AUTOMATION CONTROL LAB

BASICS OF SCADA

STUDENT MANUAL

DesignTech
Technology for designing the future

GTTC
GOVERNMENT TOOL ROOM & TRAINING CENTRE
SKILL FOR THE TECHNOLOGY OF THE FUTURE

ptc

AUTOMATION CONTROL LABORATORY

In this lab we use different types of Automation and control techniques, we will work on the different applications according to real-time industry scenarios. Understand the role of programmable logic controllers in complex mechatronic systems, modules, and subsystems.

The list of courses offered,

S.No	Name of the Course	Duration
1	Basics of PLC	50 Hours
2	Basic SCADA	50 Hours
3	Industrial Level control and Batch Process Reactor System	30 Hours
4	Process Instrumentation Technology	30 Hours
5	Advance Process Control Techniques	40 Hours
6	Advanced Industrial Electro-Pneumatic System	40 Hours
7	Industrial Electro-Hydraulic System	40 Hours

BASICS OF SCADA

Supervisory control and data acquisition (SCADA) is a system of software and hardware elements that allows industrial organizations to: Control industrial processes locally or at remote locations. Monitor, gather, and process real-time data.

Some of the industries that use SCADA in their daily operations include water and wastewater systems, electric generation, transmission and distribution systems and oil and gas systems. Manufacturing plants, food production facilities and mass transit systems also rely heavily on SCADA

S. No	Name of the Course	Duration
1	Basics of SCADA	50 Hours

Hardware Equipped

- PLC (Siemens S7-1200)

Software Equipped

- TIA Portal, Win CC

Table of Contents

SCADA Overview

What is SCADA

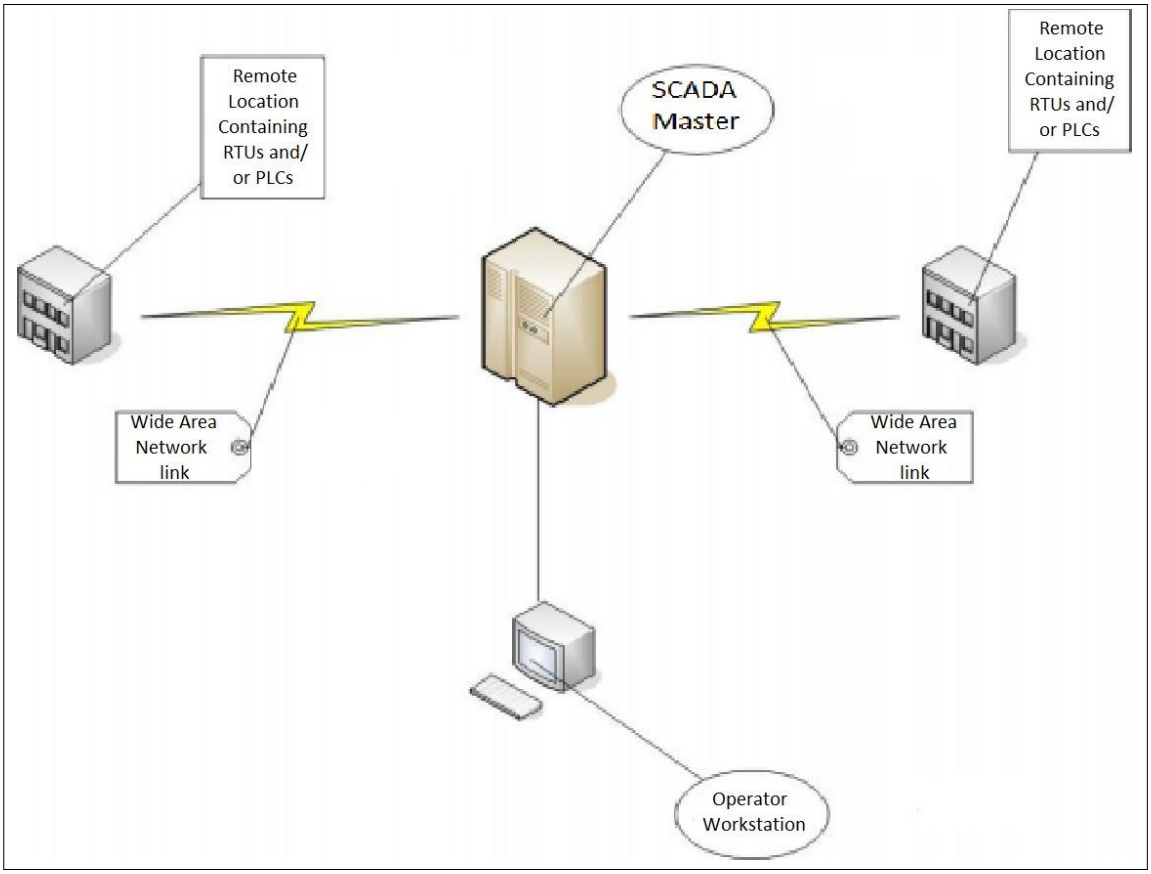
SCADA is an acronym for Supervisory Control and Data Acquisition. SCADA systems are used to monitor and control a plant or equipment in industries such as telecommunications, water and waste control, energy, oil and gas refining and transportation. These systems encompass the transfer of data between a SCADA central host computer and a number of Remote Terminal Units (RTU) and/or Programmable Logic Controllers (PLCs), and the central host and the operator terminals.

A SCADA system gathers information (such as where a leak on a pipeline has occurred), transfers the information back to a central site, then alerts the home station that a leak has occurred, carrying out necessary analysis and control, such as determining if the leak is critical, and displaying the information in a logical and organized fashion. These systems can be relatively simple, such as one that monitors environmental conditions of a small office building, or very complex, such as a system that monitors all the activity in a nuclear power plant or the activity of a municipal water system. Traditionally, SCADA systems have made use of the Public Switched Network (PSN) for monitoring purposes. Today many systems are monitored using the infrastructure of the corporate Local Area Network (LAN)/Wide Area Network (WAN). Wireless technologies are now being widely deployed for purposes of monitoring.

