Engineering
Part 3: HMI and Operating Stations

English Version 3.0

Content

About this Manual

BatchXpert is a control system for managing and visualizing batch processes. It supports recipe execution, batch tracking (including material handling), alarm and event management, trend recording, and batch reporting via BatchXpert stations and HMI clients.
The system is modular and can be scaled from single units (for example, CIP or pasteurization skids) to complete production lines. In a typical automation project, BatchXpert integrates three layers—PLC control, SCADA/HMI visualization, and the engineering/database toolchain—that operate together as one system.

This manual is organized into three parts: **Part 1** provides a general overview of BatchXpert, **Part 2** focuses mainly on PLC-related topics and engineering in SIMATIC TIA Portal, and **Part 3** explains how to create the HMI and Operating Station. **Part 4** contains information about additional modules that can be used in the BatchXpert system.

Target Audience of this Manual

This manual is intended for project engineers and automation specialists who implement or maintain BatchXpert projects. It assumes familiarity with batch/process automation concepts and general commissioning practices. This document explains how BatchXpert applies these concepts; it is not an introduction to PLC programming. Working knowledge of Siemens SIMATIC S7 programming and the associated engineering tools (SIMATIC Manager and/or TIA Portal) is required.

AI-assisted editing notice:

This manual was produced with the support of AI tools. AI was used primarily to assist with grammar and spelling corrections and to improve readability. The technical content, procedures, and engineering guidance are based on the author's expertise and project knowledge, not generated by AI.

Version of this Manual

Version 3.0	Reworked completely Separated into multiple Parts

Overview of HMIs

The BatchXpert system primarily uses the VisXpert SCADA system for its HMI screens, which is integrated with BatchXpert. BatchXpert cannot function without at least one full VisXpert station available. All HMI field panels should be seen as an “auxiliary operating station,” more than a fully-fledged station. Only BatchXpert operating stations using the VisXpert system support the full functionality of BatchXpert, such as Recipe editing, Reports and others.
For this reason, the use of dedicated “traditional” Hmi touch devices such as Simatic Panels, are discouraged in favor of “Touch Panel PC’s”. These Panel PCs are running the same “VisXpert” SCADA system as all other Operating stations, and thus use the same project without modification, thus eliminating the need to maintain multiple HMI and SCADA systems synchronized.

Screen Resolutions

BatchXpert includes libraries for the following SCADA and HMI systems and resolutions. New Resolutions can be added, but the Templates must be adjusted manually to fit the screen size. VisXpert supports full screen scaling, enabling the user of different Screen resolution.

- **VisXpert** Visualization
  - 1920 x 1080 (Full HD)
  - 1600 X 900 (HD)
  - 1366 x 768 (Notebook)

As stated above, the use of traditional touch panels is discouraged in favor of Panel PCs. Thus, the following list of HMI devices is supported, but should not be used for new projects.

- **Movicon** Display for Touch Screens
  - 1024 x 768" (10")
  - 800 x 480 (7")

- **WinCC Flexible** for Simatic Touch Screens (without TIA Portal)
  - 800 x 480

- **WinCC Basic and WinCC Comfort** for Simatic Touch Screens (with TIA Portal)
  - 800 x 480
  - 1280 x 800

Adjust Screen Resolution

In VisXpert, the screen resolution is chosen based on the “BaseProcessWindow,” which comes with versions for different screen resolutions. These different windows have adjusted layout. To choose an “BaseWindow” and a corresponding Screen resolution, you must adjust the default base window in the “Settings” variables in the “VisXpert” variable editor. By default, the screen resolution is set to “Full HD” (1920 x 1080).

Since VisXpert Version 10, the HMI system supports scaling the Hmi screens to fit the current screen resolution. To take advantage of this, you must enable the corresponding setting in the Graphics editor of VisXpert.

This way the recommended solution is to create all process screens for a native “Full HD” resolution and then scale them down for stations that have lower resolutions. However, this scaling can result in suboptimal graphics, which can only be avoided by designing the Graphics display to the desired resolution.

For the other HMI Project templates, there are projects with different resolution prepared in the specific folders.

Multiple Monitors

Multiple monitors are supported by the VisXpert system by using the “MultiMon” VisXpert module. For details, please refer to the “Manual BatchXpert Multi Monitor Support” manual.

"Visu Extern" System (Touch Screens)

The BatchXpert system implements and sophisticated HMI customization mechanism that allows you to implement custom HDMI systems with custom status words and even customized data exchange. This communication channel is referred to as “Visu Extern” and allows you to fully customize your communication mechanism with your customized HMI solution. You can implement multiple different “Visu Extern” communication channels on the same PLC, thus supporting multiple HMI mechanisms at the same time.
This mechanism is used for simpler touch panels that do not run the full VisXpert SCADA system, for example WinCC Comfort or MoViCon Touch panels.

Since this communication mechanism builds upon the internal HDMI mechanism of batch expert, all historic data recording for manual interventions is still fully supported, even for external HMI systems.

Nowadays the use of “Visu Extern” for “Touch HMI Systems” is discouraged in favor of touch panel PC's running on the VisXpert SCADA System.

External HMI

This communication channel is commonly used for WinCC and MoViCon based touch panels come on but also text panels or even legacy “Mosaik” systems. The BatchXpert system separates between two types of Visualization (HMI)

- **Internal View:** The main system, integrated into the BatchXpert System. Only the "VisXpert" SCADA system is available.
- **External View**: All other HMIs. For example, Siemens WinCC Flexible, Progea Movicon, or other types of touch screens.

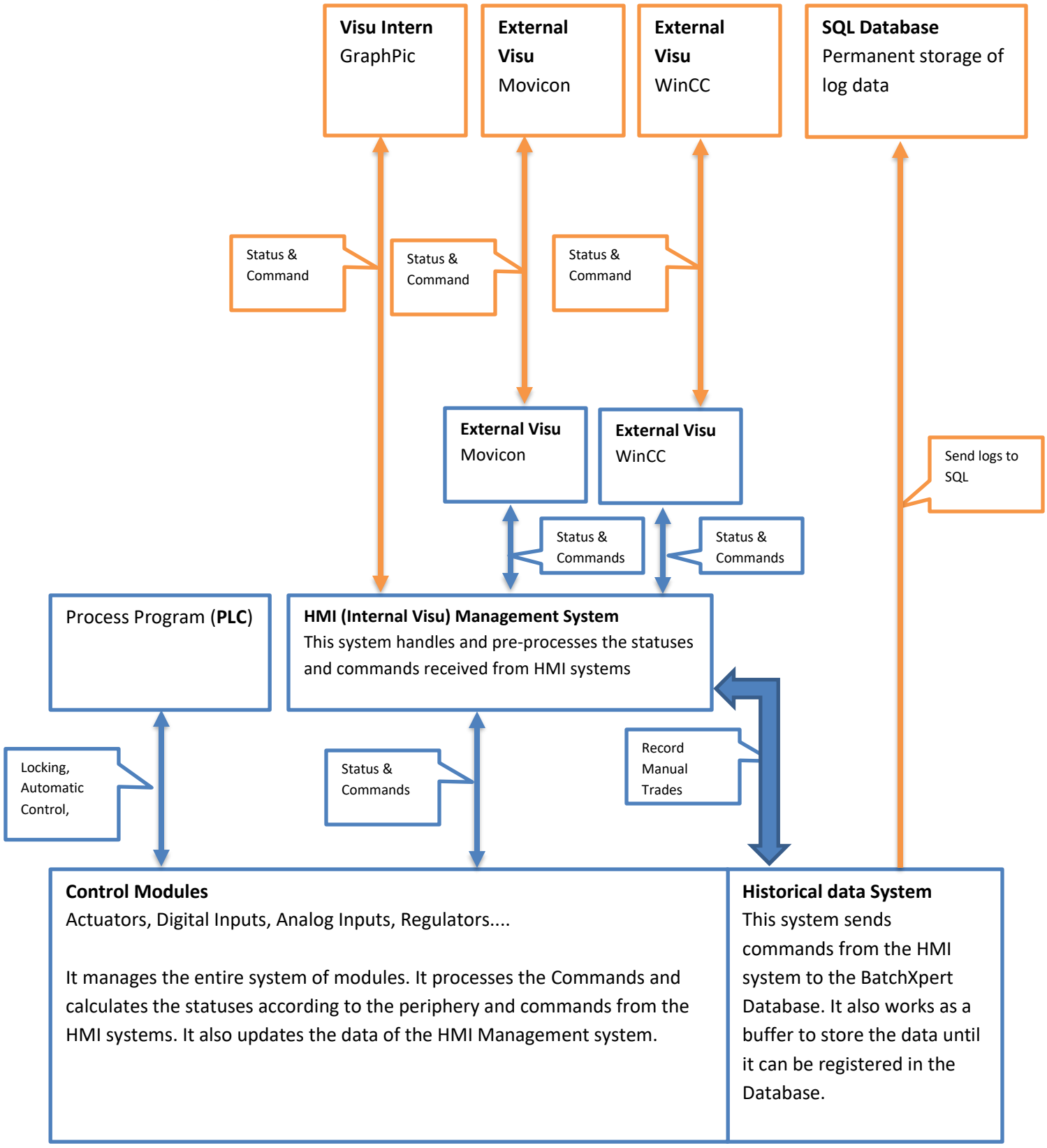
Regardless of the type of Visualization (Internal or External), the system manages and records all the actions performed to some control module, in such a way that there are records in the system's databases. This means that there are also Records on Manual Operations not only of the Internal View, but also of all the External Views added to the system.

Fundamentally, these “External HMI” blocks depend on your symbol library for the Hmi systems that you use. BatchXpert by default comes with libraries for “WinCC Flexible,” “WinCC Comfort” and Movicon. Since WinCC and Movicon have different requirements for the data exchange, they both come with their own “External Visu” block, that can be used.

Disadvantages of using HMI Panels

If you are using external HDMI systems, you must manage each HMI system separately, which generally means that you must implement your HMI screens in all your HMI systems. This is the reason why this concept is generally discouraged in favor of VisXpert SCADA, since this allows you to manage one single SCADA project for all your touch panel applications and even operating stations.

Functional Diagram



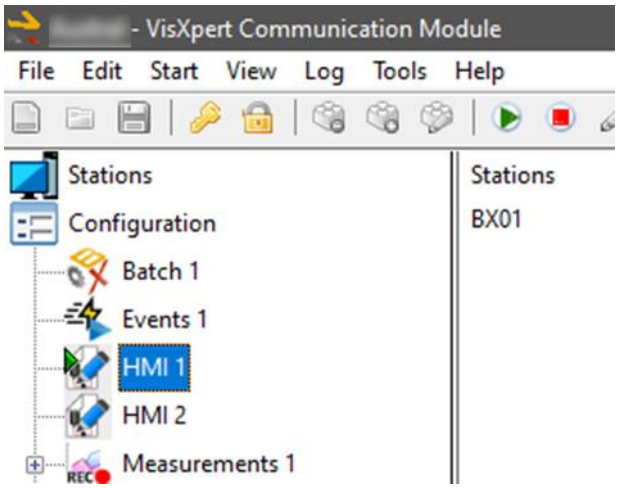
Process Graphics with VisXpert

VisXpert is the basis for all the HMI graphics of BatchXpert. You can find manuals in the MLogics Knowledgebase with more details. In this manual we only want to touch the BatchXpert specific parts of creating a process screen for your application and tries to give you an overview of the general functionality and how you can achieve common use cases with this expert graphics editor.

Here are some resources you should read first, or be already familiar with:

- Mlogics Knowledge base: https://docu.mlogics-automation.com/
- VisXpert Graphics Editor: https://docu.mlogics-automation.com/overview-1/visualization/visualization-editor/create-a-process-image/

Communication Module



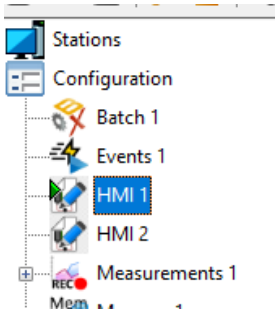
This expert is a modular SCADA system that implements all its functionality through the implementation of different modules. Each module serves a different purpose, but all modules are managed by previous expert communication module.

The communication module is also responsible for managing all the different variables, also called Tags, of your BatchXpert application. This means that the communication module is the application that manages all the data of internal variables, and the data from external OPC-servers or PLCs.

Each of the modules has a runtime and a design time editor, which allows you to adjust your configuration of your module and then execute your configuration during runtime. An example of such a module is the HMI module or the trend recording module.

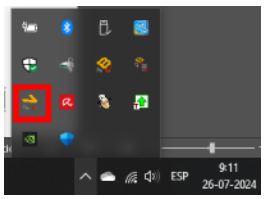
VisXpert Also includes a rich programming interface, in which you can implement custom modules in C#, C++ or Delphi. You can find more information in our knowledge base, or in the installation directory of this expert, where you can find programming examples for the ".net" platform.

HMI Graphics Editor



The Graphics editor is available in the VisXpert Communication module in the "Configuration section". It can only be activated if you have sufficient user access rights.

To open the configuration section of VisXpert, you may have to bring the "communication module" window into the foreground. Usually, this window will be minimized when starting the HDMI application and is not visible to the user. When the communication module is minimized, it shows an icon in the windows "SysTray", which is the icon area left to where the windows clock is located. If you open this "SysTray ", you can bring the communication model to the foreground by clicking on the VisXpert icon.

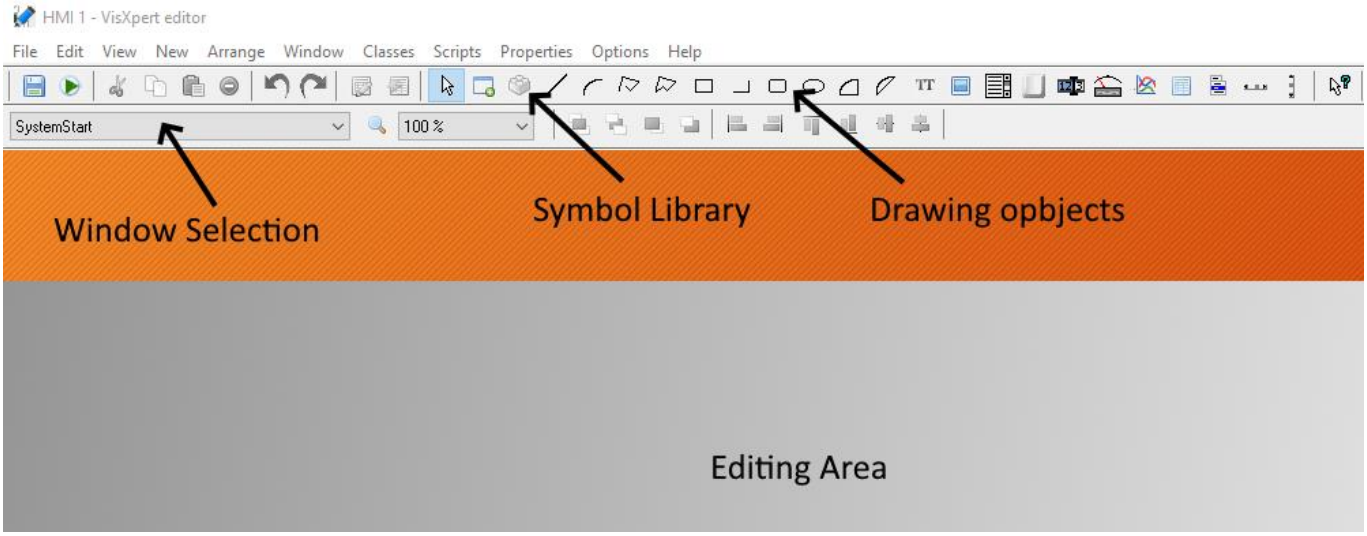


When opening the configuration three in this expert communication module, you might encounter multiple HMI applications. The reason for this is that BatchXpert supports multiple monitors, where each monitor corresponds to one HMI application. Usually you want to use the “multimon” this expert module to manage HMI applications. This module will synchronize the different HDMI application between each of the modules come up which means you should only ever edit and modify an HMI application called “HMI 1”. All other HDMI applications will be synchronized with this HMI application.

If you have sufficient user rights, you can double click on the HMI application to start the graphics editor.

The Graphics editor

The Graphics editor features a drop-down selection where you select the window that you are currently editing.



You select the window that you want to edit from the window selection dropdown menu and can then select, edit or add new drawing objects or symbols to the process screens.

All graphical screens and face plates are implemented in this graphics editor, which also means that they can be edited to fit your needs. However, some advanced dialogues, that heavily depend on data from BatchXpert database, are implemented outside of the graphics editor as in separate application to provide a better user experience.

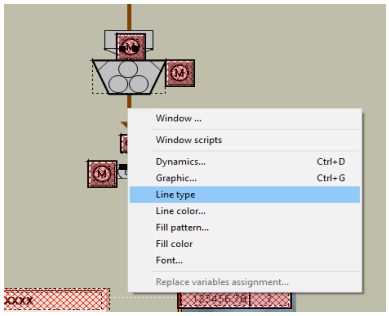
Images and Bitmaps

Images and bitmaps are an essential part of all Process images, and the reason the HMI can create modern looking process screens. BatchXpert includes a library for process screens as part of the “BatchXpert SDK” in the directory: “C:\Program Files (x86)\BatchXpert SDK\Visu\Images”

VisXpert supports a variety of image types, which includes “PNG” including transparency. **Since PNG files are the only image type that supports true transparency, we recommend that you use these image files format**. 32-bit BMP files are also supported (8 bit per color channel and 8 bits for Alpha), which also support transparency, but not all image processing applications support these image formats, which makes it harder to edit.

The image library still includes a lot of images like bmp files with static-colored backgrounds, for which we recommend converting them to PNG files with true transparency before using them.

Graphics and Dynamics properties



The graphics editor allows you to modify basic object parameters by right clicking on the respective item. The context menu gives you access to the most basic appearance properties such as line type, line color or fill colors.

This graphic properties are static properties and only define the basic appearance of all the drawing objects. To animate these properties, you must use “Dynamics”.

The ”graphic” menu option on the context menu gives you access to element specific graphical options such as text alignment tool tips value formats and such. Most notably you can adjust many text properties like text layout text rotation and such for text objects and change the graphics file used for image object.

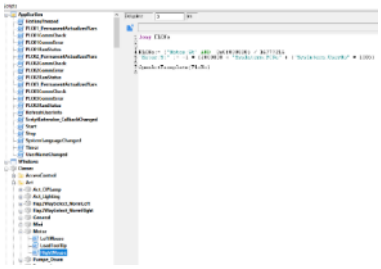
Dynamics

Dynamics is the term used in this expert for the concept of linking variables to graphic objects to animate them, change colors, change text or change their appearance.

The “dynamics” Menu option on the context menu allows you to adjust dynamic animations for a different graphical property of these drawing objects. This allows you for example to change the background color depending on the value of a variable, animate the text property of a graphics object to reflect the current value of and variable, or even execute scripts and other actions when being clicked.

You can only change graphical properties for simple objects, not for symbols from the library. The graphical properties of symbols, called “Objects”, must be changed in the “Object” editor that is also available inside the graphics editor.

