in any ports being subjected to unnecessary loads, i.e., that data will only be forwarded to the port connected to the intended destination. With the exception of their higher price, switches are superior to hubs in every way.

System character set

Font type and size used to output system messages.

System error message

These error messages are system-specific and are generated by GRS. The two basic types are error messages and information messages. Errors will be shown in a superimposed red window, while information messages will be shown in a green window. Fatal errors will cause the program to crash.

System History

For each system message, it is possible to define whether it should be stored with the corresponding date and time.

System port

Serial interface on touch panel.

T

Target System

PC or touch panel

TDB

Graph Block tag values in binary format

Toolbar

The toolbar provides all important functions so that they can be accessed directly. All the buttons in a toolbar can also be found as menu options in the menu.

Touch bezel

Connects IR elements to the touch controller

Touch Controller

Takes care of signal processing and evaluates all IR elements.

Touch filter

The plastic bezel behind which the IR elements are located.

Touch panel

An HMI that enables users to intervene directly in the process by touching the objects on the screen.

Touch zone

A rectangular area on the panel that can be custom-defined and that will respond to touch

Transfer parameters

Baud rate, data bit, start bit, stop bit, and parity

Transport Layer Security

encrypted network connection between a server and a client (browser)

TxD

Transmit cable for transmitted data

U

User

Operator using the device on which the user interface created with Galileo is running.

V

Variable

Defined data point on the HMI or PLC Addresses are used to link the PLC's and HMI's data points.

Variables

View

View on the HMI that contains the objects configured in GALILEO (e.g., controls, indicators, function keys, etc.).

W

WebVisu (web-based visualization)

HTML5-based visualization for PLCs

Widescreen

Widescreen format

Windows

Dialog boxes, prompts, etc. that open while the application is running and remain on the current program page Synonyms: dialogue box, dialog These windows are shown by the application in various situations in order to obtain specific input or confirmations from the user. Dialog boxes expect input from the user, while prompts are shown to get the user's confirmation for specific messages.

Windows CE Image

Operating system for Windows CE devices

Windows Embedded Compact 7 pro

A component-based, multi-functional real-time operating system designed to meet the needs of even the most demanding industrial applications. The operating system will unpack itself into the device's RAM memory every time the device is started, making it possible to de-energize the device without the need for a UPS or other similar precautions. As the successor to WinCE 6, Windows Embedded Compact 7 features a large selection of standard technologies, communication servers/clients, and web services. • Fast boot time • Multi-touch support (two-finger) • Long-term availability

WinGRS

Runtime system that can be run in Windows, but that requires an appropriate communication module as a license in order to be fully executable.

WINS

Windows Internet Name Service, Name resolution service within Microsoft networks. In order for this service to be used, there must be a WINS server. If there is no WINS server, names will be resolved using broadcasts and other mechanisms. A fixed name can be assigned to an IP address in WINS so that a computer will continue to be recognized even if its IP address changes.

Wizard

User interface wizard designed to guide users through multiple dialog boxes used to enter data or select parameters

Alphabetical index

Powering Business Worldwide

Eaton Industries GmbH
Hein-Moeller-Str. 7–11
D-53115 Bonn

08/2022 MN048018 EN (PMCC)