SCADA systems consist of

- One or more field data interface devices, usually RTU, or PLCs, which interface to field sensing devices and local control switchboxes and valve actuators
- A communications system used to transfer data between field data interface devices and control units and the computers in the SCADA central host. The system can be radio, telephone, cable, satellite, etc., or any combination of these.
- A central host computer server or servers (sometimes called a SCADA Centre, master station, or Master Terminal Unit (MTU)
- A collection of standard and/or custom software [sometimes called Human Machine Interface (HMI) software or Man Machine Interface (MMI) software] systems used to provide the SCADA central host and operator terminal application, support the communications system, and monitor and control remotely located field data interface devices.

Typical SCADA system



SCADA functions

A supervisory control and data acquisition (SCADA) system performs the following major functions:

1. Human-machine interface (HMI)
2. Electrical communication
3. Data acquisition (DAQ)
4. Monitoring
5. Control
6. Data collection, storage and retrieval
7. Calculation
8. Report generation

Various functions of SCADA system are now described one-by-one in detail in the following sections.

1) Human-Machine Interface (HMI)

The SCADA system is designed to monitor and control the process/ plant automatically most of the time. However, for various reasons provisions are made for human operators to continuously watch its operation and to intervene as and when felt necessary by them. This requires an interface between the SCADA system and the human operators. The same is provided as a standard practice in the MTU located in the control room. The MTU is built and

functions around a computer. Therefore, the human-SCADA interface is realized through human-computer interface, commonly known as human-machine interface (HMI) and sometimes as graphic operator interface (GOI)

2) Electrical communication

Electrical communication is required:

1. Between the MTU and each RTU
2. Between each RTU and the field devices connected to it.

3) Data acquisition (DAQ)

Data are acquired and processed by RTU and transmitted to MTU on the MTU-RTU data network.

4) Monitoring

It is a common practice to monitor (a) status, (b) events, (c) limits and (d) trends. This Function (monitoring) is carried out jointly by RTU and MTU.

5) Control

Control instructions (set points and discrete control commands) are sent by MTU to the

RTU. The set points received by an RTU are delivered by it to the concerned automatic Controllers.

6) Data collection, storage and retrieval

As explained earlier, each RTU acquires certain data from the controlled process/ plant, processes it appropriately, and then transmits to the MTU at appropriate instants. Some of the data so received by the MTU is stored in the mass-storage media of the MTU. An operator can later on retrieve a block of data of his interest from the storage and recreate an event, sequence or history for visualization and analysis

7) Calculation

Calculations are made both in RTU and MTU. The nature and extent of these calculations are brought out below:

1. Calculations in RTU

   The microprocessor of an RTU is required to perform simple calculations or data processing, such as:

   a. Filtering the data acquired by it to remove noise,
   b. Extraction of desired information, like maximum, minimum, r.m.s or average value or rate of change, from filtered data,
   c. Conversion of numbers to values in engineering units, and
   d. Compression of data to reduce data-transmission-rate and storage requirements.

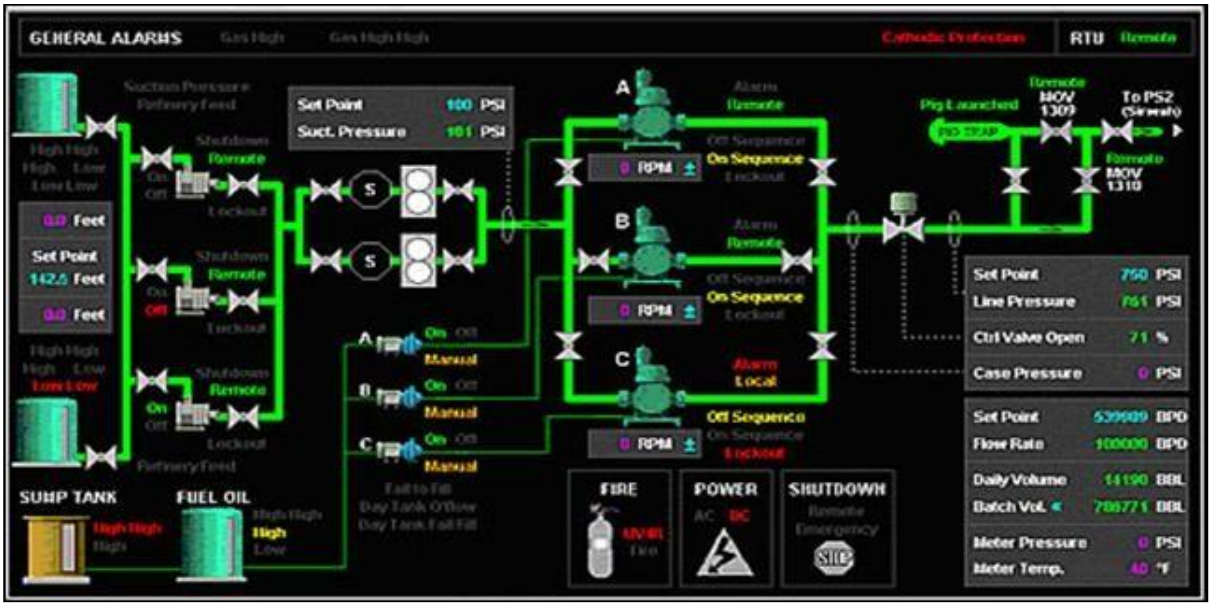
2. Calculations in MTU

The calculations that need to be made by the computer of the MTU are in general fairly extensive and complex. These calculations are made for predicting the behavior of the system (controlled process) through mathematicalmodeling for certain anticipated conditions and certain inputs to the system, both for normal and contingency operation. The output of these calculations is a set control instructions to be sent to different RTU for each set of system conditions and inputs. The calculations are usually made on floating-point numbers and in batch mode.

8) Report generation

One of the important functions of SCADA software is to generate a vast number of reports on the basis of the data stored by the MTU. To that end, SCADA software includes a report generator module, which retrieves data from the MTU database and generates the desired reports from it. The software module allows the user to choose the format of reports, customize the style of reports, insert graphics and even perform calculations.