Scripts



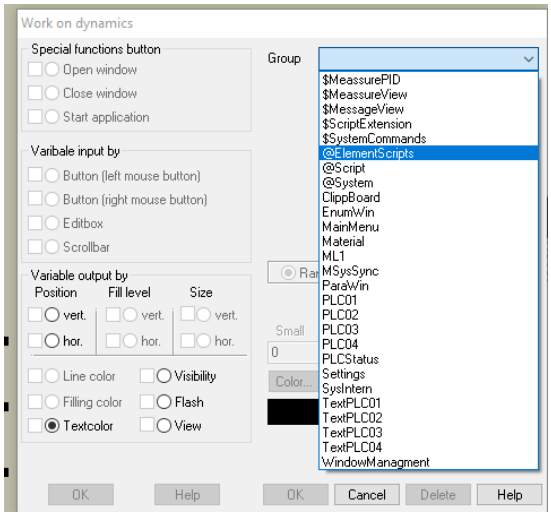
VisXpert supports a powerful scripting language. This scripting language is extensively used in the HMI symbol library of BatchXpert.

You can find mor information here: https://docu.mlogics-automation.com/overview-1/visualization/visualization-script/

The Scripting language uses an “pascal” like syntax an includes the following features:

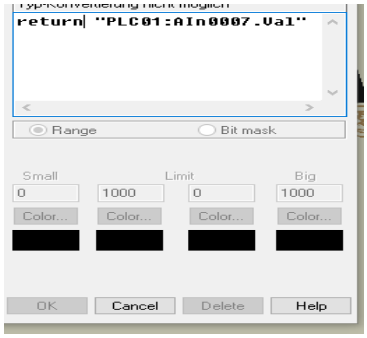
- High level, imperative language
- Database access using queries for reading and modifying data
- Manipulate Graphics properties of Drawing objects
- Access to variables of the PLC
- File access
- String manipulation, trigonometry, and other functionality

Element Scripts



The graphics editor allows you to create powerful customized animations based on the scripting language. With this scripting language you can implement most of your animation requirements that cannot be met by the normal dynamic's functionality.

To use element scripts, you must open the dynamics dialog select the property that you want to animate (in the example provided it is the text color), and then select "@ElementScripts" from the groups drop down menu.

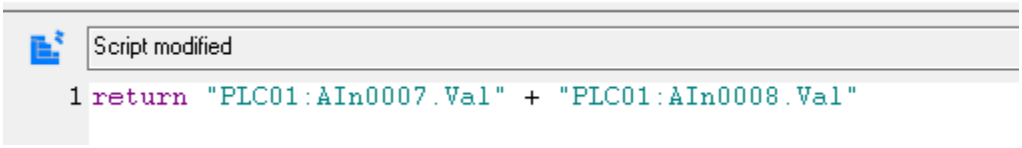


This allows you to write a customized script which calculates the value for the property that you want to animate come up based on the result of your script. This script can access any variables, functions, databases, or any file you need.

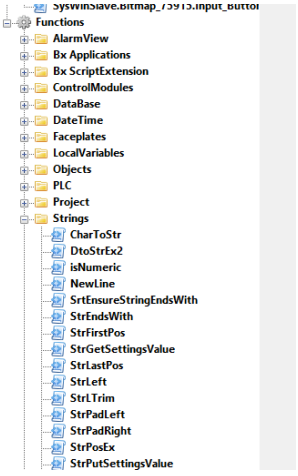
If you click on the "Switch to Script dialog" button, the full scripting interface will open, and you can write your script in the usual scripting dialog.

Usually, you use this functionality to animate drawing object properties based on sometimes complex calculations, for example: showing different colors depending on complex ranges of values of multiple analog inputs. Or showing customized texts for different values or value ranges.

Example you can see that do you result is being calculated by taking two multiple analog input values which are both different variables, doing a calculation on them and returning the result to be shown on the HMI.



Script Functions

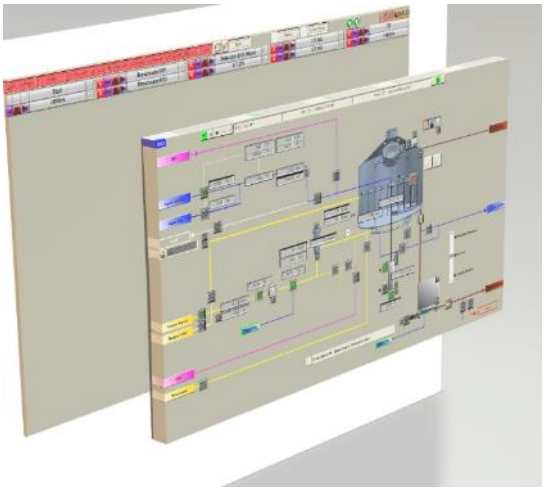


the graphics editor allows you to implement commonly used scripts into script functions. These Script functions can then be used in all other scripts, such as element scripts, object scripts and even in other script functions. By default, the BatchXpert application comes with a vast library of script functions that cover multiple use cases and extend the functionality of the scripting language itself.

It implements an advanced string processing library that enhances the string processing capabilities of your HMI applications period. You can also implement your own functions to realize project specific functionality and centralized commonly used code in one single script instead of copying and pasting scripting code into multiple objects. All provided script functions include and header with a detailed description about what descript function does and how to use it.

We strongly recommend you implement common scripts into script functions, so that you can manage them in one single place but still have them available in all other scripts. We also recommend you include a script header with a detailed description about your script function for documentation.

BatchXpert Process Screens

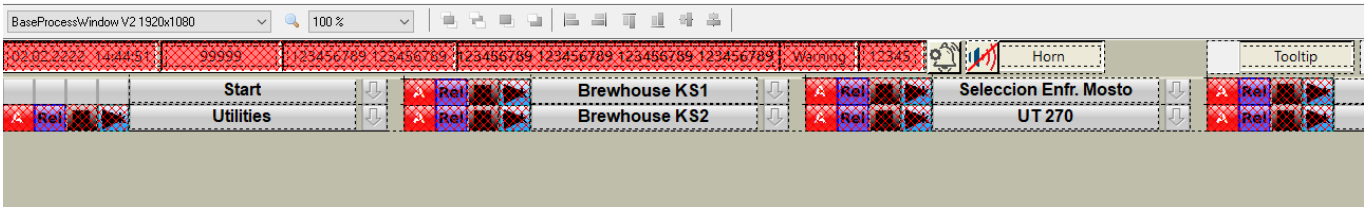


When the BatchXpert HMI is started, first it will open the Window “StartPage”, which is the loading screen of BatchXpert. After loading finished, it will open the “BaseProcessWindow V2 1920x1080” window.

This window forms the basis on top of which all the process windows will be opened. This “BaseProcessWindow V2 1920x1080” always stays open in the background, so the menu bar and all system components will always be visible.

The Base Process Window will hold all controls and drawing objects that are always visible, such as the alarm bar, the Main menu bar, and other controls to control “Expert Mode”, “tooltips” and the current Time. This base window should never be closed during the runtime of your HMI application, as it also runs some scripts in the background that provide additional functionality for the whole application

The Menu bar is also an essential part of this base window which is implemented by an “VisXpert object” that must be configured from scripts. More details later. The Main menu provides access to all your process areas by dispatching to the corresponding “overview screens” and allowing direct access to all individual windows through the drop-down menu for each area. More information can be found in the “The HMI Main Menu” chapter.

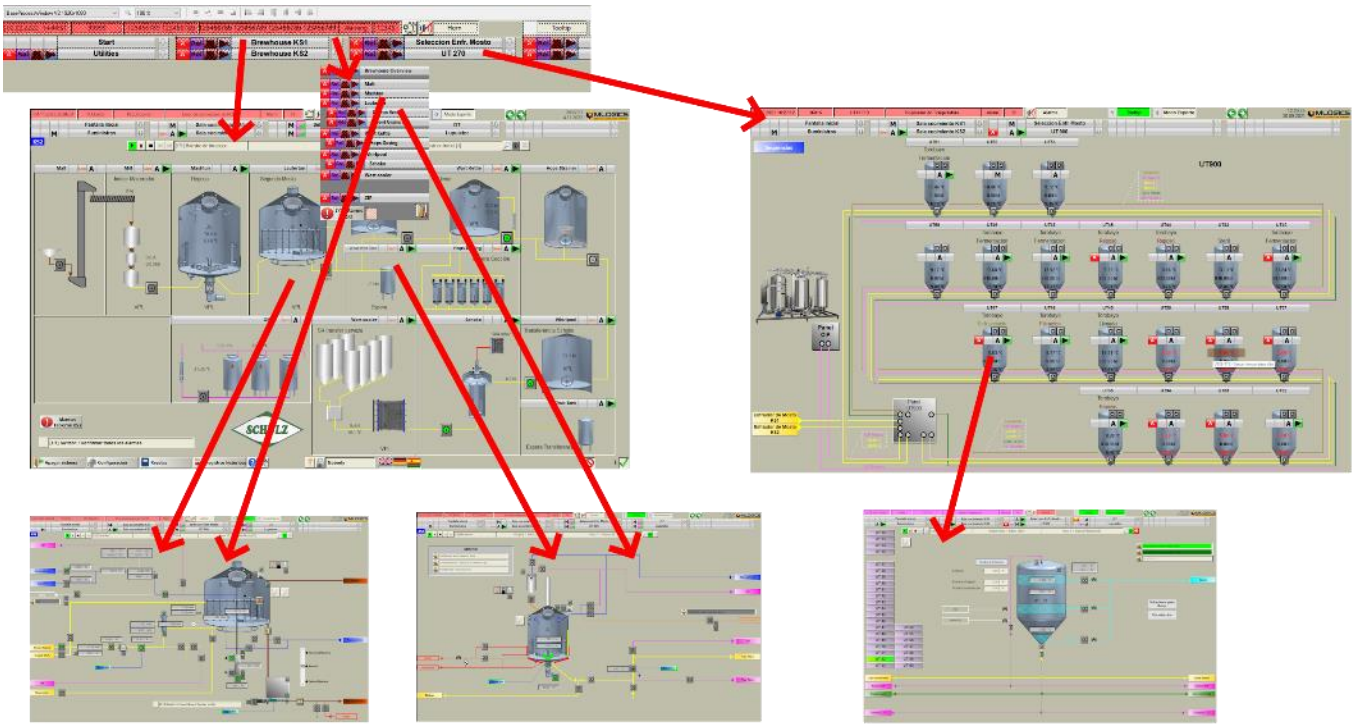


Process Screen Hierarchy

The Main Menu visible on the “BaseProcessWindow V2 1920x1080” Window, is the essential menu that allows the user to dispatch to all process screens. The “Main Button” dispatches to the Process overview, and the drop-down menu gives access to all individual process images of each area.

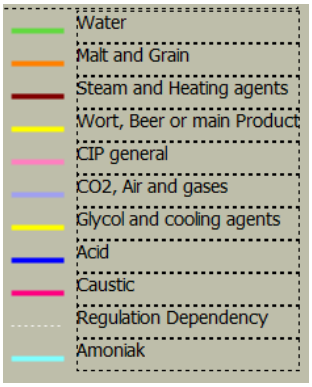
Each menu area is reserved for a single “Processing Area” such as “Utilities”, “Brewhouse”, “Fermenting”, “Maturation”, etc. The first menu item is reserved for the “Start” menu, which provides access to all BatchXpert applications and serves as main menu for BatchXpert itself.

All process areas are organized multi-level process image hierarchy. On the top level there is his start page which shows information about the connection status and alarm status of all process areas, below that you can access an overview picture for each of the process areas that are available from the main menu, and below that are process images with details for each processing unit. The main menu allows you to access these different process window hierarchies by using drop down menus that allow you to open do you required unit detail process windows directly from the main menu.



Drawing Process Screens

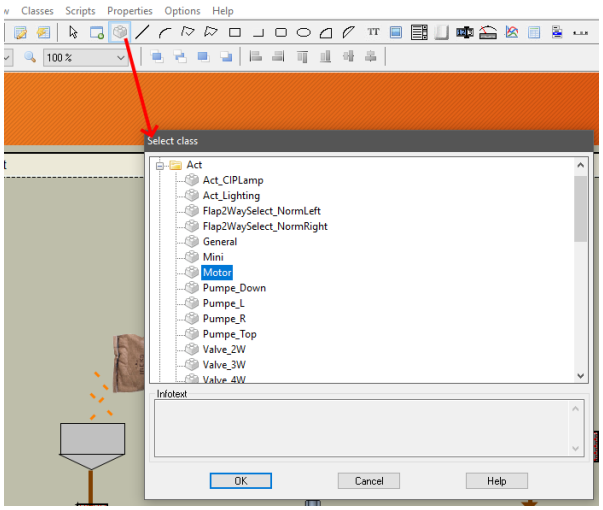
Since the process screens are opened “on top” of the “Base Process Screen” they must have a smaller size and position offset to the “base process window”. By default, the size and position of process windows should be as shown in the image.



The Process Screens can be created by applying Screen layouts, coloring etc. as the client wishes. BatchXpert provides an “Standard” for coloring of process lines, and an image library for implementation in the process screens, which are available in the “Library” process Screen in the HMI template.

All Control modules are represented by “Objects” which encapsulate all functionality of a module and can be easily configured by assigning variables to them, more on their configuration below.

The Symbol Library

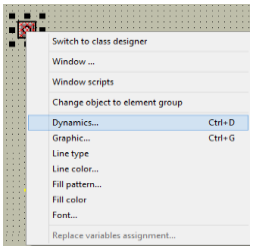


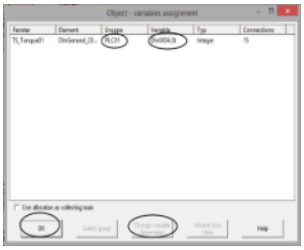
BatchXpert implements its symbol library by using “VisXpert Objects”. These objects are self-contained drawing objects that contain dynamics, scripts, and all graphics components of a control module.

They are also fully integrated into the BatchXpert HMI system and incorporate features like Opening Faceplates, Sending commands to the PLC, etc.

When you add a Symbol to the Process graphics, you must specify its data connection. For example, for actuators, you would have to specify on which Actuator Variable will operate.

Configuring BatchXpert Symbols





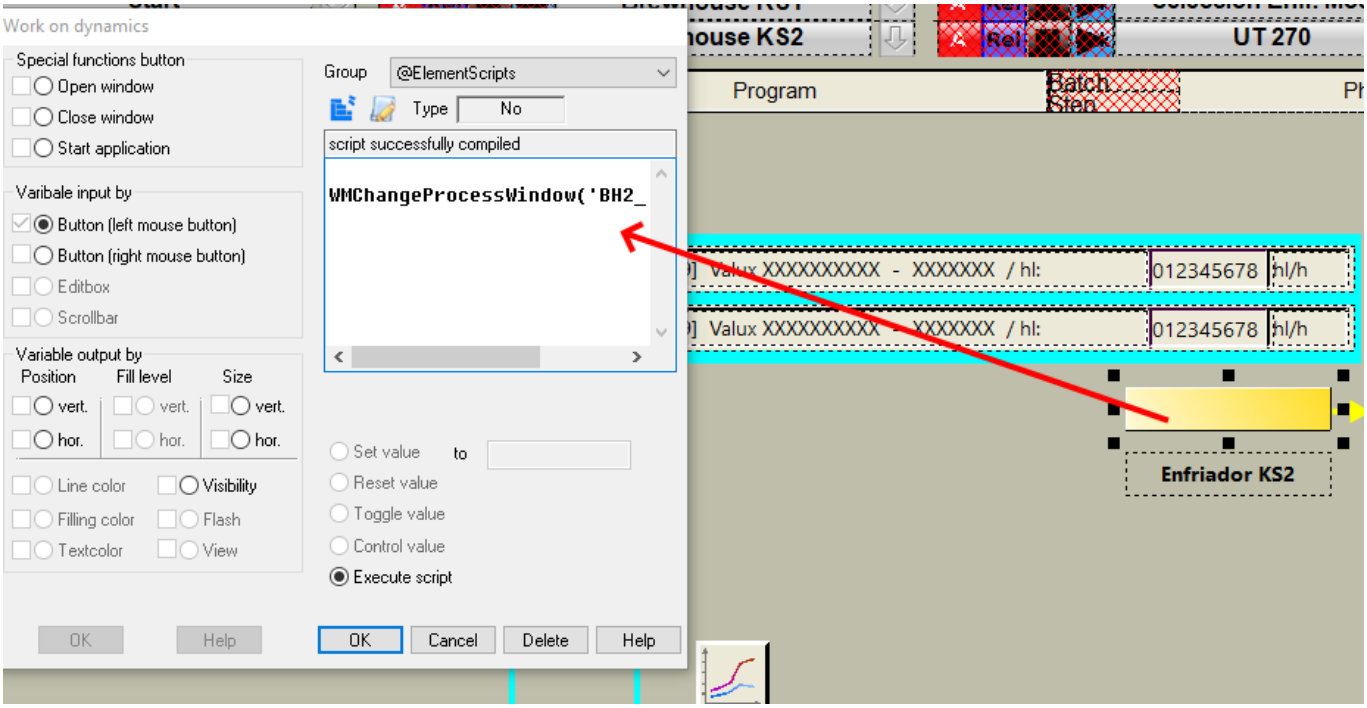
In the project images the objects have a PLC variable assignment, right-clicking on the object will open a menu where we will select "Dynamics" where you must select the group and the variable, example, group: PLC01, Variable: DIn0004.
To edit the existing variable, click on the variable and click on "change variable base name".
Once all the image configurations are finished, save, and start the simulation to verify that everything is correct.

Change Process Windows

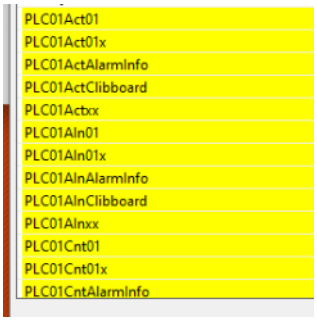
To create buttons that switch the visible process window, you should create a script action on a button or image and call the “WMChangeProcessWindow(string windowName)” function from the provided scripting library.

You should avoid using the open and close window functionality of buttons or other objects, because the window manager script functions provide additional functionality for the HMI system, such as a previous image buffer, and special handling for face plates and modal dialogues. BatchXpert provides multiple window manager script functions that you can use for different purposes throughout your project.

The buttons on your process screens that allow the user to change process screens are usually composed of and a bitmap of one of the prepared button colors with the appropriate text overlaid on top of this bitmap. The change process windows script is then programmed in the button left mouse click event of the bitmap, as seen in the image below.



Remove unused “Windows”



If you copy a project from a different project, you should remove all screens that are not actively in use. This ensures that you do not have references to variables that you are not using anymore and reduce the size of the HMI Project file.
Especially you should remove all “PLC0x” windows, where x corresponds to an PLC number that you are not using. All these windows can easily be imported again from the HMI library as described in chapter Importing the PLC dependent Windows from your library

Process Graphics Style Guide

The following chapter we are going to present our recommendations for styling process graphics in the BatchXpert System. Of course, you can adapt this guides to fit your company and client preferences, the following are only guides after all. The following rules complement the existing guidelines and are intended to keep all process screens consistent across projects, stations, and engineering teams. Apply them whenever you create new screens or modify existing ones.