Examples



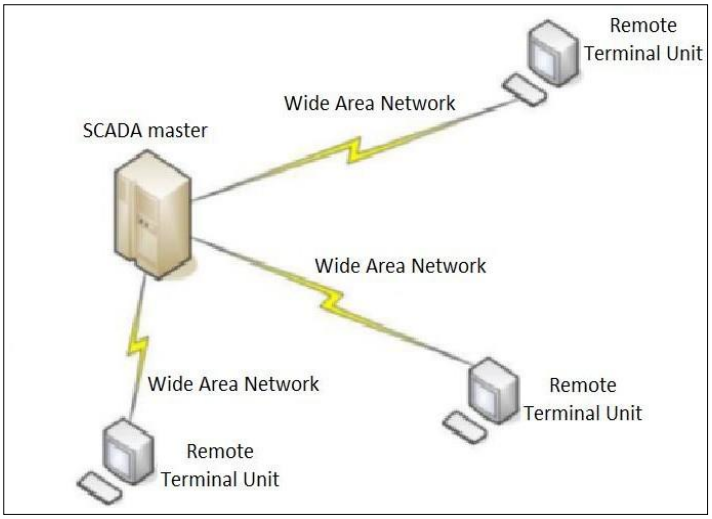
Data Acquisition

SCADA systems have evolved in parallel with the growth and sophistication of modern computing technology. The following sections will provide a description of the following three generations of SCADA systems:

1. First Generation – Monolithic
2. Second Generation – Distributed
3. Third Generation – Networked
4. Fourth generation – "Internet of Things"

Monolithic SCADA Systems

Minicomputers are used earlier for computing the SCADA systems. In earlier times, during the time of first generation, monolithic SCADA systems were developed wherein the common network services were not available. Hence, these are independent systems without having any connectivity to other systems.

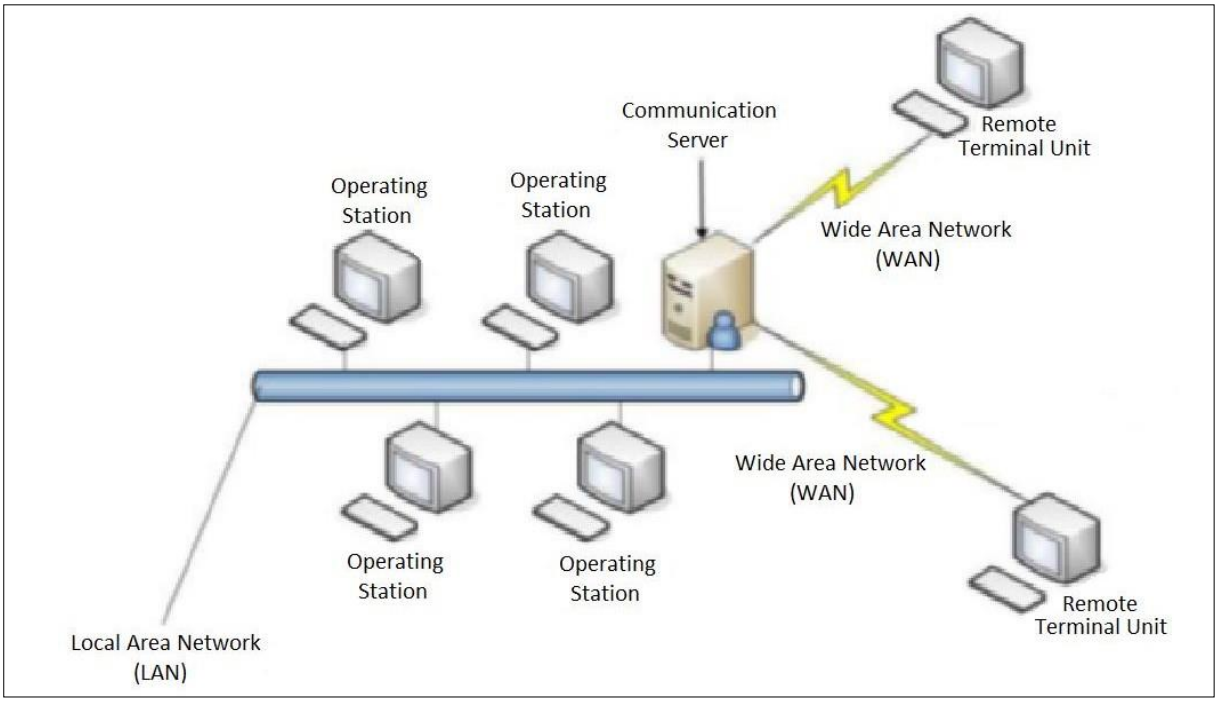


All the remote terminal unit sites would connect to a back-up mainframe system for achieving the first generation SCADA system redundancy, which was used in case

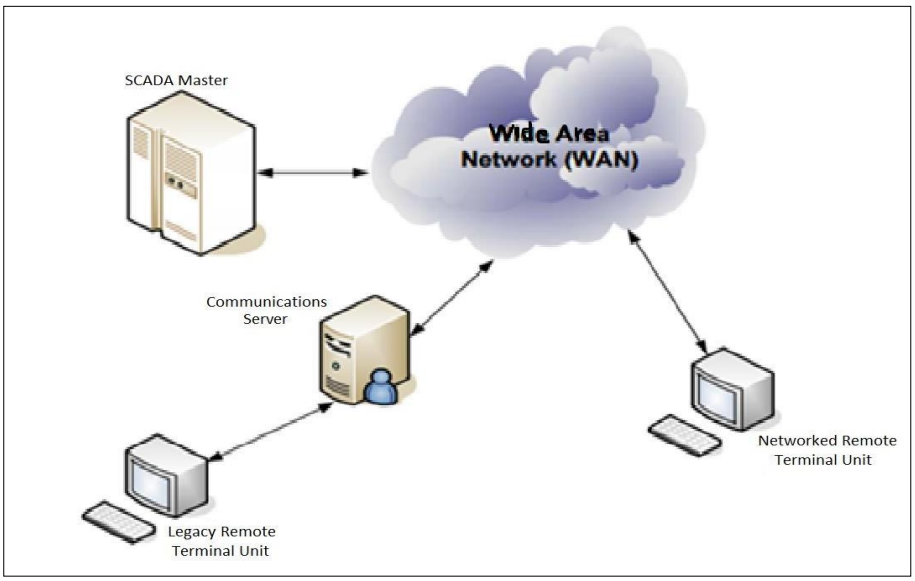
of failure of the primary mainframe system. The functions of the monolithic SCADA systems in the early first generation were limited to monitoring sensors in the system and flagging any operations in case of surpassing programmed alarm levels

Distributed SCADA System

In the second generation, the sharing of control functions is distributed across the multiple systems connected to each other using Local Area Network (LAN). Hence, these were termed as distributed SCADA systems. These individual stations were used to share real-time information and command processing for performing control tasks to trip the alarm levels of possible problems.



The cost and size of the station were reduced compared to the first generation system, as each system of the second generation was responsible for performing a particular task with reduced size and cost. But even in the second generation systems also the network protocols were not standardized. The security of the SCADA installation was determined by a very few people beyond the developers, as the protocols were proprietary. But generally the security of the SCADA installation was ignored.



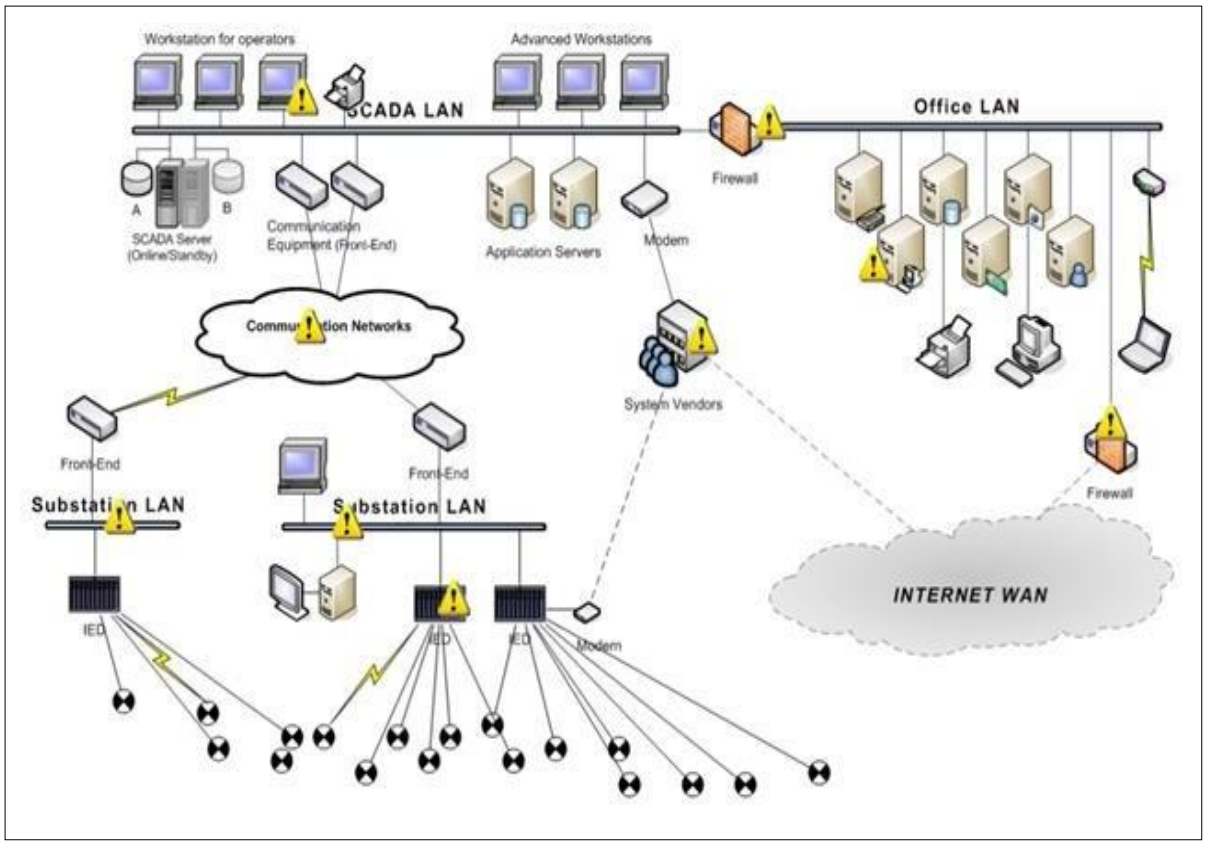
Networked SCADA Systems

The current SCADA systems are generally networked and communicate using Wide Area Network (WAN) Systems over data lines or phone. These systems use Ethernet or Fiber Optic Connections for transmitting data between the nodes frequently. These third generation SCADA systems use Programmable Logic Controllers (PLC) for monitoring and adjusting the routine flagging operators only in case of major decisions requirement.

The first and second generation SCADA systems are limited to single site networks or single building called as sealed systems. In these systems, we cannot have any risk compared to the third generation SCADA system which are connected to the internet causing the security risks. There will be several parallel working distributed SCADA systems under a single supervisor in network architecture.

Internet of Things SCADA System

In fourth generation, the infrastructure cost of the SCADA systems is reduced by adopting the internet of things technology with the commercially available cloud computing. The maintenance and integration is also very easy for the fourth generation compared to the earlier SCADA systems. These SCADA systems are able to report state in real time by using the horizontal scale from the cloud computing facility; thus, more complex control algorithms can be implemented which are practically sufficient to implement on traditional PLCs. The security risks in case of decentralized SCADA implementations such as a heterogonous mix of proprietary network protocols can be surpassed using the open network protocols such as TLS inherent in the internet of things which will provide comprehendible and manageable security boundary.



Data communication

SCADA systems have traditionally used combinations of radio and direct wired connections, although SONET/SDH is also frequently used for large systems such as railways and power stations. The remote management or monitoring function of a SCADA system is often referred to as telemetry.

Some users want SCADA data to travel over their pre-established corporate networks or to share the network with other applications. The legacy of the early low-bandwidth protocols remains, though. SCADA protocols are designed to be very compact. Many are designed to send information only when the master station polls the RTU. Typical legacy SCADAprotocols include Modbus RTU, RP-570, Profibus and Conitel. These communication protocols are all SCADA-vendor specific but are widely adopted and used. Standard protocolsare IEC 60870-5-101 or 104, IEC 61850 and DNP3. These communication protocols are standardized and recognized by all major SCADA vendors. Many of these protocols now contain extensions to operate over TCP/IP. Although the use of conventional networking specifications, such as TCP/IP, blurs the line between traditional and industrial networking, they each fulfill fundamentally differing requirements.