- **Use a consistent grid.** Align all symbols, labels, and value fields to an underlying grid (recommended: 8 px or 10 px). Avoid "almost aligned" placements; if two objects belong together, they must share the same left edge, centerline, or baseline.
- **Standard margins.** Keep a minimum margin of 24 px from the screen edge and 16 px between functional groups. Reserve the top area for navigation and the bottom area for alarms/status (if your project uses a fixed header/footer).
- **Typography hierarchy.** Use one font family across the project. Keep three text sizes only: *Title* (screen name), *Section labels* (group headers), and *Body/value text*. Do not mix bold/italic for emphasis unless the guideline explicitly calls for it.
- **Readable tag/value labels.** Labels must be left-aligned, values right-aligned (or aligned by decimal separator if supported). Include engineering units (e.g., °C, bar, m^3/h) directly next to the value.
- **Line weights and connectors.** Use a limited set of line weights (e.g., thin for pipes, medium for equipment outlines, thick only for main headers). Keep connector directions orthogonal (horizontal/vertical) and avoid diagonal lines except where the symbol standard requires it.
- **Color meaning is reserved.** Do not use alarm colors (red/yellow) for decoration or grouping. Use neutral colors for backgrounds and containers; only dynamic objects may change color to indicate state.
- **Status indication rules.** Prefer a single, consistent method per object type (e.g., fill color or indicator lamp). Do not combine multiple competing indicators (e.g., blinking + color change + icon) unless required for safety-critical conditions.
- **Animation restraint.** Do not use blinking or continuous animations. These should be avoided because they increase operator fatigue and reduce alarm salience.
- **Clickable area and affordance.** All interactive objects must have a sufficiently large click/touch area (minimum 32×32 px; 44×44 px for touch panels).
- **Faceplate consistency.** Every instance of the same equipment type must open the same faceplate and expose the same primary actions and status fields. If a special instance needs extra fields, add them as an optional section—do not fork a new faceplate unless absolutely necessary.
- **Navigation and drill-down.** Provide a predictable navigation path: Overview → Area → Unit → Equipment detail. Use breadcrumb or consistent back buttons. Avoid "dead-end" screens with no route back to overview.
- **Naming conventions.** Use a consistent tag naming style for labels (e.g., "P-101 Feed Pump" rather than "Pump1"). Screen titles should match P&ID or engineering documentation identifiers where available.

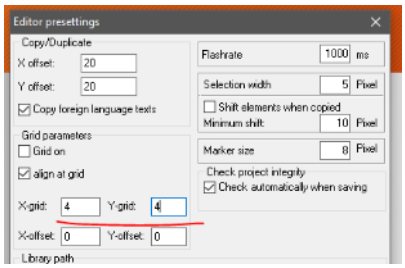
- **Accessibility checks.** Ensure sufficient contrast between text and background. Do not rely on color alone to convey state; add shape/icon/text cues for critical statuses when possible.

High Performance HMI Handbook

We recommend that you read the “High Performance HMI Handbook” by “Bill Hollifield”, which contains a lot of great advice and guidance for creating HMI interfaces for all sort of Processes, vehicles and Machines. BatchXpert adopts some aspects of these guidelines but does interpret others more loosely. The Color scheme used by BatchXpert uses many more colors than is recommended. However, you should adhere to the following general rules:

- Do not use “Flashing” or “Blinking” animations, except for exceptionally critical indications such as things that may damage personal or equipment. Normal Process alarms do not fall into this category and should not be made “Blinking”. In practice, BatchXpert never uses blinking animations in any of its defaults Objects and screens.
- Red is reserved for “Alarms”, and should never be used in any process screen, except for Alarms
- Flow diagrams should reflect the logical flow, and not be an exact representation of the P&ID. You should omit non necessary symbols such as, manual valves, check valves, filters etc., wherever not relevant.
- Avoid Moving Animations. You should not use animations that “move” objects, such as “Rotating Agitators” or such.

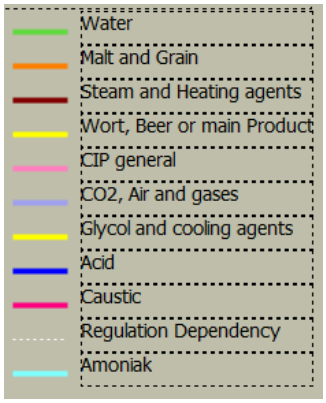
Preferred Editor Settings



All BatchXpert Objects are designed to be 36x36 pixels in height and width, which makes the “Centerlines” align exactly to grid points of 4 x 4. This way you can position all your objects precisely in the middle of lines.

In the Graphics editor you can open “Options-> Editor” you can set these grid size and activate “Align at Grid”, so that objects automatically align to the nearest multiple of your Grid settings. We recommend the settings on the left.

Pipes



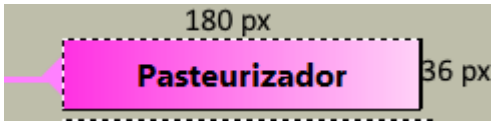
The following process colors are recommendations, but not mandatory to be used for your process graphics. Of course, this is only a guideline, and you can adapt any coloring scheme you wish for in your projects.

You should differentiate between Primary product flow and secondary flow by using different line thicknesses. Usually, the primary product flow has a line thickness of 3, and all secondary product flow have a line thickness of 1 pixel. Secondary product lines should never interrupt primary product lines, neither horizontally nor vertically. Primary Product lines should clearly denote the main product flow.

You should connect pipes in 90° angles with each other where possible and avoid using arcs or bends to “simulate Pipe bends”. Use simple 90° angels to create easy to follow diagrams, which are similar but simpler than P&ID drawings.

Process interconnections (Window Jumps)

Connections to different process screens, meaning pipes that come and go from/to different parts of the process that are represented on other Process screens, are represented by an “Page Jump”, that is represented by an “Color graded Rectangle” in the same color as the pipe. Usually, these page jump indicators should have an accompanying “Arrowhead” to indicate flow direction.



These page jump indicators are implemented as Bitmaps, which are available in multiple colors to fit the piping that is indicated. The size of these page jumps should be 180 x 36 pixels. The bitmap itself should then hold the Script logic to “Change Process Windows”.

Equipment representations

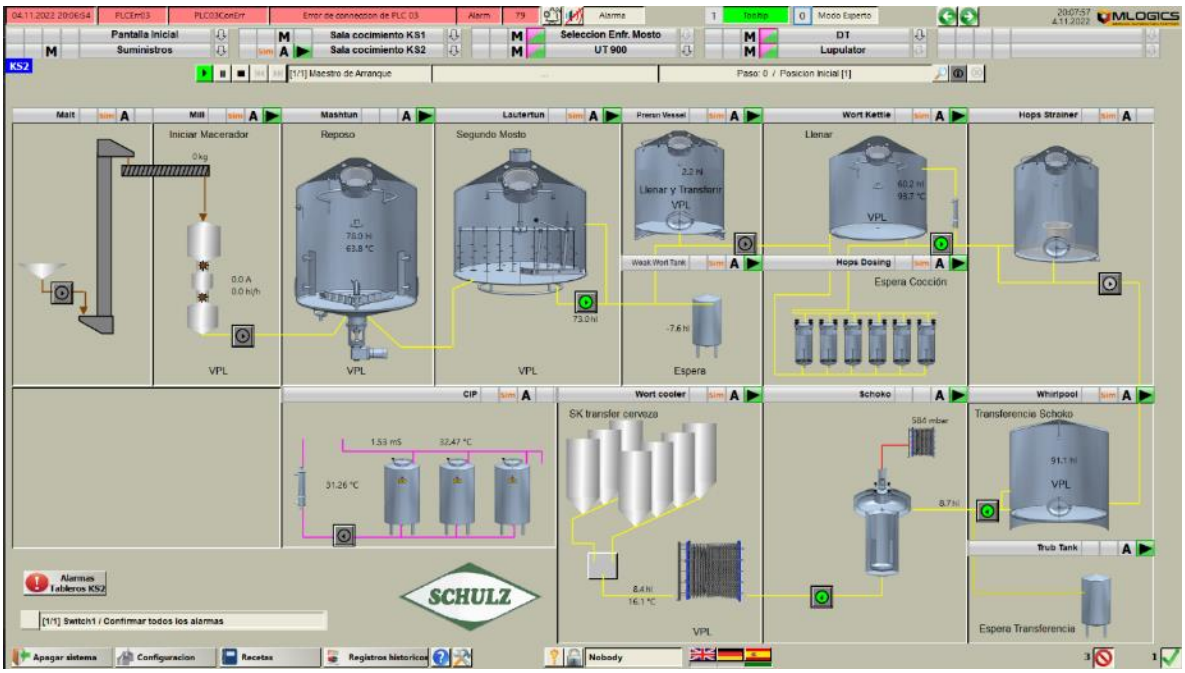
To represent different equipment types such as Lauter tun, Mash tun or Heat exchangers, you should use representative bitmaps or PNG images. These images Should ideally be in muted pastel colors or ideally in grayscale colors, as to not distract the user’s attention away from important information.

Overview Screens

Each processing area should have one overview image, which represents the overall process layout of this area, with as little details as possible, while maintaining the most important control modules. This is usually done by representing a small pictogram off the production vessel and only showing vague schematics for production lines and only representing the most important control modules, such as the main product pump or main product valves, and omitting all CIP, water and other supply lines.

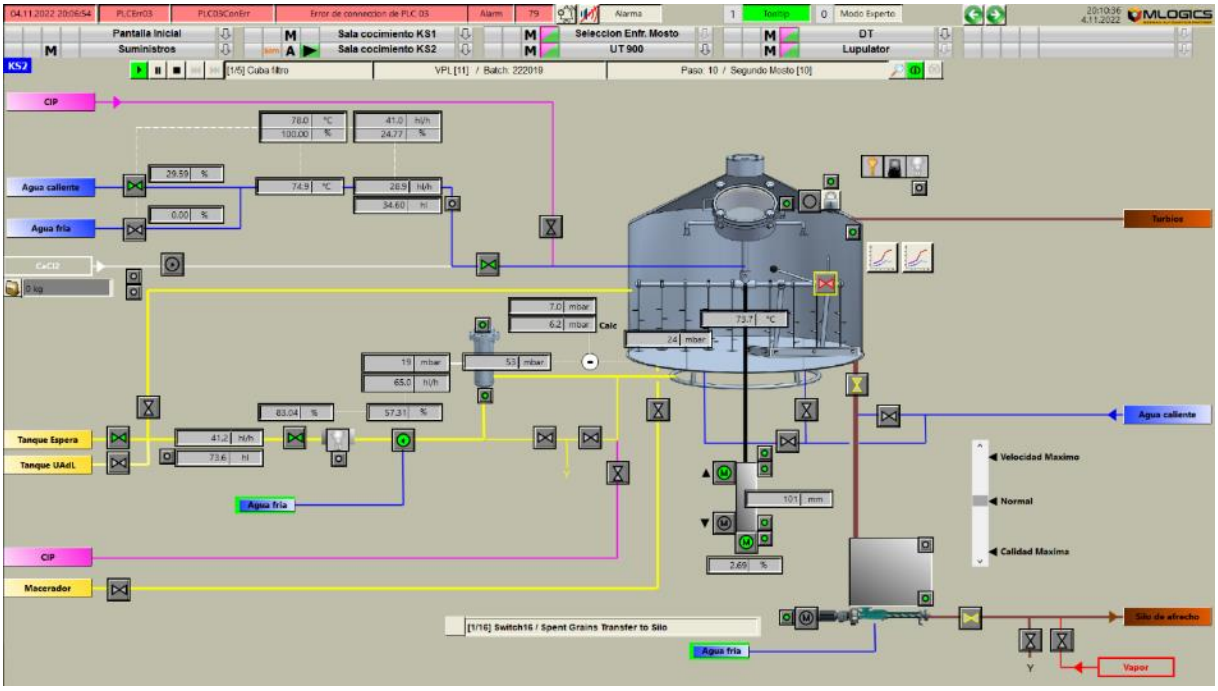
The idea of such an overview image is to give a quick overview and easy to read information to the user. This way the user can easily and rapidly understand the overall status of the whole processing area. All relevant processing steps should also visualize the current operating procedure and recipe, as the rapidly allow user to determine what each equipment is doing in the process.

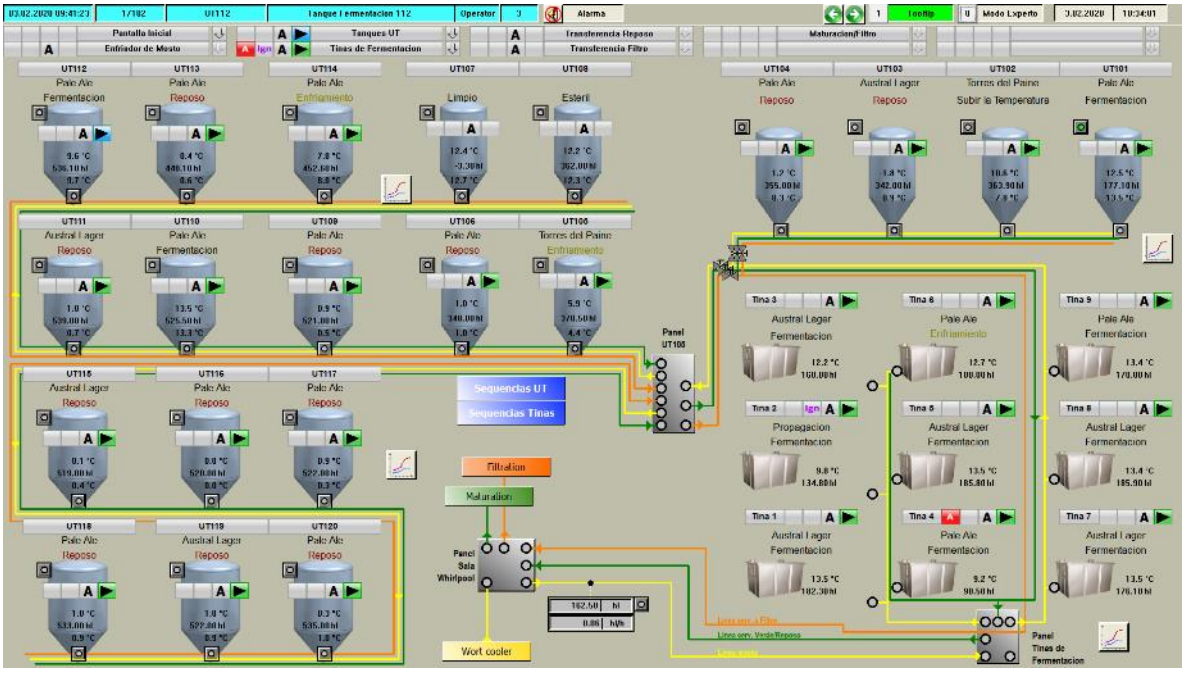
The following image is an example of an overview of a brewhouse:



Example screens

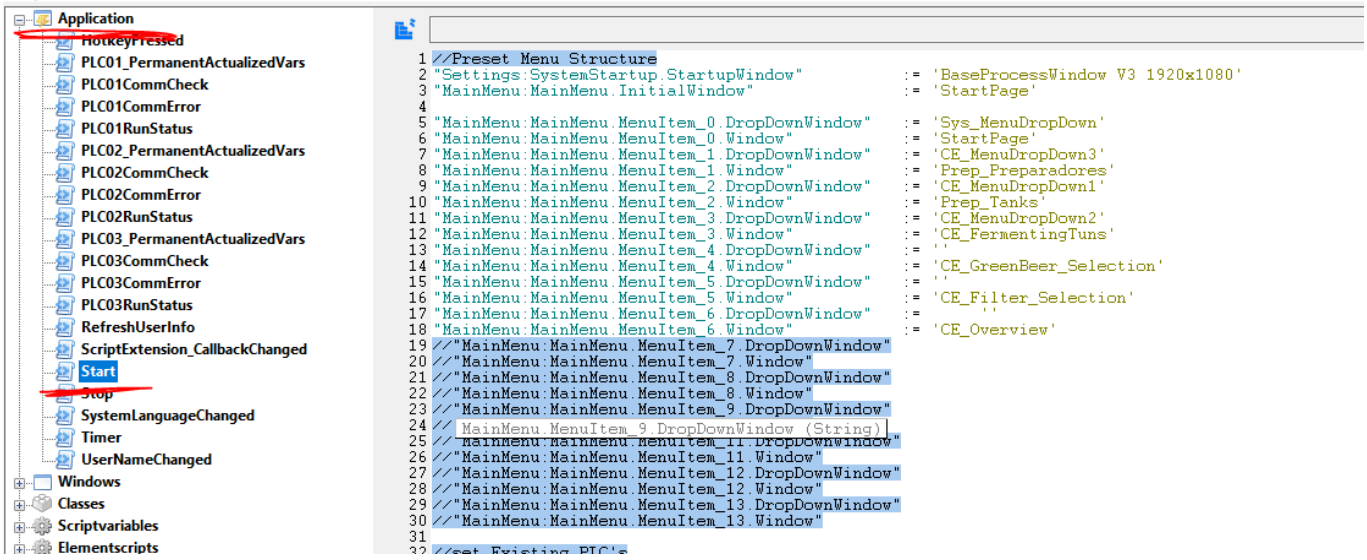
The following screens are example screens that represent typical processing graphics of BatchXpert. Of course, you can deviate from this style guide to fit your company's and the client's requirements.





Application Start Script

The HMI Application of your Project includes an “Application. Start” script, which runs as soon as the HMI runtime is started up. This script is used to configure and set up multiple global variables such as the Main menu and optionally synchronize the Operating stations time with the PLC time.



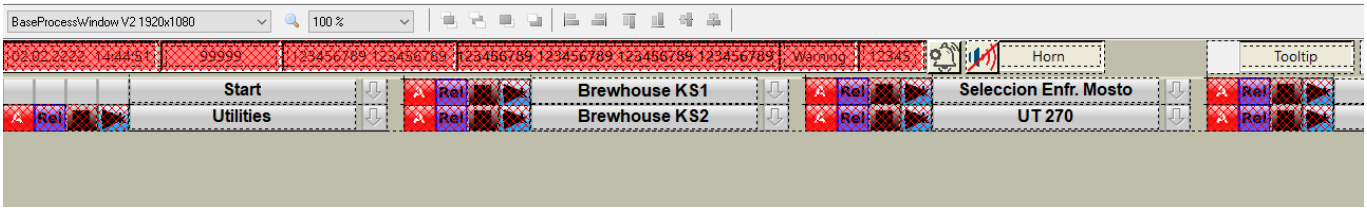
The Application Start script will perform the following actions:

- Configure the Main Menu
- Optionally set the PLC time
- Set the Application Startup language for the HMI according to the project settings
- Optionally block access to the operating system
- Reset PLC system alarms, which may have been “stuck”

You can easily add your custom startup logic here. You can find a lot of comments in the Application start.

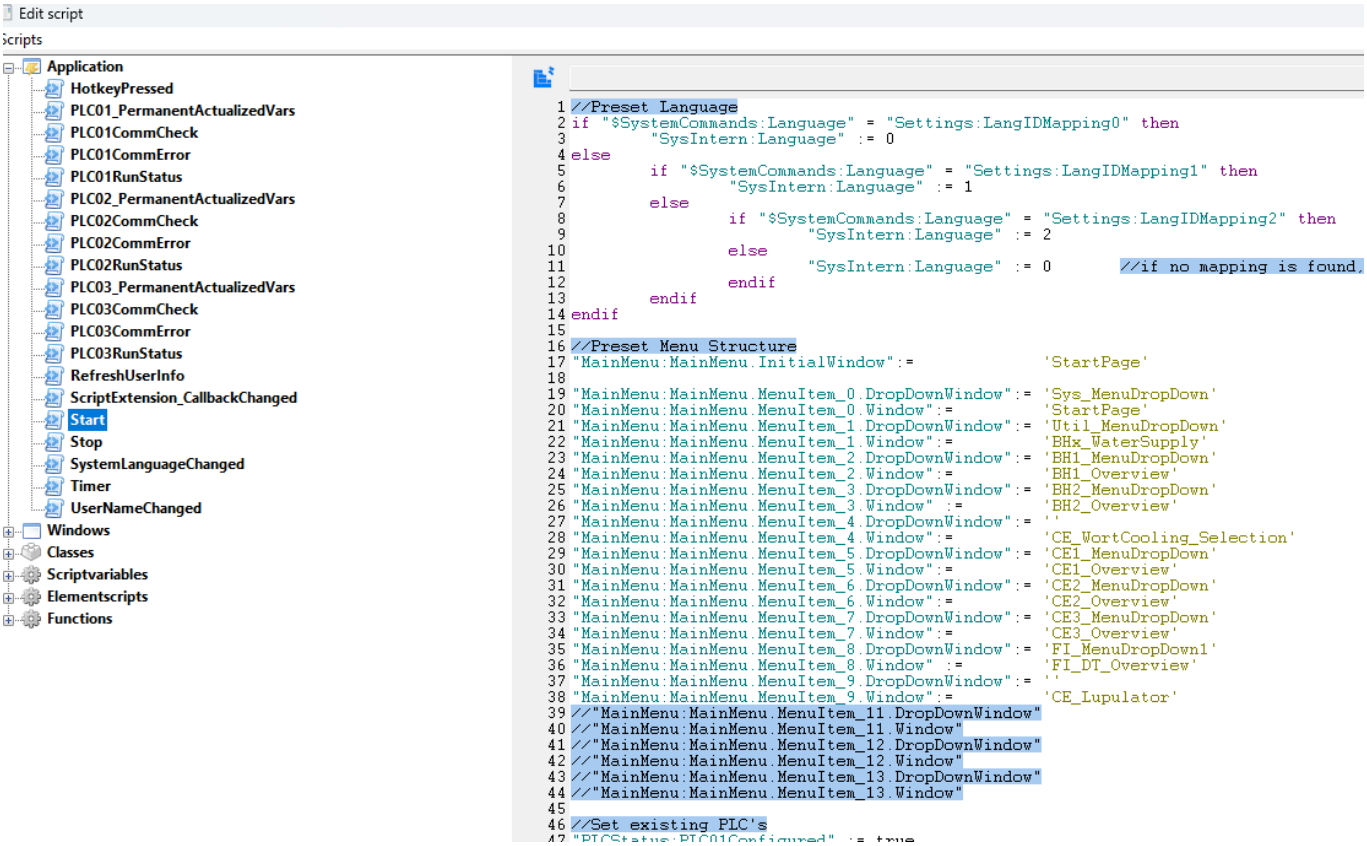
The HMI Main Menu

The HMI Main Menu is located on the “BaseProcessWindow V2 1920x1080” or “BaseProcessWindow V3 1920x1080”. This menu allows the user to quickly and efficiently move to different HMI screens of the plant. The main menu is implemented as a library object, and it's located on the base window of all process screens. The main menu has an area where you put an alarm group, a main button and an optional dropdown button. Usually there is only one single menu per plant, which means that the menu stays the same for all process screens.



Configuration of these Buttons

The menu is a library object, and all its buttons’ functionalities are configured through the variables assigned to the library object. The Variables are usually contained in Group “MainMenu”. You can define an “Main Window” and an “Drop Down” window, which opens when clicking on the small “arrow” on the right side.
The configurations of the windows that should be linked to the main menu is usually done in the “Application.Start” script.



Main button Text and Translations

The text of each main button is configured directly in the HMI editor on the “BaseProcessWindow. The Main button Text is a simple text on top of the “Main Menu” library object. This means that the normal Translation mechanism of VisXpert applies to these texts as well.

Alarm Group

Each button usually also includes an Alarm Group, which shows the process of the underlying Area with colors and pictograms .These Alarm groups are also normal library objects and can be configured the same way as ordinary objects, by assigning a Variable name and Group to them.

Drop Down Windows

Drop down windows are small windows that pop up and show all underlying process pictures that can be navigated to. The drop-down window in and of itself is an ordinary window and can be configured like any other window and assigned to the main menu from the application scripts. The Drop-down button includes a special functionality to always move the Drop-down window to where the Drop-down button was clicked, to appear as a drop-down window. This also means that the configured location of the drop-down windows is not required and will always be overridden when a drop-down button is clicked. This also means that one drop down window can be used in multiple drop-down buttons.

Configure communication with the PLC

All variables that are being exchanged with a connected BatchXpert PLC, are managed by the “PLC Data” module, which is accessible from the “VisXpert communication module”. Double clicking on this module, you will give access to the “VisXpert variable editor”, which allows you to change the configuration of all configured variables in your project. It allows you to add new variables, or change the configuration of existing variables, and the communication settings for all connected PLCs.

Each PLC is represented by its own group which follows the PLCxx naming scheme. If you want to change the settings for PLC01, you must select the corresponding group from the group list and then edit its communication parameters.

For specific Settings required in a S7-1500 and S7-1200 PLC, please refer to Connecting to an S7-1500 and S7-1200 PLC.

Memory Variables

The VisXpert variable editor also manages internal memory variables that are not connected to any external data source such as OPC servers or PLCs. This means that you can also add custom memory variables if you should need them in your application. To add custom memory variables, you can add a new group and select the memory module for its data driver.

Memory variables only exists on the local station and are not synchronized between all operating stations. This means that memory variables always have and local scope and should be treated as local to the operating station. Each operating station will have individual values on these memory variables, since they are never synchronized between operating stations. Should you need custom variables that are shared between operating stations, you should share them using and data area in your PLC.

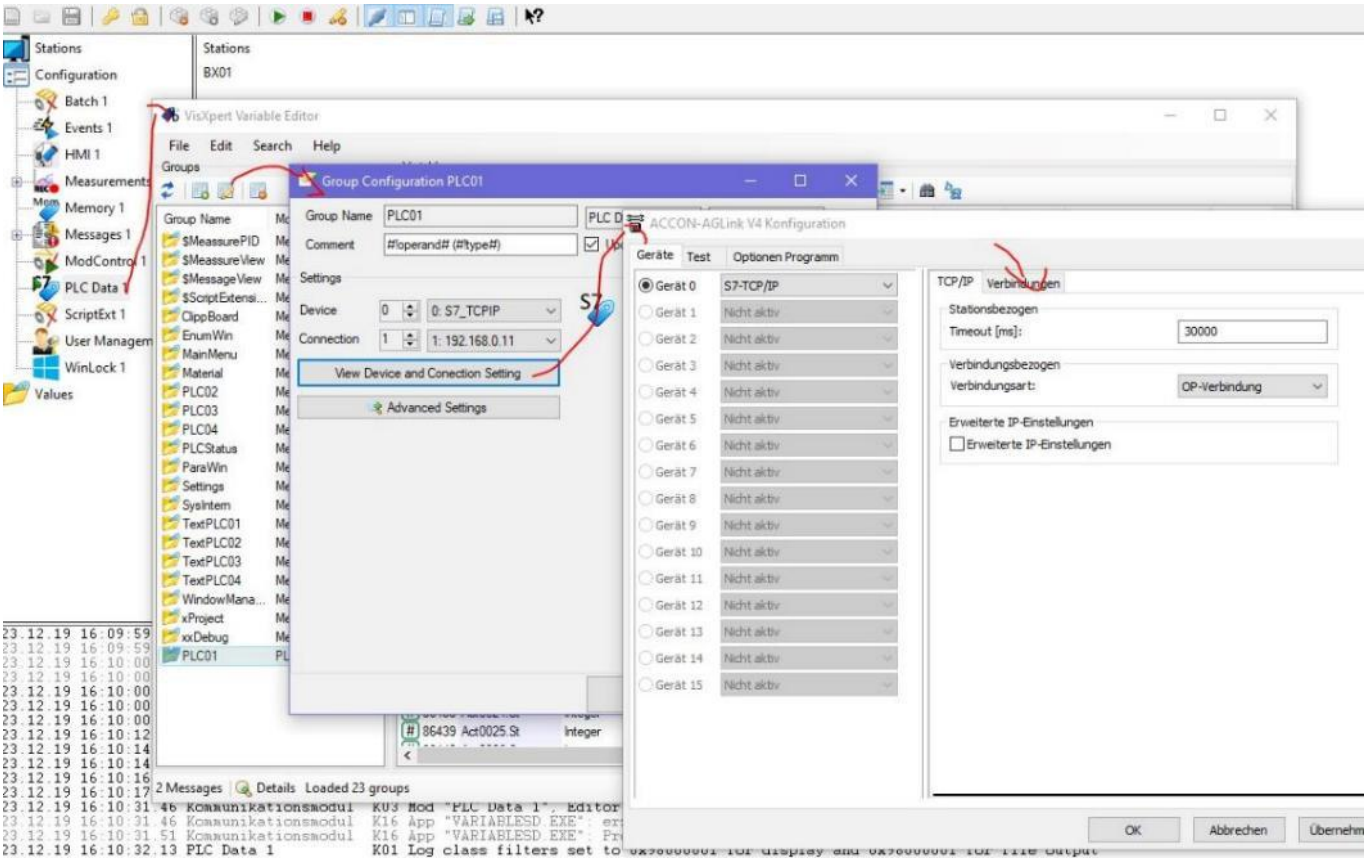
Custom Variables in the PLC

If you want to add custom variables that are communicated with any of the PLCS of your BatchXpert system, you can do so by adding your custom variables and their address configuration to one of the existing PLCxx variable groups. You must input the data address of your custom variables in the standard Simatic addressing format.

0/1	84590	WinActPC1Open	Boolean	DB 5.DBX 5.0 BOOL TB 1	BOOL	1	cyclic	readwri
0/1	84589	WinAlnPC1Open	Boolean	DB 5.DBX 5.2 BOOL TB 1	BOOL	1	cyclic	readwri
#	94539	WinAlarmAreas	Integer	DB 5.DBD 100 DINT TG 4	DINT	4	cyclic	readwri
0/1	84591	WinCntPC1Open	Boolean	DB 5.DBX 5.5 BOOL TB 1	BOOL	1	cyclic	readwri
0/1	84592	WinDInPC1Open	Boolean	DB 5.DBX 5.1 BOOL TB 1	BOOL	1	cyclic	readwri
0/1	84593	WinMessagePC1Open	Boolean	DB 5.DBX 5.4 BOOL TB 1	BOOL	1	cyclic	readwri

S7 Communications Settings

The S7-Connection require the Slot and Rack communication god parameters, which should be defined according to which PLC you are trying to communicate with. If you want to change the settings for PLC01, you must select the corresponding group from the group list and then edit its communication parameters.



AG-Number: The AG-number, or PLC number, refers to and running number of your PC connection. This should be the number of you PLC as configured in your “Batch Configuration”. for PLC01 it should be 1, for PLC02 it should be 2 and so on.

AG-Nr	IP-Adresse	Rack	Slot	Typ
1	192.168.0.11	0	2	S7-300/400

IP Address: The IP address should be the address of the specific network adapter used by your PLC to connect to the project expert station. This is usually the IP address of your internal CPU, or of your external communication processor, it’s configured in your hardware configuration of your PLC program.

Rack and Slot: Rack and slot off your PLC depends on your specific hardware configuration, but usually follows this table:

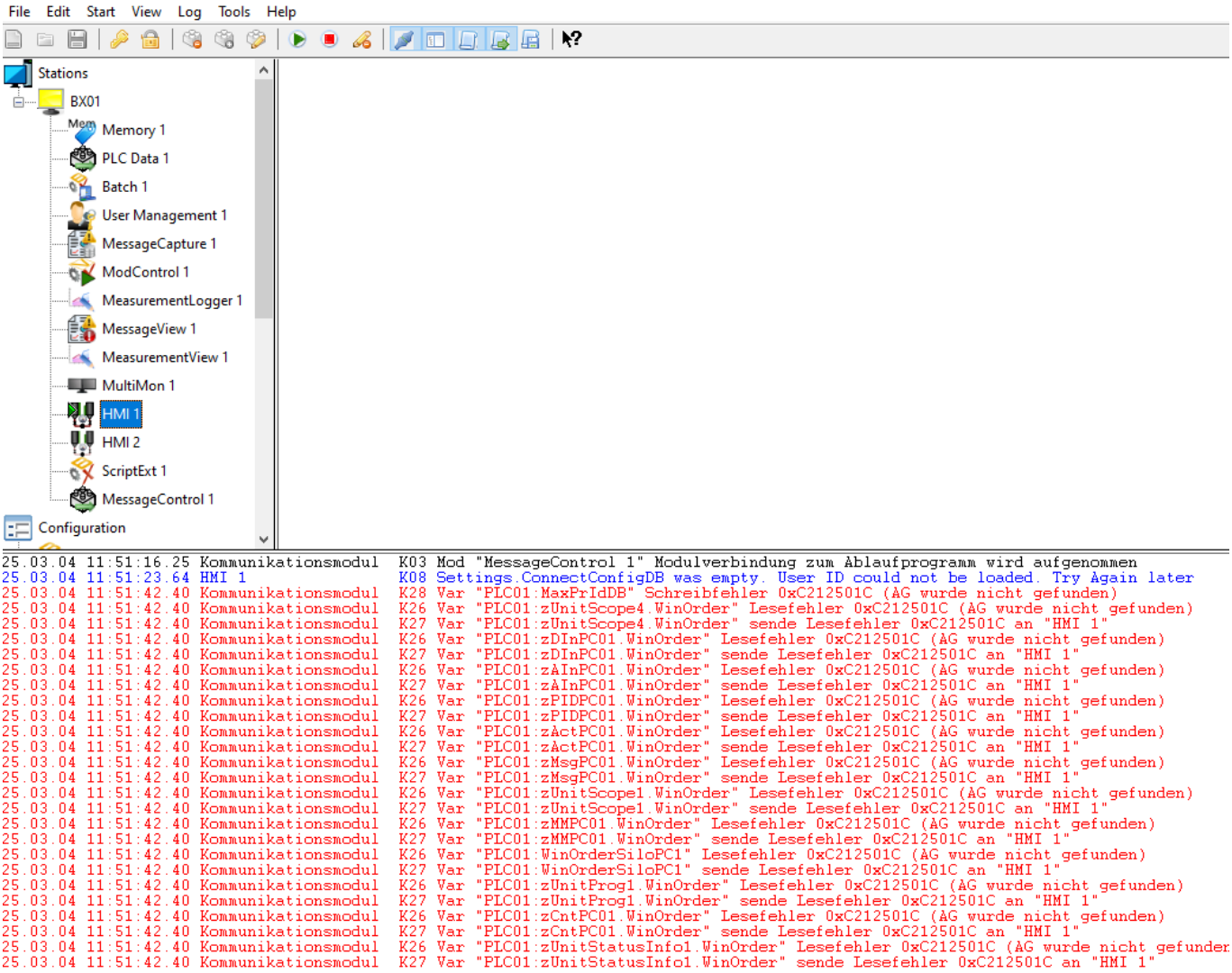
S7-300 series	Rack = 2, Slot = 0
S7-400 Series	Rack = 3 (usually), Slot = 4. But both parameters depend on your specific configuration
S7-1500 Series	Rack = 0, Slot = 0
S7-1200 Series	Rack = 0, Slot = 0

Type: The type can usually be left at 300/400 series, but it is recommended to select the correct PLC type you are connecting to.

Diagnosing PLC communication problems

BatchXpert uses the VisXpert driver infrastructure for data exchange between the HMI application and the process controllers. If a VisXpert driver encounters any problem while trying to connect exchanging data with PLC, it will write an appropriate error message into the VisXpert Logging window.

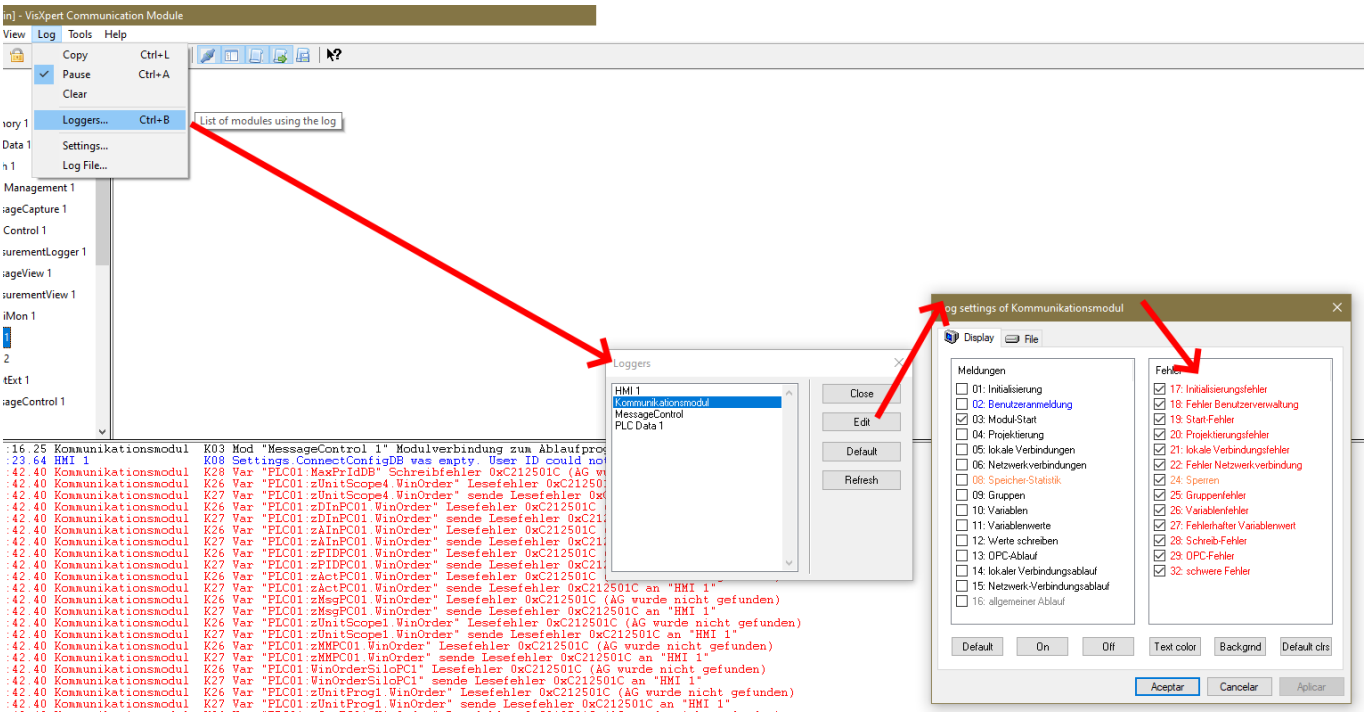
This logging window is part of the VisXpert communication module and should be your first point of reference when you encounter communication errors with the controller.



In the communication module main window, you can see all error messages in the lower part of the application.

Adjusting Log Level

VisXpert Allows you to set different logging levels to show and hide specific error messages in the log window. You can customize these log levels from the main menu of the communication module as shown in the picture below.



General Information about error messages

All error messages are dependent on the driver used, but for S7 controllers you can observe the following error messages. The error code returned from the controller can be seen between the parentheses of the error message of the variable that failed to connect.

"AG Wurde nicht gefunden" ("PLC could not be found")

This means the driver attempted to establish a link with the PLC but received no response. In practice, this usually indicates a breakdown somewhere in the communication chain — network, configuration, or the PLC itself.

1. **Something is blocking communication**

Several factors can interrupt communication between a PC and a PLC:
- A firewall on the PC or network may be filtering the required ports.
- An antivirus or security suite might be blocking the PLC communication protocol.
- A switch, router, or VLAN configuration could prevent traffic from reaching the PLC.
- A faulty cable or disconnected network segment can also cause silent failures.