With increasing security demands (such as North American Electric Reliability Corporation (NERC) and Critical Infrastructure Protection (CIP) in the US), there is increasing use of satellite-based communication. This has the key advantages that the infrastructure can be self-contained (not using circuits from the public telephone system), can have built-in encryption,

and can be engineered to the availability and reliability required by the SCADA system operator. Earlier experiences using consumer-grade VSAT were poor. Modern carrier-class systems provide the quality of service required for SCADA.

RTU and other automatic controller devices were developed before the advent of industry wide standards for interoperability. The result is that developers and their management created a multitude of control protocols. Among the larger vendors, there was also the incentive to create their own protocol to "lock in" their customer base. A list of automation protocols is compiled here.

Recently, OLE for process control (OPC) has become a widely accepted solution for intercommunicating different hardware and software, allowing communication even between devices originally not intended to be part of an industrial network.

Data presentation

The SCADA system usually presents the information to the operating personnel graphically, in the form of a mimic diagram. This means that the operator can see a schematic representation of the plant being controlled. For example, a picture of a pump connected to a pipe can show the operator that the pump is running and how much fluid it is pumping through the pipe at the moment. The operator can then switch the pump off. The HMI software will show the flow rate of the fluid in the pipe decrease in real time. Mimic diagrams may consist of line graphics and schematic symbols to represent process elements, or may consist of digital photographs of the process equipment overlain with animated symbols.

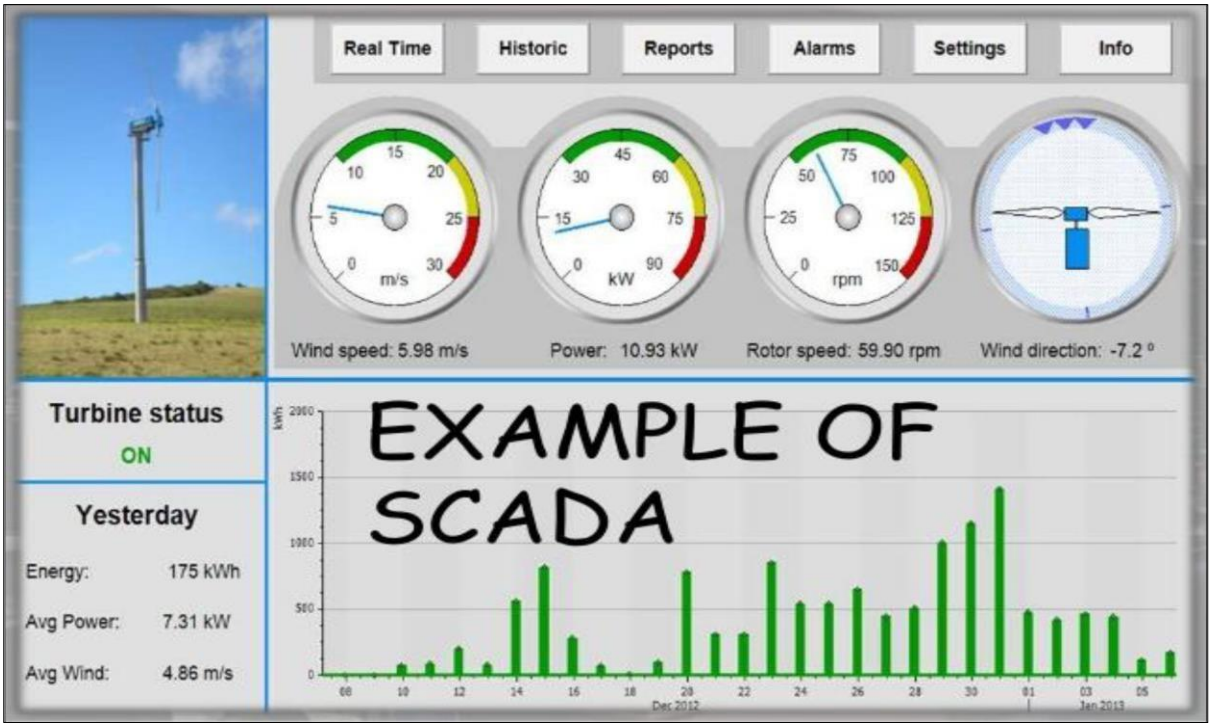
The HMI package for the SCADA system typically includes a drawing program that the operators or system maintenance personnel use to change the way these points are represented in the interface. These representations can be as simple as an on-screen traffic light, which represents the state of an actual traffic light in the field, or as complex as a multi-projector display representing the position of all of the elevators in a skyscraper or all of the trains on a railway.