This point highlights that the issue may not be with the PLC at all, but with the infrastructure around it.

2. **The IP address might be wrong**

If the PLC's IP address is incorrect — or if the PC is trying to reach the wrong address — communication will fail. Common causes include:
- The PLC was assigned a new IP address during commissioning.
- The PLC reverted to a default IP after a reset.
- The PC and PLC are on different subnets without routing.
- A typo or outdated configuration in the engineering software.

Even a small mismatch (e.g., subnet mask) can break communication entirely.

3. **The PLC might be blocking communication**

Some PLCs have built-in security features that restrict access:
- Access control lists (ACLs) that only allow certain IP ranges.
- Disabled programming ports or protocols.
- Password-protected or locked communication channels.
- A mode switch (RUN/PROG) that restricts external access.

If these protections are active, the PLC will simply ignore connection attempts.

"DB existiert nicht" ("Data block does not exist")

The data block you are trying to read or write from/to is not loaded in the PLC. Please check your PLC program and our driver variable configuration. This usually happens when your Plc has no project loaded yet, for example if its memory was reset.

1. **Check if the has a project loaded**

Especially when installing a new PLC or during testing with Simulation PLC's, it may happen that the PLC starts up with its memory reset. This means that it does not have any data blocks, or code blocks, which execute. In that case you must download a project into the controller first.

2. **Check if the PLC memory has been reset**

This may happen if the PLC has lost power to its power supply for some reason. If you are using the recommended Battery backed power supply for the S7-1500 series Plcs, the CPU resets its memory, if the CPU gets disconnected from its power supply.

Another reason, although very rarely, is that you might encounter error in the PLC during power cuts or unstable voltages.

3. **Check your Project and Variable configuration**

It may also be the case that you have configured variables that do not exist in the Plc. This may be because you imported variables that you are not using the PLC, or you have removed blocks from the PLC, but the variables still exists. In any case, you must check the variables that present this error.

"Access denied"

the PLC actively rejected the communication request. There are some settings that are not correctly set, a Password was set in the PLC, or some firewall is rejecting connections. Usually this means that you are missing the Required settings in the PLC's Hardware configuration. Connecting to a S7-1500 or S7-1200 PLC

"Unspecified Error while initializing the driver"

This error can happen if there is an error in your VisXpert. You should reinstall VisXpert with the "Reinstall Driver" setting from the installer. It may also be the case that an Antivirus Blocking Plc Driver.

Antivirus Blocking Plc Driver

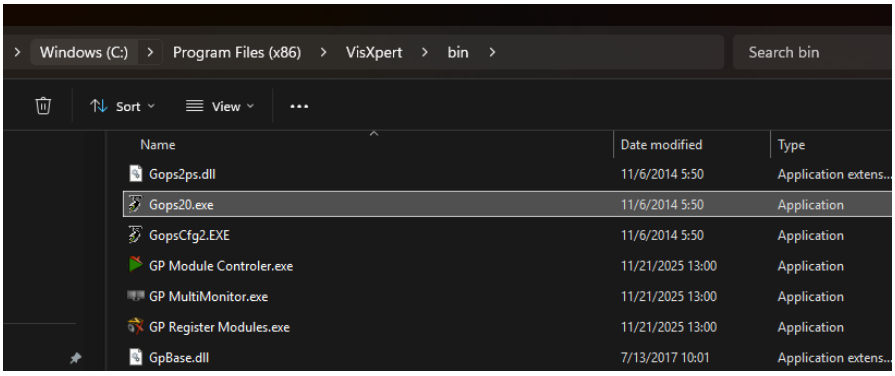
BatchXpert uses the integrated “GOPS” communication driver of VisXpert. This driver is usually installed automatically when you install the VisXpert SCADA system. This usually manifests itself in strange behaviors, where the PLC driver cannot even be started.

It may happen that antivirus software blocks the executable from executing or even being installed during installation. This usually manifests itself in an Antivirus dialog, indicating that it put a file in Quarantine.

In BatchXpert you can diagnose this problem by looking at he is following message in the VisXpert log window:

```
25.12.17 15:00:46.47 Onlinedaten-Server   K01 Log class filters set to 0xFEFF0021 for display and 0xFEFF0001 for file output
25.12.17 15:00:46.47 Onlinedaten-Server   K01 GEFASOFT OPC PLCData-Server V2.0.1.5 gestartet
25.12.17 15:00:46.50 Onlinedaten-Server   K01 COpcClt 0x02640040 angelegt in Thread 0x000053B
25.12.17 15:00:46.50 Onlinedaten-Server   K01 COpcClt 0x02640040: 'start()' aufgerufen
25.12.17 15:00:46.50 Onlinedaten-Server   K01 COpcClt 0x02640040 Thread 0x00005744 gestar
25.12.17 15:00:46.50 Onlinedaten-Server   K01 COpcClt 0x02640040: Start-Event gesetzt
25.12.17 15:00:46.50 Kommunikationsmodul  K29 OpcSrv "!GEFASOFT.GOPS2.1": laden Fehler 0xC4000016()
25.12.17 15:00:46.50 Onlinedaten-Server   K01 COpcClt 0x02640040: wird gleich freigeben, Thread 0x00005B40
25.12.17 15:00:46.50 Kommunikationsmodul  K25 Grp "PLC01" Fehler 0xC4000016(Unbestimmter Fehler)
25.12.17 15:00:46.50 MessageCapture 1     K17 Gruppe PLC01: Verbindungsfehler: 0xC4000016: Unbestimmter Fehler
25.12.17 15:00:46.50 MeasurementLogger 1  K17 Gruppe PLC01: Verbindungsfehler: 0xC4000016: Unbestimmter Fehler
25.12.17 15:00:46.50 Onlinedaten-Server   K01 COpcClt 0x02640040 Thread 0x00005744 beendet
25.12.17 15:00:46.50 Onlinedaten-Server   K01 COpcClt 0x02640040 freigeben in Thread 0x00005B40
25.12.17 15:00:46.50 Onlinedaten-Server   K01 COpcClt 0x02640040 angelegt in Thread 0x00005B40
25.12.17 15:00:46.50 Onlinedaten-Server   K01 COpcClt 0x02640040: 'start()' aufgerufen
25.12.17 15:00:46.50 Onlinedaten-Server   K01 COpcClt 0x02640040: Start-Event gesetzt
25.12.17 15:00:46.50 Onlinedaten-Server   K01 COpcClt 0x02640040 Thread 0x00005428 gestartet
25.12.17 15:00:46.50 Kommunikationsmodul  K29 OpcSrv "!GEFASOFT.GOPS2.1": laden Fehler 0xC4000016()
25.12.17 15:00:46.50 Onlinedaten-Server   K01 COpcClt 0x02640040: wird gleich freigeben, Thread 0x00005B40
25.12.17 15:00:46.50 Onlinedaten-Server   K01 COpcClt 0x02640040 Thread 0x00005428 beendet
25.12.17 15:00:46.50 Onlinedaten-Server   K01 COpcClt 0x02640040 freigeben in Thread 0x00005B40
25.12.17 15:00:46.58 Kommunikationsmodul  K25 Grp "PLC02" Fehler 0xC4000016(Unbestimmter Fehler)
25.12.17 15:00:46.58 MessageCapture 1     K17 Gruppe PLC02: Verbindungsfehler: 0xC4000016: Unbestimmter Fehler
25.12.17 15:00:46.58 MeasurementLogger 1  K17 Gruppe PLC02: Verbindungsfehler: 0xC4000016: Unbestimmter Fehler
25.12.17 15:00:52.52 Onlinedaten-Server   K01 Beendet!
```

The important part here is the “laden Fehler” which translates to “Loading error” indicating, that the driver could not even be loaded. In that case you must make sure that an antivirus is not blocking any executable and that the Driver executable exists (it may have been removed or blocked during installation by your Antivirus Software).



Make sure that the following executable exists and is not blocked by your Antivirus.

Endianness

One important characteristic of Simatic 7 controllers is that these types of controllers use an CPU with Big endian architecture. This contrasts with most computer systems which are based on the X86 CPU architecture, and which uses little endian.

The Endianness of CPU architecture defines how integers are stored in memory and thus also defines how the memory of these integers is laid out. This becomes important when you are transferring data between two different CPU architectures with different endianness, such as when you are reading data from a Simatic 7 CPU on a SCADA operating station. You must keep the endianness in mind since the data layout of status words from control modules individualization data blocks has to be adapted accordingly on the SCADA systems.

In addition to this you must keep in mind that schematic PLCS implement and byte-oriented memory model whereas modern SCADA systems implement and word or double word-oriented memory model. For example, what would be Byte 1.2 would become bit 18 on the SCADA system, since the endianness is going to be corrected by the corresponding IO communication driver.

Batch expert uses double integers (int32) as their status words for all control modules, which means that all statuses from the PLC must be converted by the following table on and Scada station:

SCADA bit number	Address in PLC
24	0.0
25	0.1
26	0.2
27	0.3
28	0.4
29	0.5
30	0.6
31	0.7
16	1.0
17	1.1
18	1.2
19	1.3
20	1.4
21	1.5
22	1.6
23	1.7
8	2.0
9	2.1
10	2.2
11	2.3
12	2.4
13	2.5
14	2.6
15	2.7
0	3.0
1	3.1
2	3.2
3	3.3
4	3.4
5	3.5
6	3.6
7	3.7

HMI Tag names

In the following description we will refer to the default batch expert take naming scheme as implemented in the VisXpert SCADA system. These tag names define symbolic names for all data that must be communicated between the PLC and the HMI system. You can find a detailed list attached to this manual.
The provided variable list defines the following naming schemes:

Control Module Tags

All variables related to control models are composed of three letter control module short name and four-digit control module number, and dot-separated a post fix denominating its data. For Example:

AIn0001.Sp
AIn0001.St
AIn0001.Val

The Ain denominates an Analog Input, the 0001 denominates Analog Input 1, and the ".Sp", ".St" and ".Val" the actual datapoint of the module. Other Examples are:

Act1273.St Status of Actuator 1273
Cnt0188.Val Current Value of Counter 188
DIn0327.St Status of Digital Input 327
...

Control Module Post fixes

Post fix	Data Type	Description
.St	Dint (Int32)	Current Status of the Module. This is a "Bitfield" where each bit represents an individual Status of a Control Module. Please check the appropriate section for information about the individual bits of this status.
.Sp	Real	Setpoint. The current Nominal Value for the Control Module. Only applies to Regulators (PID) and Frequency Converters (FConv)
.Val	Real	Current Value. The current value of an analog Measurement, counter, or frequency drive
.Out	Real	Output. The current Analog Output between 0 a 100% of the module. Only applies to Regulators (PID) and FConv

Parameter Channel variables

For easier identification of variables related to parameter channels, all parameter channel variables start with a lowercase "z", followed by the control module's short name and the parameter channel number which is PC01.

- zActPC01. Data of Parameter channel 1 for Actuators
- zDInPC01. Data of Parameter channel 1 for Digital Inputs
- zAinPC01. Data of Parameter channel 1 for Analog Inputs

Status Tag Definitions of Control modules

As mentioned in the Endianness section of this manual the addresses from the PLC must be converted to appropriate addresses for the HMI system. For example, the very first address pit 0.0 in the PLC becomes bit 24 on the SCADA system after the driver uploaded the data and converted it by its endianness. Each individual bit of the control module status has the following meaning:

Actuator

HMI Address	PLC Address	Symbol	Type	Remark
24	0.0	ACo	BOOL	automatic control
25	0.1	ExCo	BOOL	extern control
26	0.2	SCS	BOOL	status check start
27	0.3	xFBa1	BOOL	feedback 1
28	0.4	xFBa2	BOOL	feedback 2
29	0.5	Rel	BOOL	release
30	0.6	Rel2	BOOL	release 2
31	0.7	xAuto	BOOL	External automatic
16	1.0	ACoHM	BOOL	automatic control help memory
17	1.1	ExCoHM	BOOL	Extern control help memory
18	1.2	FBaOn	BOOL	feedback ON intern
19	1.3	FBaOff	BOOL	feedback OFF intern
20	1.4	FBaChange	BOOL	change extern feedback (0 FBa1=OFF FBa2=ON / 1 FBa1=ON FBa2=OFF)
21	1.5	FBa1Active	BOOL	feedback 1 active
22	1.6	FBa2Active	BOOL	feedback 2 active
23	1.7	xAutoHM	BOOL	extern automatic old
8	2.0	GAlQuitt	BOOL	general alarm acknowledge
9	2.1	Ign	BOOL	ignore alarm
10	2.2	Sim	BOOL	simulation
11	2.3	Auto	BOOL	automatic mode
12	2.4	MCo	BOOL	manual control
13	2.5	EmRel	BOOL	emergency release
14	2.6	InterlockGAl	BOOL	interlock by alarm
15	2.7	Maint	BOOL	maintenance
0	3.0	GAl	BOOL	general alarm
1	3.1	GAlS	BOOL	general alarm save
2	3.2	SCE	BOOL	status check error
3	3.3	Mov	BOOL	Actuator is moving for visu
4	3.4	On	BOOL	actuator is ON
5	3.5	Off	BOOL	actuator is OFF
6	3.6	Out	BOOL	output
7	3.7	User	BOOL	free for user program

Digital Input

HMI Address	PLC Address	Symbol	Type	Remark
24	0.0	EA0	BOOL	enable alarm by 0-signal
25	0.1	EA1	BOOL	enable alarm by 1-signal
26	0.2	SCS0	BOOL	status check alarm by 0-signal
27	0.3	SCS1	BOOL	status check alarm by 1-signal
28	0.4	xSig	BOOL	signal extern
29	0.5	B29	BOOL	spare
30	0.6	B30	BOOL	spare
31	0.7	B31	BOOL	spare
16	1.0	AlHM	BOOL	help memory for alarm
17	1.1	ImpHM	BOOL	help memory for impulse
18	1.2	xSigHM	BOOL	signal extern help memory
19	1.3	B19	BOOL	spare
20	1.4	B20	BOOL	spare
21	1.5	B21	BOOL	spare
22	1.6	B22	BOOL	spare
23	1.7	B23	BOOL	spare
8	2.0	GAlQuitt	BOOL	general alarm acknowledge
9	2.1	Ign	BOOL	ignore alarm
10	2.2	Sim	BOOL	simulation
11	2.3	iEA0	BOOL	intern alarm by 0
12	2.4	iEA1	BOOL	intern alarm by 1
13	2.5	ImpProt	BOOL	write impulse flank to protocol
14	2.6	ImpNegProt	BOOL	write negative impulse flank to protocol
15	2.7	Switch	BOOL	convert as switch output
0	3.0	GAl	BOOL	general alarm
1	3.1	GAlS	BOOL	general alarm save
2	3.2	SCE	BOOL	status check error
3	3.3	Sig	BOOL	signal state
4	3.4	Imp	BOOL	impulse flank
5	3.5	ImpNeg	BOOL	negative impulse flank
6	3.6	B06	BOOL	spare
7	3.7	User	BOOL	free for user

Analog Input

HMI Address	PLC Address	Symbol	Type	Remark
24	0.0	ELLA	BOOL	enable low low alarm
25	0.1	EHHA	BOOL	enable high high alarm
26	0.2	xAI	BOOL	alarm from extern
27	0.3	NPA	BOOL	no periphery adaption
28	0.4	B28	BOOL	spare
29	0.5	B29	BOOL	spare
30	0.6	B30	BOOL	spare
31	0.7	B31	BOOL	spare
16	1.0	MLLA	BOOL	low low alarm - alarm if enabled
17	1.1	MLL	BOOL	low low limit - warning if enabled
18	1.2	ML	BOOL	low limit
19	1.3	MSp	BOOL	setpoint
20	1.4	MH	BOOL	high limit
21	1.5	MHH	BOOL	high high limit - warning if enabled
22	1.6	MHHA	BOOL	high high alarm - alarm if enabled
23	1.7	MHWA	BOOL	alarm from hardware
8	2.0	GAlQuitt	BOOL	general alarm acknowledge
9	2.1	Ign	BOOL	ignore alarm
10	2.2	Sim	BOOL	simulation
11	2.3	iEHWA	BOOL	enable hardware alarm
12	2.4	iELLA	BOOL	enable LL alarm
13	2.5	iEHHA	BOOL	enable HH alarm
14	2.6	iELLW	BOOL	enable LL warning
15	2.7	iEHHW	BOOL	enable HH warning
0	3.0	GAI	BOOL	general alarm
1	3.1	GAIS	BOOL	general alarm save
2	3.2	Warn	BOOL	general warning
3	3.3	Filter1	BOOL	filter 1 on (75%)
4	3.4	Filter2	BOOL	filter 2 on (88%)
5	3.5	Filter3	BOOL	filter 3 on (94%)
6	3.6	ManuInp	BOOL	manual input (no periphery)
7	3.7	User	BOOL	memory free for user

Regulator

HMI Address	PLC Address	Symbol	Type	Remark
24	0.0	EAl	BOOL	Enable alarm
25	0.1	SCS	BOOL	status check start
26	0.2	MStC	BOOL	static output value
27	0.3	MStrt	BOOL	starting value
28	0.4	MOVMin	BOOL	output value min.
29	0.5	MOVMax	BOOL	output value max.
30	0.6	OVOn	BOOL	output value on
31	0.7	B31	BOOL	spare
16	1.0	B16	BOOL	spare
17	1.1	B17	BOOL	spare
18	1.2	B18	BOOL	spare
19	1.3	B19	BOOL	spare
20	1.4	AlHM	BOOL	help memory for alarm
21	1.5	AHystHM	BOOL	help memory outside hysteresis
22	1.6	StrtHM	BOOL	help memory starting value active
23	1.7	Warn	BOOL	warning
8	2.0	GAlQuitt	BOOL	general alarm acknowledge
9	2.1	Ign	BOOL	ignore alarm
10	2.2	Sim	BOOL	simulation
11	2.3	MCOn	BOOL	mode controller on (0=off)
12	2.4	MSpExt	BOOL	mode setpoint extern (0=intern)
13	2.5	DisOut	BOOL	disable output periphery (0=enable)
14	2.6	EW	BOOL	enable warning
15	2.7	B15	BOOL	spare
0	3.0	GAl	BOOL	general alarm
1	3.1	GAlS	BOOL	general alarm save
2	3.2	SCE	BOOL	status check error
3	3.3	Filter1	BOOL	filter 1 on (75%)
4	3.4	Filter2	BOOL	filter 2 on (88%)
5	3.5	Filter3	BOOL	filter 3 on (94%)
6	3.6	CA	BOOL	control acting (1 = inverse)
7	3.7	User	BOOL	memory free for user

Counter

HMI Address	PLC Address	Symbol	Type	Remark
24	0.0	EAImp	BOOL	enable impulse alarm
25	0.1	ELLA	BOOL	enable low low alarm
26	0.2	EHHA	BOOL	enable high high alarm
27	0.3	xAI	BOOL	alarm from extern
28	0.4	ResetBlock	BOOL	interlock counter reset
29	0.5	xSig	BOOL	impulse input
30	0.6	B30	BOOL	spare
31	0.7	B31	BOOL	spare
16	1.0	MLLA	BOOL	low low alarm - alarm if enabled
17	1.1	MLL	BOOL	low low limit - warning if enabled
18	1.2	ML	BOOL	low limit
19	1.3	MSp	BOOL	setpoint
20	1.4	MH	BOOL	high limit
21	1.5	MHH	BOOL	high high limit - warning if enabled
22	1.6	MHHA	BOOL	high high alarm - alarm if enabled
23	1.7	ImpHM	BOOL	impulse help memory
8	2.0	GAIQuitt	BOOL	general alarm acknowledge
9	2.1	Ign	BOOL	ignore alarm
10	2.2	Sim	BOOL	simulation
11	2.3	Reset	BOOL	reset counter
12	2.4	iELLA	BOOL	counting reserve
13	2.5	iEHHA	BOOL	enable HH alarm
14	2.6	iELLW	BOOL	enable LL warning
15	2.7	iEHHW	BOOL	enable HH warning
0	3.0	GAI	BOOL	general alarm
1	3.1	GAIS	BOOL	general alarm save
2	3.2	Warn	BOOL	general warning
3	3.3	Imp	BOOL	impulse flank
4	3.4	B04	BOOL	spare
5	3.5	B05	BOOL	spare
6	3.6	B06	BOOL	spare
7	3.7	User	BOOL	memory free for user

Message

HMI Address	PLC Address	Symbol	Type	Remark
24	0.0	B24	BOOL	spare
25	0.1	B25	BOOL	spare
26	0.2	B26	BOOL	spare
27	0.3	B27	BOOL	spare
28	0.4	xAlarm	BOOL	signal extern for alarm condition
29	0.5	B29	BOOL	spare
30	0.6	B30	BOOL	spare
31	0.7	B31	BOOL	spare
16	1.0	B16	BOOL	spare
17	1.1	B17	BOOL	spare
18	1.2	B18	BOOL	spare
19	1.3	B19	BOOL	spare
20	1.4	B20	BOOL	spare
21	1.5	B21	BOOL	spare
22	1.6	B22	BOOL	spare
23	1.7	B23	BOOL	spare
8	2.0	GAlQuitt	BOOL	general alarm acknowledge
9	2.1	Ign	BOOL	ignore alarm
10	2.2	Sim	BOOL	simulation
11	2.3	OPMsg	BOOL	operator message
12	2.4	B12	BOOL	spare
13	2.5	B13	BOOL	spare
14	2.6	B14	BOOL	spare
15	2.7	B15	BOOL	spare
0	3.0	GAI	BOOL	general alarm
1	3.1	GAIS	BOOL	general alarm save
2	3.2	OPMsgActive	BOOL	operator message active
3	3.3	AlarmMsgActive	BOOL	alarm message active
4	3.4	iAlarm	BOOL	Alarm active intern
5	3.5	B05	BOOL	spare
6	3.6	B06	BOOL	spare
7	3.7	User	BOOL	free for user

Switch

HMI Address	PLC Address	Symbol	Type	Remark
24	0.0	Set	BOOL	set software switch
25	0.1	Reset	BOOL	reset software switch
26	0.2	B26	BOOL	spare
27	0.3	B27	BOOL	spare
28	0.4	B28	BOOL	spare
29	0.5	B29	BOOL	spare
30	0.6	B30	BOOL	spare
31	0.7	B31	BOOL	spare
16	1.0	B16	BOOL	spare
17	1.1	B17	BOOL	spare
18	1.2	B18	BOOL	spare
19	1.3	B19	BOOL	spare
20	1.4	B20	BOOL	spare
21	1.5	B21	BOOL	spare
22	1.6	B22	BOOL	spare
23	1.7	B23	BOOL	spare
8	2.0	B08	BOOL	spare
9	2.1	B09	BOOL	spare
10	2.2	B10	BOOL	spare
11	2.3	B11	BOOL	spare
12	2.4	B12	BOOL	spare
13	2.5	B13	BOOL	spare
14	2.6	B14	BOOL	spare
15	2.7	B15	BOOL	spare
0	3.0	B00	BOOL	spare
1	3.1	B01	BOOL	spare
2	3.2	B02	BOOL	spare
3	3.3	Sig	BOOL	spare
4	3.4	B04	BOOL	spare
5	3.5	B05	BOOL	spare
6	3.6	B06	BOOL	spare
7	3.7	User	BOOL	free for user

Alarm Groups

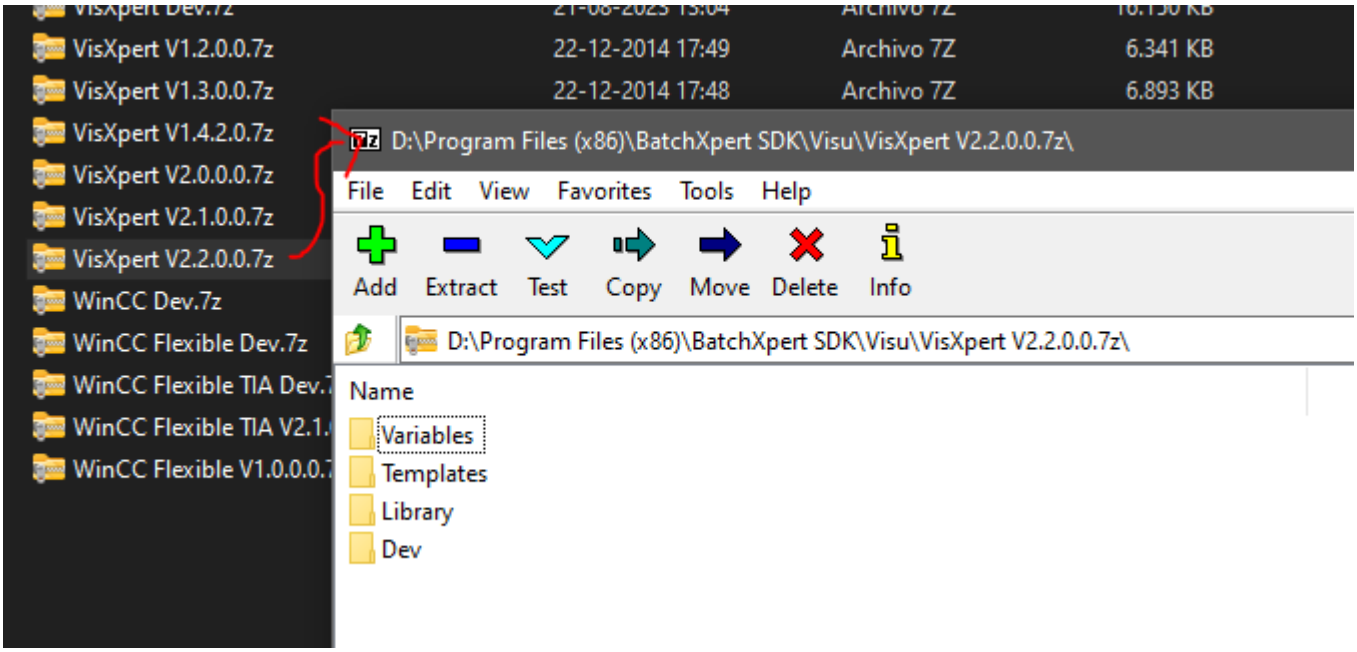
HMI Address	PLC Address	Symbol	Type	Remark
24	0.0	B24	BOOL	set software switch
25	0.1	Ign	BOOL	reset software switch
26	0.2	Sim	BOOL	spare
27	0.3	Auto	BOOL	spare
28	0.4	B28	BOOL	spare
29	0.5	EmRel	BOOL	spare
30	0.6	B30	BOOL	spare
31	0.7	Maint	BOOL	spare
16	1.0	Gal	BOOL	spare
17	1.1	Gals	BOOL	spare
18	1.2	SCE	BOOL	spare
19	1.3	Warn	BOOL	spare
20	1.4	Msg	BOOL	spare
21	1.5	ProcRun	BOOL	spare
22	1.6	ProcProd	BOOL	spare
23	1.7	ProcCIP	BOOL	spare
8	2.0	B08	BOOL	spare
9	2.1	B09	BOOL	spare
10	2.2	B10	BOOL	spare
11	2.3	B11	BOOL	spare
12	2.4	B12	BOOL	spare
13	2.5	B13	BOOL	spare
14	2.6	B14	BOOL	spare
15	2.7	B15	BOOL	spare
0	3.0	B00	BOOL	spare
1	3.1	B01	BOOL	spare
2	3.2	B02	BOOL	spare
3	3.3	Sig	BOOL	spare
4	3.4	B04	BOOL	spare
5	3.5	B05	BOOL	spare
6	3.6	B06	BOOL	spare
7	3.7	User	BOOL	free for user

The HMI Library

When installing the BatchXpert SDK, this also installs multiple HMI libraries into your BatchXpert SDK installation directory:
“D:\Program Files (x86)\BatchXpert SDK\Visu”

This directory contains an image library, with many images that you can implement into your process screens. You can find images for different machines, brewing equipment and other type of symbols that are helpful to make appealing process screens.

VisXpert archive contains all templates, variable tables, process symbol library and a Project Template for you to use in your project. In these archives you can find all necessary data to update or create new HMI applications.

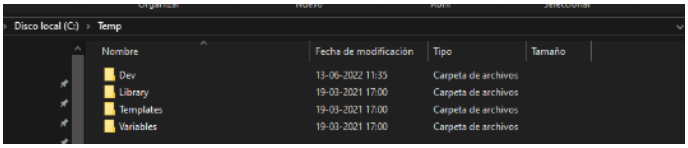


To update existing HMI applications to newer BatchXpert HMI versions, you should consult the appropriate Manual, which explains in more detail how to go about updating an existing HMI application from a newer VisXpert library
If you want to update your existing HMI application to the newest library version, you should refer to the dedicated manual called “Manual BatchXpert Updating VisXpert HMI” for this, which describes the update procedure in more detail.

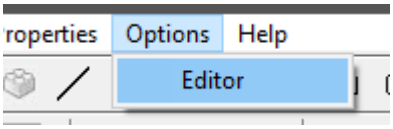
Working with the HMI Library

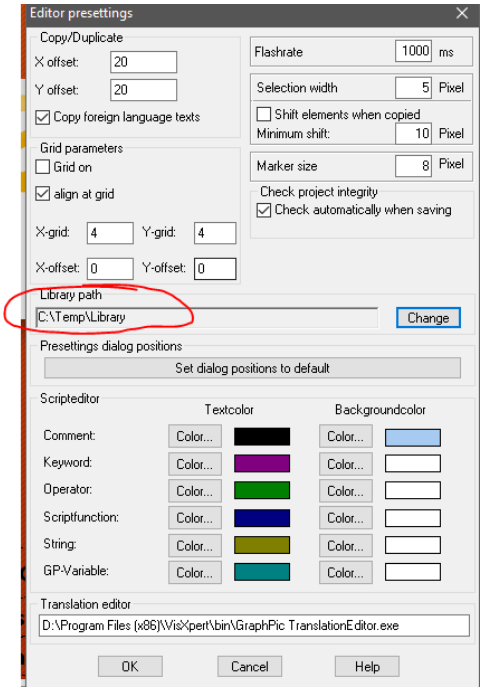
When working with an HMI library, you want to 1st extract the content of this HMI version that you want to use into a working directory, for example: “C:\Temp”, and then point do you VisXpert library Path to the newly created library objects. You can do this in the editor settings of your graphics editor. After setting up the library directory, all Classes, Windows and script objects can be updated from this library.

Extract to Working directory, in this case C:\Temp



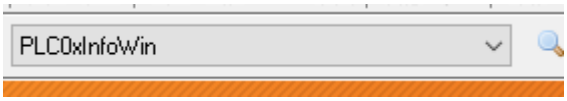
Point Graphics Editor Library path to this Working Directory

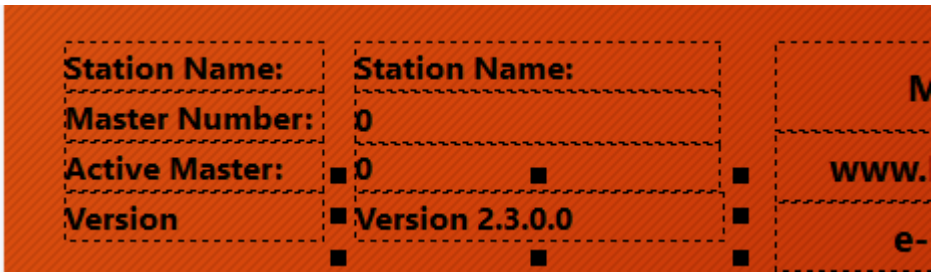




Checking your HMI Version

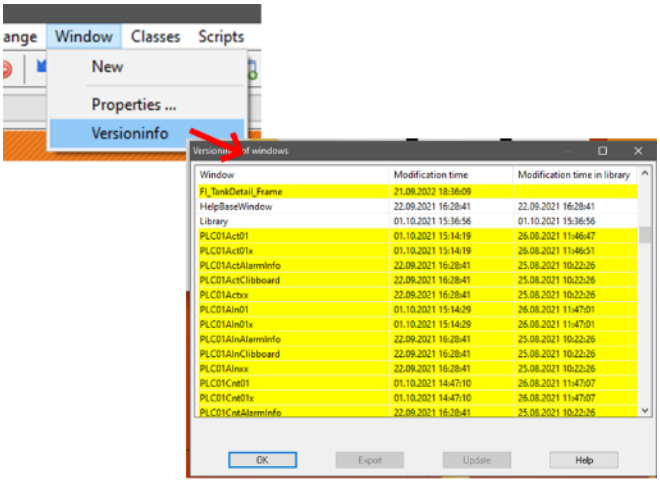
The current HMI version that you are using in your project is always available in the "**PLC0xInfoWin**" HDMI screen, which is available in your graphics editor and during runtime. For more detailed information about which library objects and windows are currently in use, you can use the version control integrated into your graphics editor.





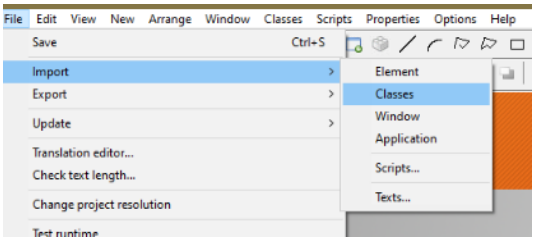
Checking Version of Individual Classes and Windows

Windows and Class objects have individual version numbers that can be checked against your extracted library by using the "Window->version info" or "Classes->version info" option from the main menu. This option opens on dialog that allows you to update existing classes.

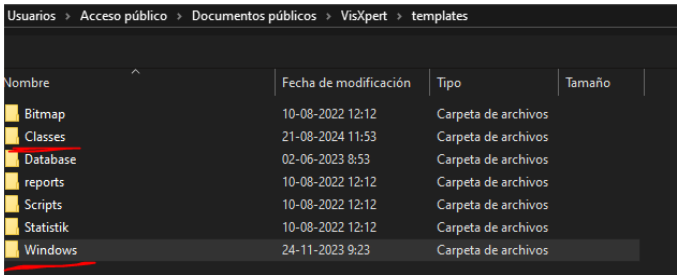


Importing Windows and Classes

The version info option only allows you to check objects that are already present in your project against the version that exist of this object in the library. If you want to add objects from the library that do not currently exist in your project, you must use the input option from the main menu. From this option you can select the window or class respectively and insert them into your project to be used. After importing the objects, you can use the version info dialog to check their version against the library version.

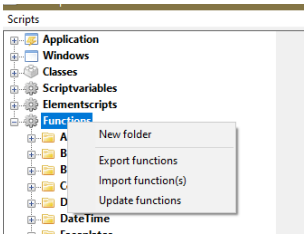


All objects that are part of the library exist in the appropriate subfolder inside your library directory. You just have to select the appropriate object and insert it into your project.



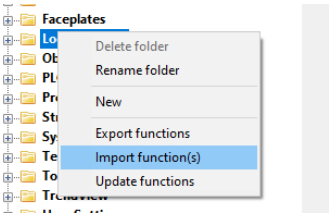
Updating/Importing HMI Scripts

Updating scripts works like updating windows and classes, but from the scripts dialog.



If you choose the “update functions” option, all functions that are currently contained in your project will be updated with the version from the selected library. However sometimes new functions are added to the library which will not be updated by the “update functions” option.

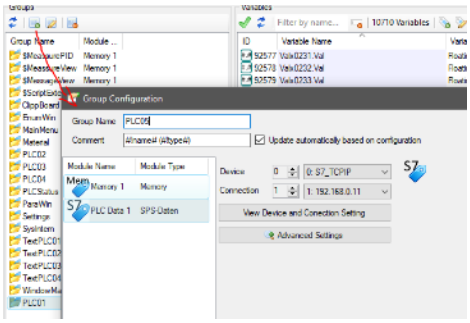
For this you can use the “Import Function(s)” option and select the script file that you want to import. Since you can group different script functions into directories, all import script files are also contained in subdirectories in the library directory. We recommend that you order all these functions in the same subdirectories as they appear in the library directory. The easiest way to do this, is by right clicking directly on the folder containing the script functions that you want to import and not right clicking from the main “Functions” three item but rather from the sub directory below this menu option. This allows you to import only functions belonging to a specific directory.



Adding a new PLC to the HMI

If you want to add a new PLC to your BatchXpert application, you need to add the following components from your corresponding with expert archive, found in your HMI library, to your HMI application.

Adding a new PLC Group

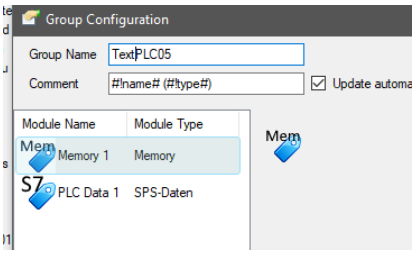


The first action that you should take this that you add a new PC group that corresponds to the PLC that you want to add to the System. You do this by opening the variable editor and removing the existing PLCxx variable group if it already exists as a memory driver. This is because the default BatchXpert project includes PLC02, PLC03 and PLC04 Already added as memory variables to the project. But since you want to add a real PLC you must change these variables to the appropriate PLC driver.

After removing the existing PLC group, you can add a new pill C group and continue with configuration of your communications settings for this PLC.

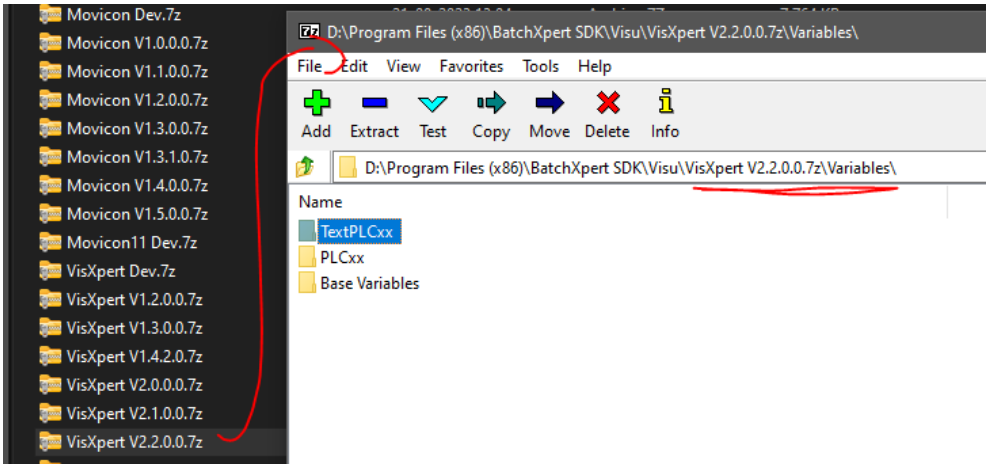
Adding new PLCxx Text Group

BatchXpert also requires some memory variables for each PLC, which are all configured in the TextPLCxx group that belongs to a memory driver.