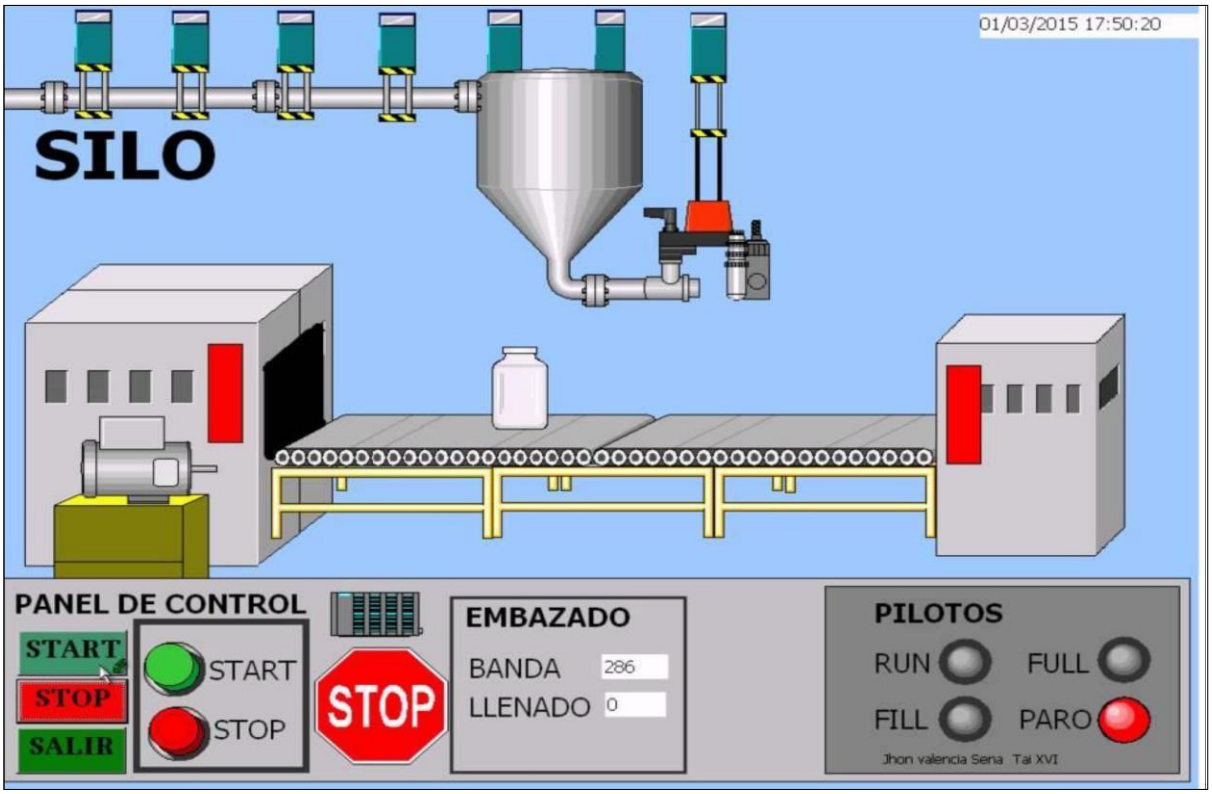
Difference between PLC and PC

S.r. no:	PLC	Computer
1.	Designed for extreme industrial environments	Designed mainly for data processing and calculation.
2.	Can operation in high temperature and Humidity	Optimized for speed
3.	High immunity to notice	Can’t operate in extreme environments
4.	Integrated command interpreter(proprietary)	Can be programmed in different languages
5.	No secondary memory available (in the PLC.)	Lots of secondary memory available
6.	Optimized for single task	Multitasking capability

Examples of SCADA

With the help of SACDA we can visualize real time processes running at the site some example photographs are shown here:





And many more…

System Overview

WinCC Overview

SIMATIC WinCC is a scalable process visualization system with powerful functions for monitoring automated processes. WinCC provides complete SCADA functionality under. Windows for all sectors – from single-user to distributed multi-user systems with redundant servers and cross location solutions with web clients.

This section provides an overview of the engineering framework "TIA Portal" and focuses on the options of configuring PC-based SCADA systems (control systems). Participants will get to know the options available for configuring user interfaces. They will get an overview of the software, supported devices and required licenses.

TIA Portal (Totally Integrated Automation)

This is a working environment central engineering framework to achieve total integration of Independent software packages throughout the system.

- Uniform operator control concept for all automation tasks with common services (For example configuration, communication and diagnostics).
- Automatic data and project consistency
- Powerful libraries covering all automation objects

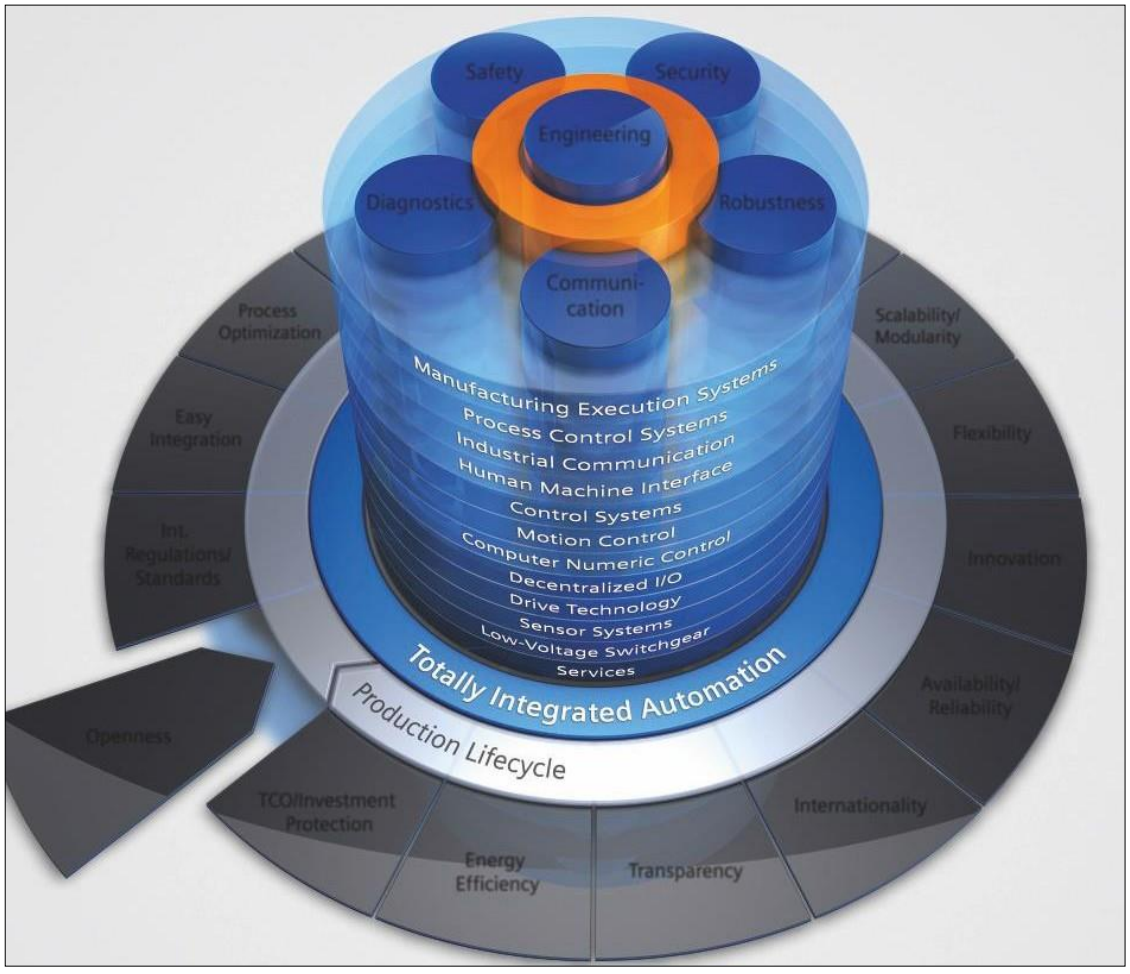
To be able to respond to the increasing international competitive pressure, it is more important than ever to consistently make full use of the potential for optimization over the complete lifecycle of a machine or Plant.