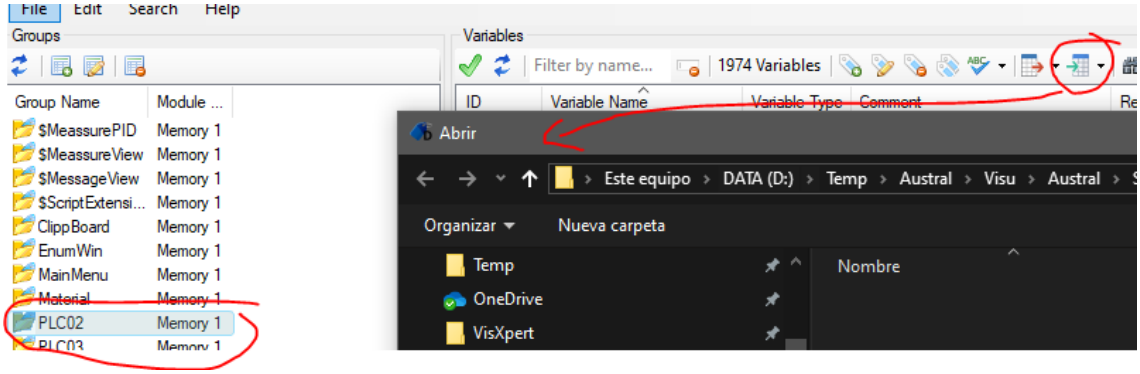


Importing the PLC Variables

After creating a new PLC group, you can now continue to import all variable tables from your HMI library into this newly created variable group. All variables of the “TextPLCxx” directory must be imported into the appropriate “TextPLCxx” memory variable group, and all variables contained in the “PLCxx” and “Base Variables” directories must be imported into the “PLCxx” variable groups. This inputs all variables related to parameter channels, system parameters and all available control models.

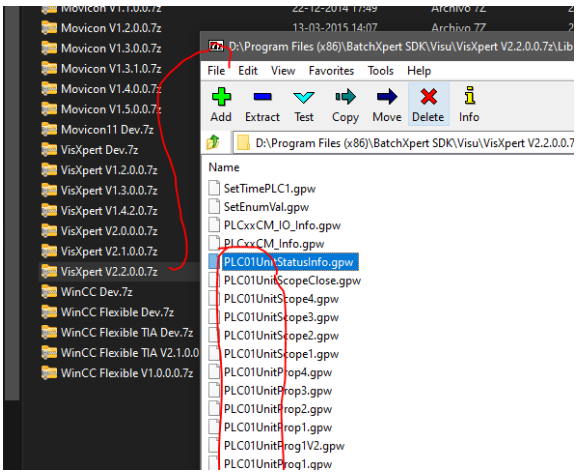


By selecting the appropriate PLC group, you can open the Import dialog from the menu bar on the top and then select the appropriate variable list that you want to import into each of the groups.



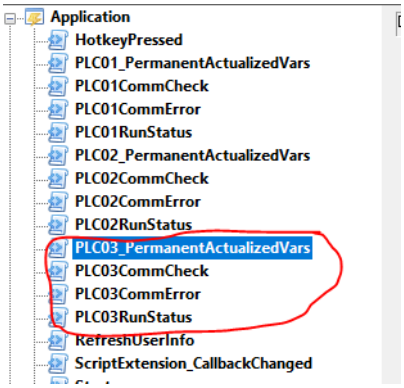
Importing the PLC dependent Windows from your library

After importing all the variables, you only must import all the face plates and PC specific windows for your HMI application. What is windows can also be found in your HMI library and all windows starting with PLCxx must be imported into your HMI application.



Currently there is a defect in VisXpert, which does not allow you to import more than about 10 windows at the same time. To circumvent this please import all windows in batches of at most about 10 windows and repeat the process until all windows are imported.

Adding Global Monitoring Scripts



Each PLC requires some commonly used global application scripts, which monitor the communication and handle some global variables for each PLC. After adding all the windows and variables you also must add these scripts to the application. You can import them from the project library or copy them from an existing PLC and adjust where it is needed.

PLCxx_PermanentActualizedVars

This script is used to maintain references to certain variables that should always be communicated with the PLC even if no window is actively using these variables. This means the variables that are used in these scripts are used so that they get activated by the HMI application and communication gets established with the controllers, even if these variables are not even used in any of the HMI Windows. This is required by some functionality of BatchXpert, so that some Variables remain a steady connection to the PLC.

PLCxxCommCheck

This script handles a live signal from the respective master station to the PLC. PLC sends a life signal that has to be confirmed by each operating station, where each operating station is identified by its master number. This script confirms this life signal, depending on the master number of the local station, to the PLC.

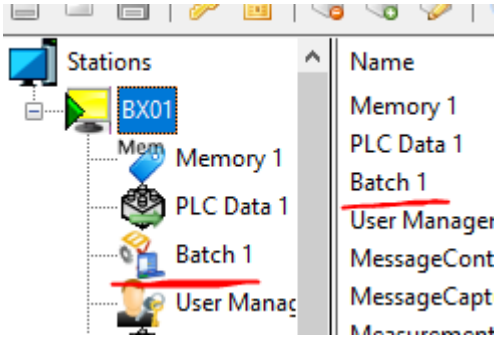
PLCxxCommError

If the PLC driver detects an error with the communication to the PLC it sets the appropriate error status in the “"PLCxx:@ConnectStatus"” variable. This script reacts to changes to this variable and activates the appropriate alarm and the alarm system and also updates the PLC's internal status.

PLCxxRunStatus

This script is used to detect if PLC is actually executing the user application. It monitors the PLC system’s time and whenever it changes it knows that the user program is actually executing. If this counter is not changed in more than 5 seconds it activates the appropriate alarm message to tell the user that it's logic controller maybe in stop mode.
This script also detects if the PLC exists by checking if the timer actually exists.

The Batch Module



BatchXpert adds an “Batch Module” to its VisXpert project structure. This module is responsible for generating trend recording data, trend view xml files, and allowing messages for all control modules that are configured in your project.

The Batch Module starts when the project is started and goes through the configuration of all your control modules and creates new trend recording data, trend few XML configuration files and allow messages in the case that new modules have been added to BatchXpert.

This means that this module always maintains trend view, trend recording and Alarm configuration in sync with your control module configuration of BatchXpert. This synchronization process is started when Module is started, which means that you should restart your project, which in turn restarts the Batch Module, when you have made changes to any of your control modules.

Show Trends with VisXpert

Opening a Trend window from the HMI

To add a Trend View Button to any of the process graphics, you first need to add a button or Bitmap which will act as a button for calling the desired Trend view. Usually you will use the “TrendButton.bmp” as a default trend view invocation object. You can find a preconfigured “Trend button” in the “Library” window.

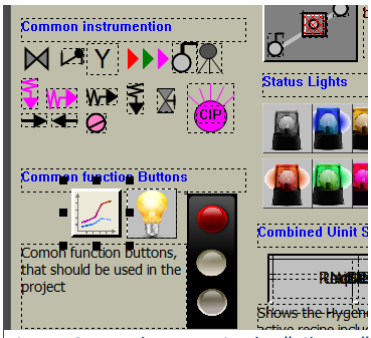


Figure 2 Trend Button in the "Library"

In the Trend button object, you must add an “Click” event, which executes an “Element Script”, which in turn calls one of the “ShowTrendView” functions. BatchXpert provides multiple script functions that you can put into any button or object you need to open the trend view. There are functions to show only the trend view of a specific Analog input, for all Analog values of a unit, or for custom trend views.

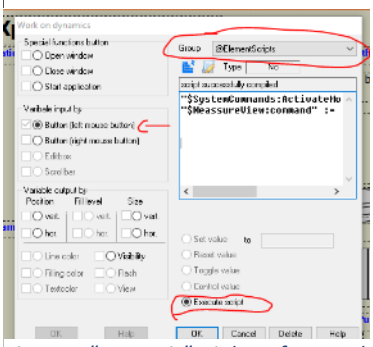


Figure 1 "Dynamic" Dialog of a Trend button

To assign the Script to the Trend button, you must open the “Dynamic” dialog by double clicking the button in the graphics editor or by pressing “Ctrl+d”. This opens the “Dynamic Dialog” which allows you to define the Script that will be executed when a left click happens.

In this script you can call one of the multiple “ShowTrendView” functions to show the desired trend data. You can also define custom trend views, which you can show by simply passing in the name of the “.xml” filename of the trend view data. For custom Trend views, the name of the view that you pass into “ShowTrendView” corresponds to the filename of the xml configuration file of the View. This configuration file defines the trend curves and their configuration in the trend viewer.

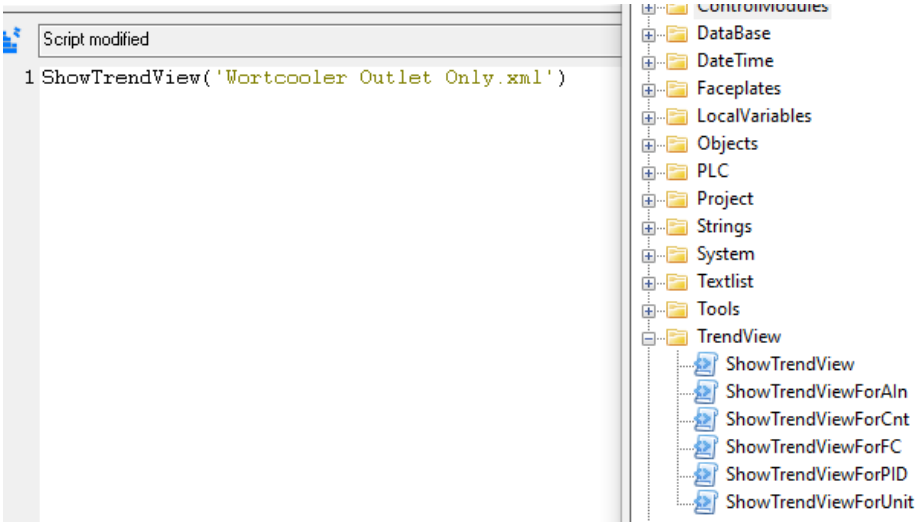
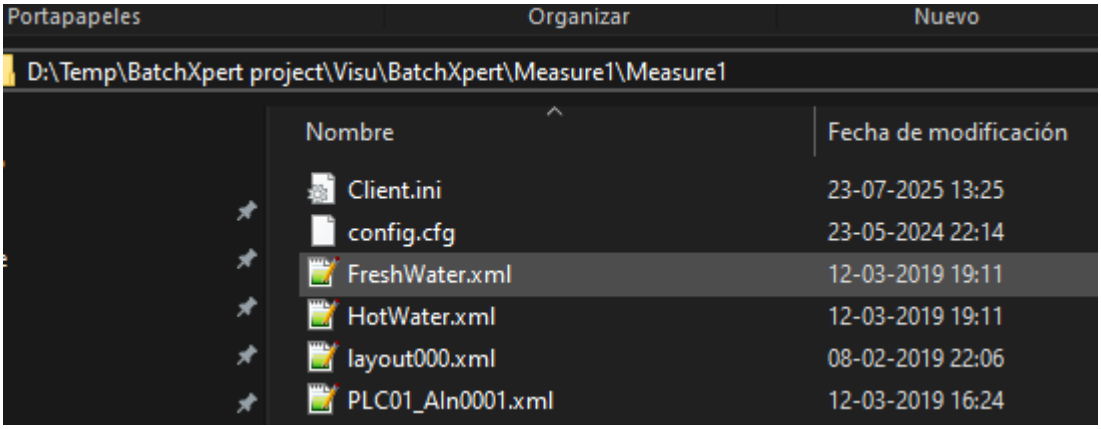


Figure 3 Show Trend Script for a custom trend view

Trend View XML files

The Trend View XML files are configuration files that are used by the “Measurement Viewer” application to configure the trend curves and their configuration to be used. These files store the trend lines that are part of this trend view, their Min and Maximum values, and other values.

These XML files are in the Project Directory:
<BatchXpert Project Directory>\Visu\BatchXpert\Measure1\Measure1\

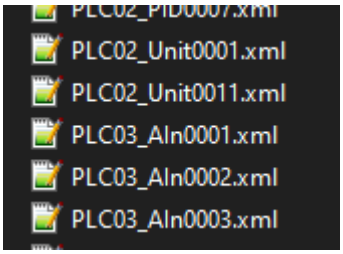


Auto Generated Trend Views

The “Batch Module” in VisXpert automatically generates default Trend views for many object, which can directly be used by calling the appropriate “ShowTrendView” functions from the buttons script. It automatically generates an “Trend View” xml files. It automatically generates files for each analog control module, such as “Analog inputs”, “Counters” or “Regulators”.

For each Unit that is configured in BatchXpert, it also generates an “Unit Trend View”, that contains all analog values that correspond to this Unit.

The auto generated Trend views will follow the following naming convention.

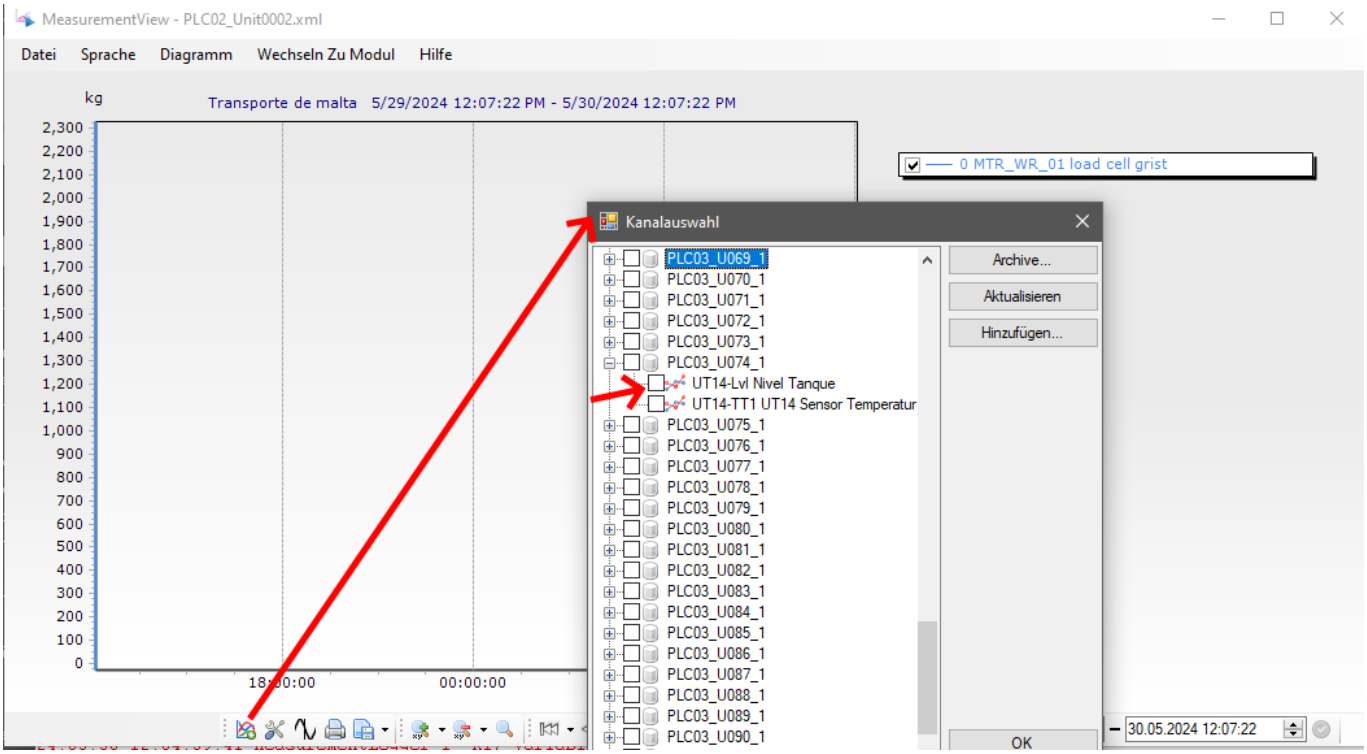


These Auto generated Trend views will be regenerated each time the Project is restarted, since a Project Restart also restarts the “Batch Module”, which in turn generates these trend views when starting up.

Custom Trend views

To add custom or additional trend lines to a trend view, you can do so directly from the “Measurement Viewer” that is being opened when clicking on a trend view button in the HMI screen. From there you can add trend lines from each of the existing archives. After adding the file, you can just save the configuration. The name of the xml Trend view file that you choose, is the one that you will have to use when calling “ShowTrendView” from your Scripts.

By default, each Process unit will have its own trend file.



Record custom Trend Value

Trend recordings are managed by the VisXpert SCADA system. BatchXpert integrates with VisXpert and automatically generates Trend configuration files for all its Control modules that have analog values. This means that usually you do not need to modify the trend configuration at all.

You can, however, add custom values into the trend configuration or change the configuration of automatically generated values. The configuration is automatically generated when the system starts up.

Supported Trend Values

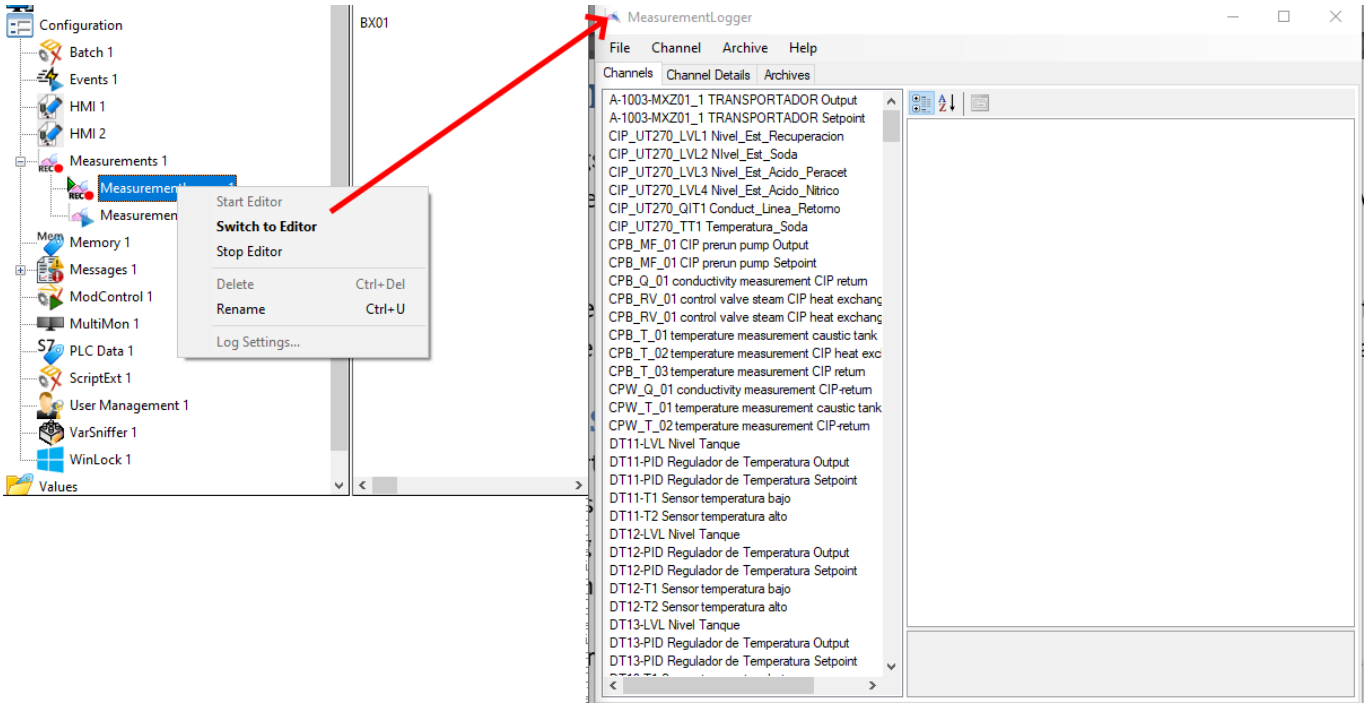
VisXpert supports the following values to be recorded as trends

- Integers
- Floating Point
- Boolean

It does however **not** support recording of variable length data such as “Strings” or “Arrays”

Adding a custom trend to be recorded

The trend recording configuration is done with the “Measurement Recording” editor. This editor allows you to add new “Channels” to the configuration file and assign a trend archive to it.