Optimized processes reduce the total cost of ownership, shorten the time to market, and improve quality. This perfect balance between quality, time, and costs is now, more than ever, the decisive success factor in industry.

Benefits of TIA Portal

Centralized engineering for all tasks in the automation system. Thanks to the identical Representation, working in different editors presents no problem.

- All editors in the TIA Portal have an identical basic layout
- Overview of the entire project structure
- Features and information are always at the same place
- The work environment is easy to customize
- One database and the cross-reference list provide a project-wide overview of where tags and Objects are used.



Totally Integrated Automation is optimized for all requirements and is open for international standards and third-party systems. With its six characteristic system characteristics engineering, communications, diagnostics, safety, security, and ruggedness), Totally Integrated Automation supports the complete lifecycle of a machine or plant.

The complete system architecture offers holistic solutions for every automation segment on the basis of a comprehensive range of products.

WinCC system software

The WinCC system software is available in two basic variants:

- WinCC complete package (RC: license for runtime and configuring)
- WinCC runtime package (RT: runtime license)

Both packages are available with 128, 512, 2k, 8k, 64k, 100k, 150k or 256k Power Tags. Only process tags that are connected to the controller or other data sources via a WinCC communication channel are designated as Power Tags. Up to 32 messages and up to 256 user definable analog alarms can be derived from one process tag. Moreover, internal tags without process connection are available for free to provide additional system performance. Power Packs allow an increase in the number of usable Power- Tags. Thus, WinCC also grows when your application grows. You can start with the smallest variant and then upgrade as needed with attractive Power Packs. With Power Packs, the number of available archive tags can also be increased, from 512 (contained in the basic scope) by 1,500, 5,000, 10,000, 30,000 or 80,000 to up to 120,000.

WinCC multi-user system

Automation and IT solutions are generally subjected to constant change. This includes, for example, modernizations of individual sections of the plant, later plant expansions, the implementation of central monitoring of various locations in a company, and the optimization of the process sequences at a location or in a company. To be able to meet growing requirements, the visualization must be expandable at any time without causing technology incompatibilities or requiring completely new configurations. Investment protection is a top priority. SIMATIC WinCC provides the required integrated scalability, from the small single user solution to the client/server solution with a central Microsoft SQL server for data archiving and operator stations on the web. Virtualization increases flexibility and offers a significant potential for savings in system administration and maintenance. The possibility of using simple, robust thin clients provides additional cost benefits and increases IT security.

From a single-user system to distributed client/server solution

Scalable means that the number of tags in your project can be upgraded as needed with Power Packs – without paying more, as if you had chosen the larger solution in the first place. From single-user systems, you can create coordinated operator stations at any time by means of the server option – i.e. a multi-user system. In all, up to 12 redundant WinCC servers with 32 WinCC clients can be used in a linked system configuration. The servers themselves can be configured as a distributed system. The distribution of the overall application or of the tasks to several servers allows a considerably higher quantity structure, relieves the individual servers and ensures good performance. Upon request, an optional central archive server, for example, can be set up for central process data archiving. The distribution also allows for the topology

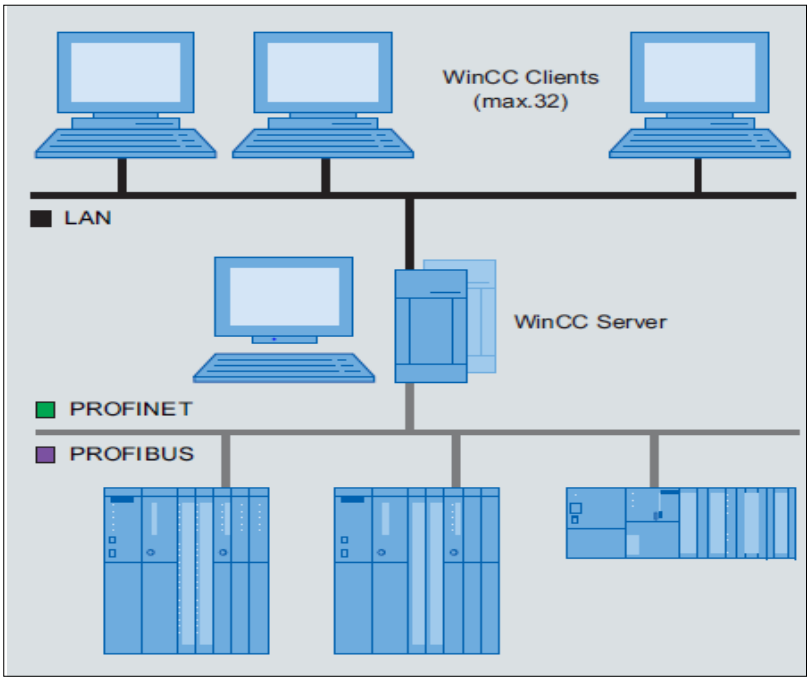
of a plant. The overall view of the plant is provided by WinCC Clients, which have simultaneous access to or viewing of screens and data of different server projects and which can also be used for online configuration. For these clients, a shared message view and trend view of archives from different servers can be configured.

Limitless operator control and monitoring

WinCC can also be expanded beyond the local network (LAN) of a site, whether in order to be able to sporadically and remotely monitor a section of a plant (e.g. in wastewater treatment plants or in station control systems) or on order to be able to access current or historical process information (e.g. for statistical analysis) from any location.

WinCC multi-user system

The WinCC/Web Navigator supports full-fledged operator control and monitoring via the web generally without requiring changes to the project. A WinCC web server can be installed on a WinCC single-user system, a WinCC server or client. Thus a web client connected to the web server has access to the projects of all (up to 12) WinCC servers in a plant from anywhere in the world. The user administration of the operator stations on the web is Included in the database of the plant on-site. Different authorizationlevels regulate the access rights. In addition to this, support of standard security mechanisms for operating on the Internet is provided. Central, plant-wide user administration, into which web clients are also integrated, can be set up using SIMATIC Logon. In addition, the use of terminal services and thin clients is also possible by means of the WinCC Web Navigator option. Via thin client solutions, rugged on-site devices (e.g. SIMATIC Thin Client) and mobile clients (PDA – Personal Digital Assistant) can be connected under Windows CE with few hardware requirements. The application itself runs on the Terminal Server.