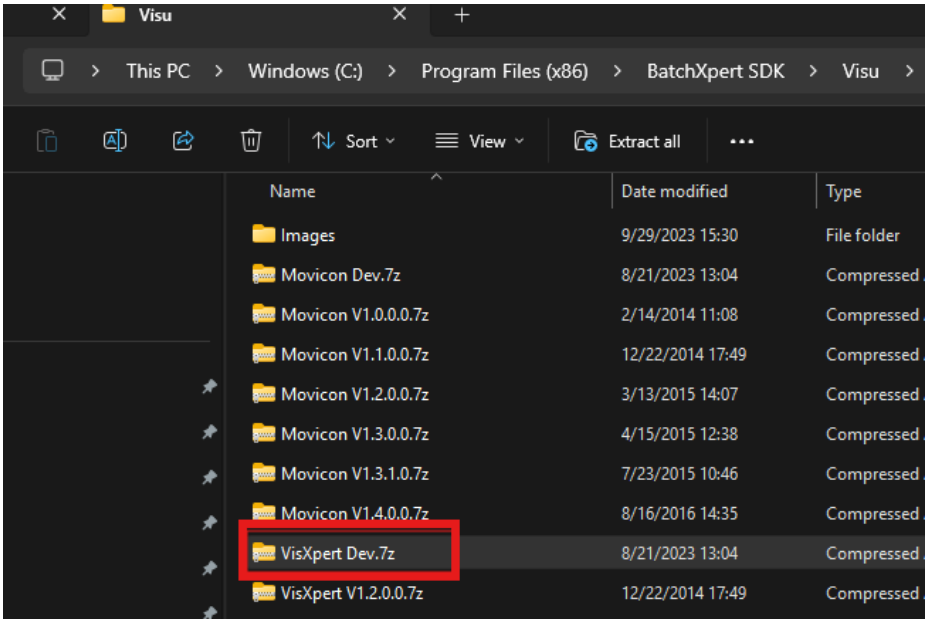
After adding the custom value, you can add them to any of your Trend View configurations for display

Adding Batch Number, Step number and Phase

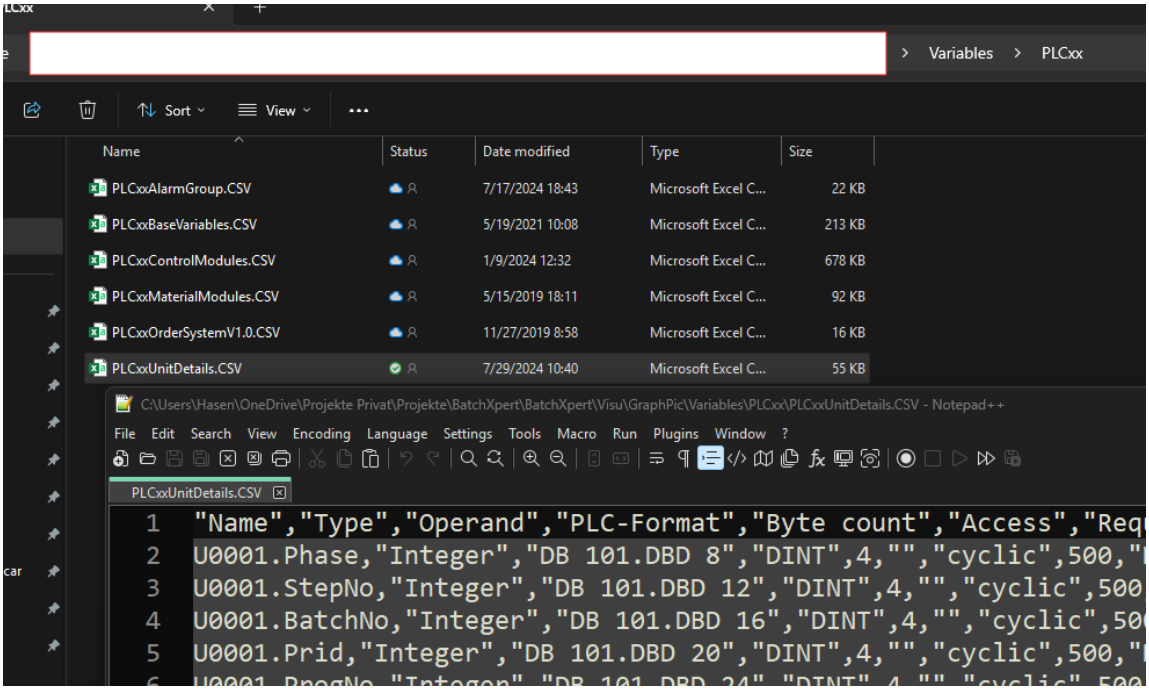
For historical reasons BatchXpert does not have dedicated variables for batch number step number or Phase numbers. However, you can manually add these variables to your variable configuration by importing the corresponding variable list into your “PLCxx” (where xx is your plc number) variable Group in the “SPS Daten” communication Driver.

Locate The Import Tag list

the import table is located inside your BatchXpert SDK directory, usually located in “C:\Program Files (x86)\BatchXpert SDK\Visu”. Inside the corresponding VisXpert archive.

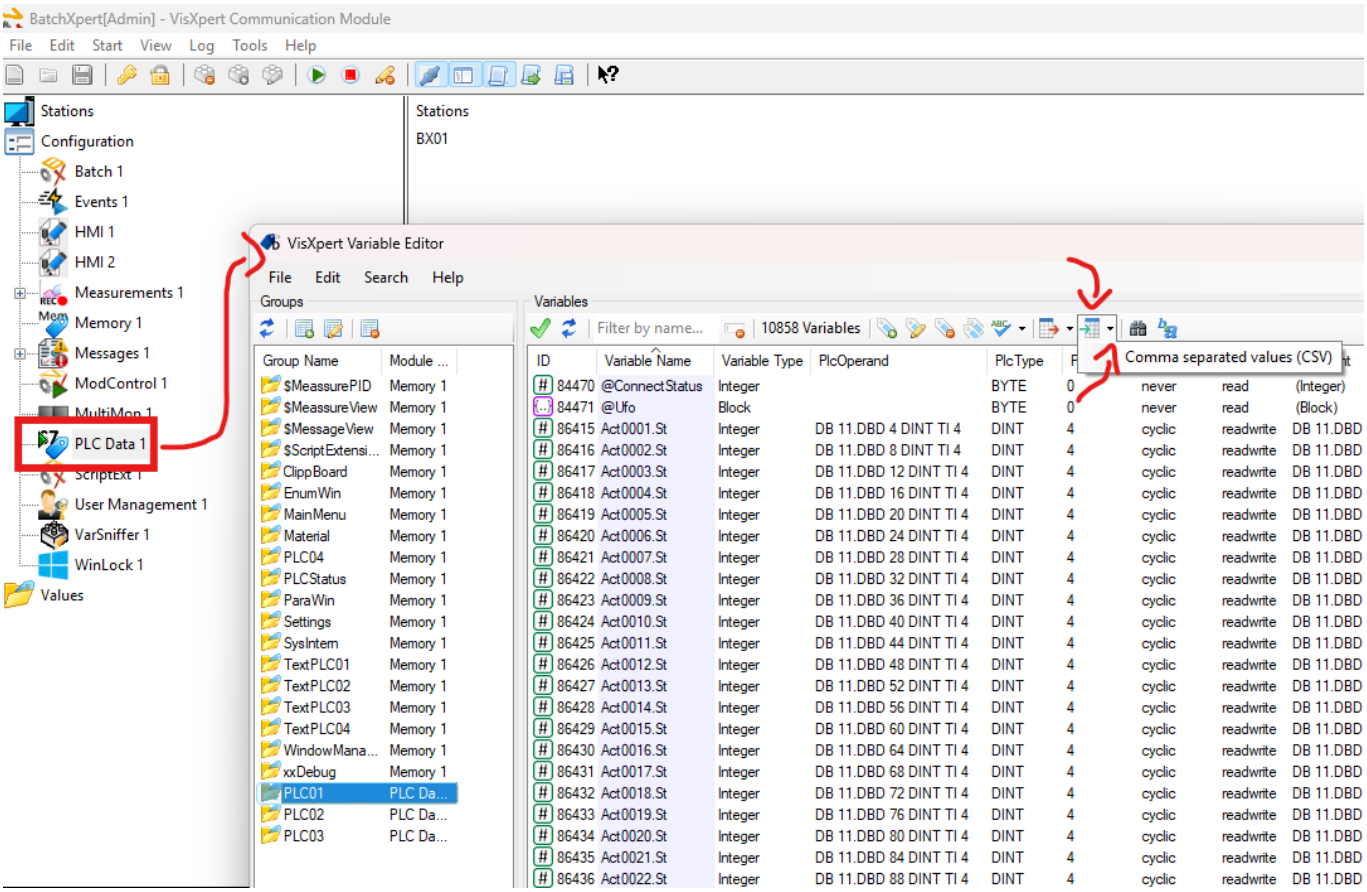


The variable list is called “PLCxxUnitDetails.CSV” a can be found inside the “Variables\PlcXX” directory, inside your VisXpert Archive mentioned above.

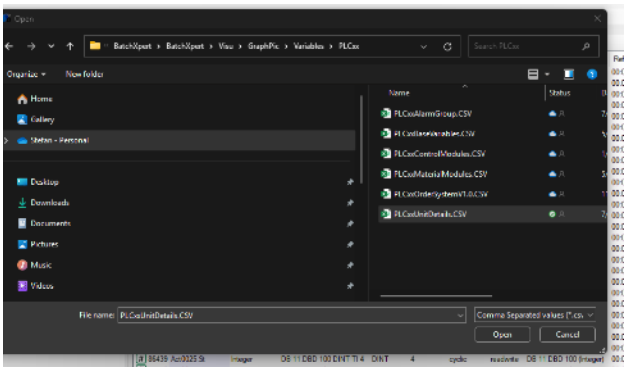


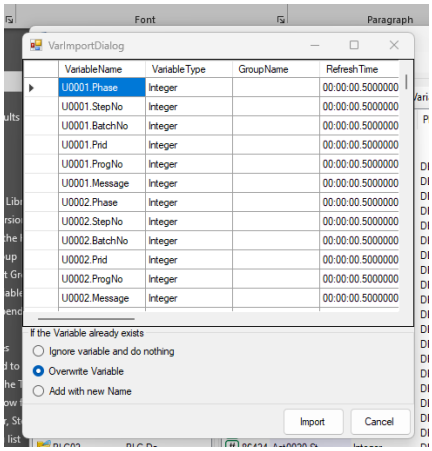
Importing Variables

These variables can be imported by using the variable editor of the “SPS Daten” module from the Communication Module.



From here you can select the file above and select “Overwrite” as import option.





Adding Batch Number to Trends

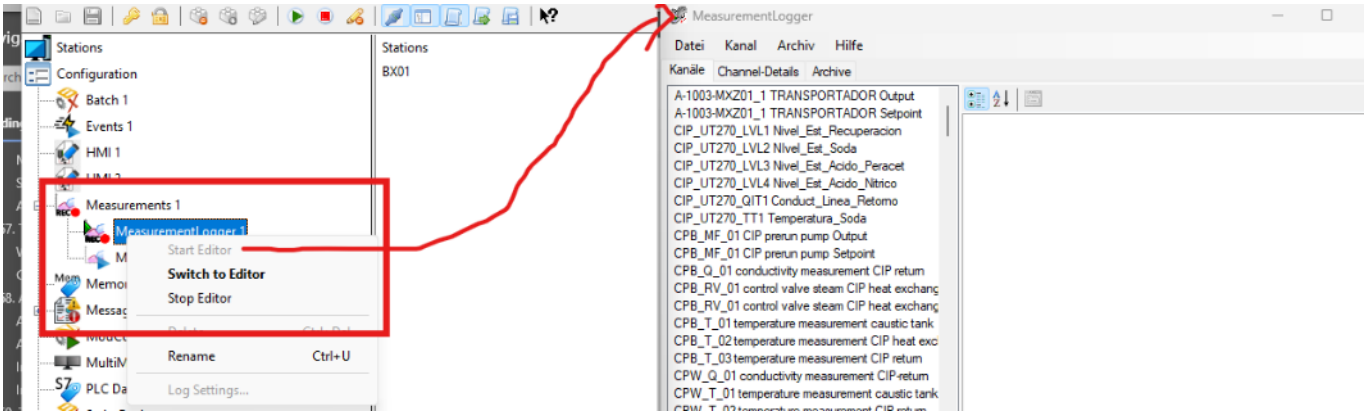
Sometimes it may be helpful to add the batch number of the currently running batch off on units to the Trend Graphs. Since the batch number in BatchXpert is an integer not a string, this can be easily achieved. By default, this batch number is not part of the trend recording but can be manually added easily by following the procedure above.

If your project does not include variables for the Batch number, you must manually import these variables from the current HMI project, in your BatchXpert SDK. Please see Adding Batch Number, Step number and Phase for more details.

Keep in Mind that VisXpert does NOT support Strings or Block variables to be added to the Trend configuration. This does not affect any standard BatchXpert variables, since these are all Integers or Real variables.

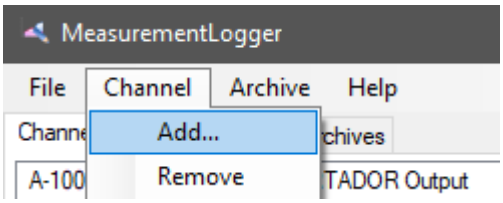
Add a custom Trend Channel to your Tend configuration

Open the “Measurement Logger” editor from the Communications Module.



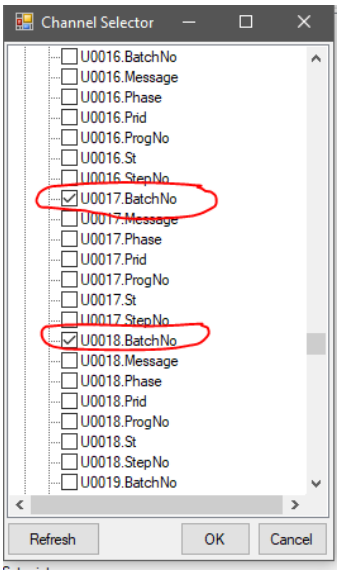
Select “Channel-Add”

This allows you to add custom variables to be added to the Measurement logging, to be recorded and shown in trends.



Select the “BatchNo” of the units that you want to record.

Select the BatchNumber, Step Number or any other variables you need.



Now you can use them in your Trend Views

Recommended Settings for Touch Panel use

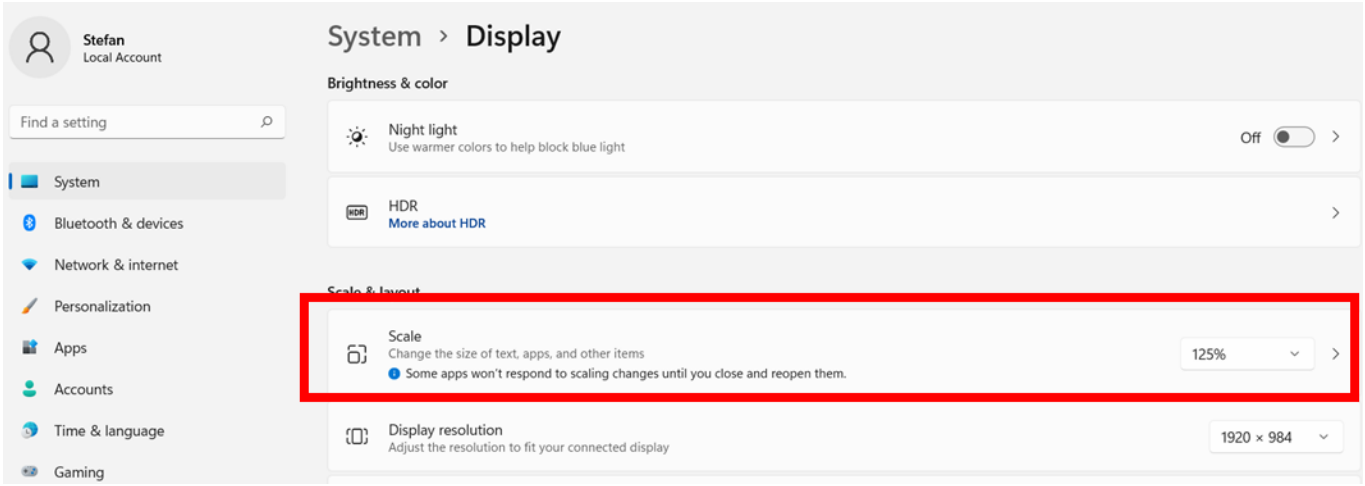
BatchXpert can be used in conjunction with “Touch Panel PC’s” that are typically installed in electrical panels and act as “Field HMI” for operators to be used. BatchXpert is optimized to be used with “traditional” computers, and operated by mouse and keyboard, however it can easily be adapted for “Touch Panel use”.

We recommend that you adjust the following settings in Windows and take the following precautions when engineering an HMI application with BatchXpert.

Set Display Scaling to 125%

Since touch operation requires having bigger buttons, to be able to be reliably being clicked on by touching the screen. In windows you can set the “Display Scale” to 125%, which increases all text, buttons and other UI elements by 25%. This increases all UI elements of windows, and BatchXpert applications to be able to easily click on them by touching the screen.

NOTE: BatchXpert HMI application will not be scaled automatically by increasing the display scale of the operating system. This means that you must adapt some UI elements that you have put in your SCADA screens.



Use bigger Unit symbols

BatchXpert HMI application will not be scaled automatically by increasing the display scale of the operating system. This means that you must adapt some UI elements that you have put in your SCADA screens. The best option to do this is to use the “Unit_ Generic _Big” library object, instead of the regular “Unit_Generic” library object in your HMI screen. This library object provides bigger patterns to make it easier to operate reliably from touch panels.

OnScreenKeyboarManager

When using the operating system, you want the on-screen keyboard to be shown whenever you enter any of the input boxes of any UI element the operating system. However, tests have shown that this functionality does not always work reliably when using touch enabled monitors. For this reason, the SCADA system used by BatchXpert includes a small application that monitors system events and shows the on-screen keyboard whenever an input box is focused.

This small application is automatically started when BatchXpert is started, and a touch monitor is detected. This allows you to use the operating system via the on-screen keyboard. When this application is running a small icon is shown in the taskbar of windows indicating that this application is available and launching the on-screen keyboard whenever an input box is active.

This application is started automatically, and no settings or manual adjustments need to be made.