Components of WinCC

WinCC consists of these components that can be expanded with options.

Engineering system

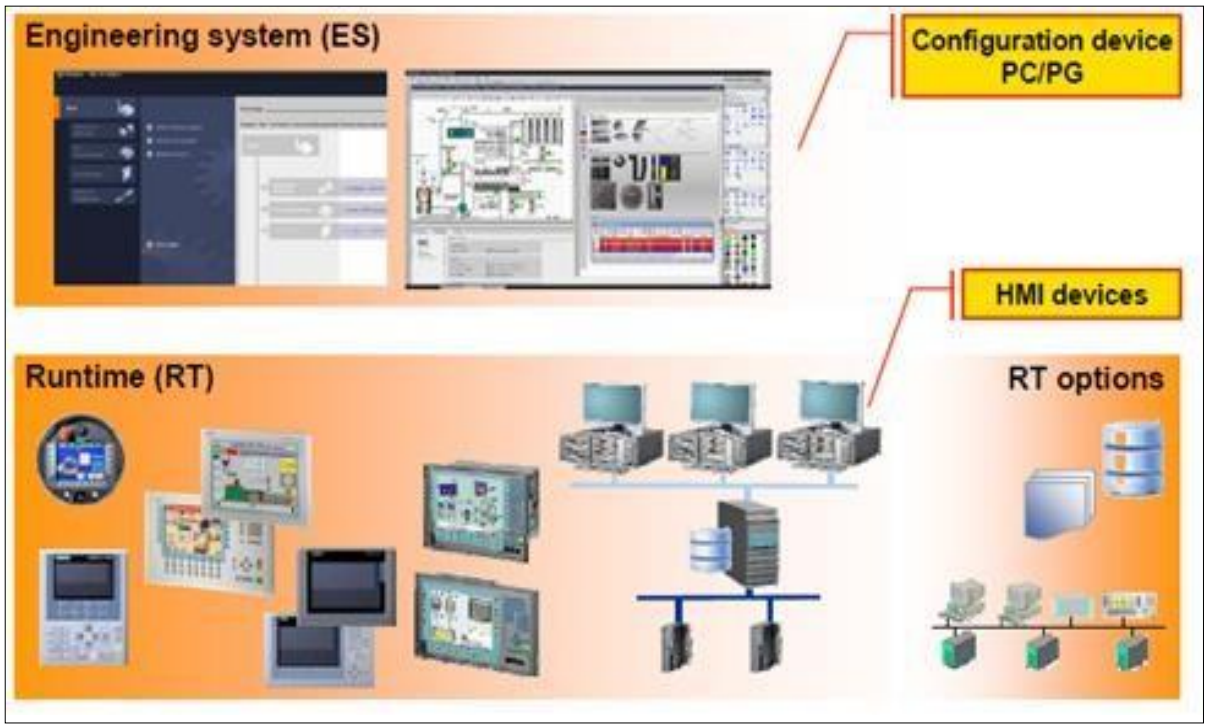
The engineering system is the software with which you can carry out all the configuration tasks required to create a user interface for operator control and monitoring of machines and plants.

Runtime

WinCC Runtime is the software for process visualization on the HMI device. With Runtime you run the project in process mode.

RT options

Runtime can be expanded by optional functions (licenses).The Runtime options depend on the target system used. If an HMI device does not support a Particular optional functionality, the option cannot be configured. Not all the options available today are available yet in the TIA Portal. These are planned for the next service packs.



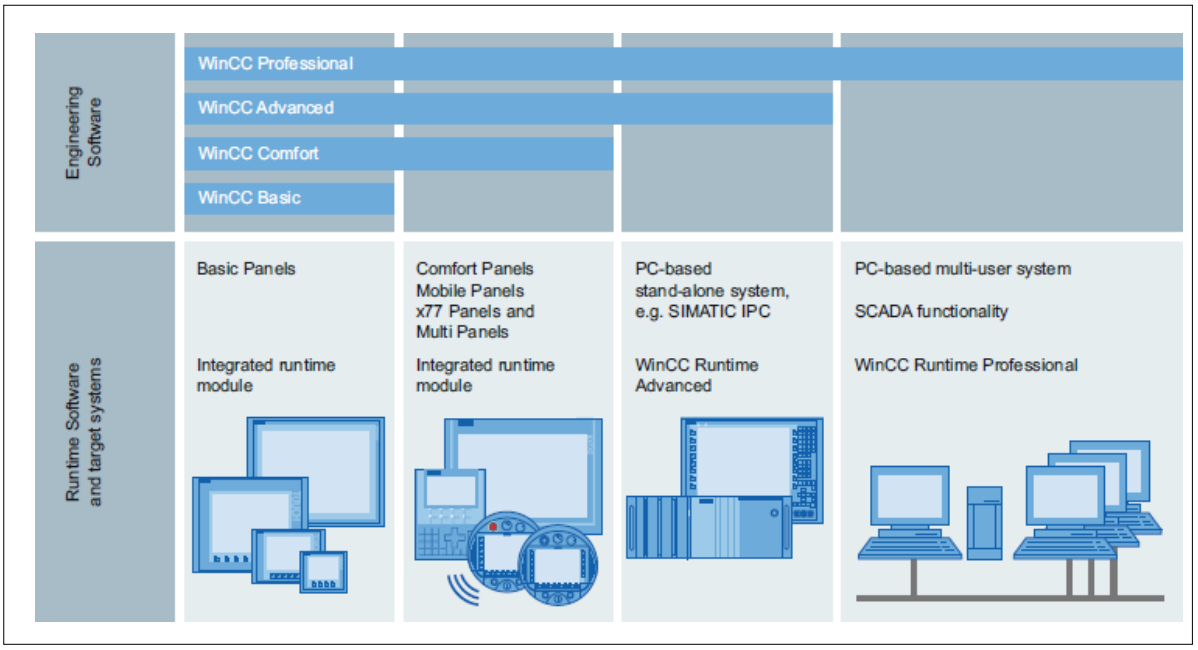
Engineering system Editions

The engineering system of WinCC has a modular design, graded in different editions. The edition decides which operator panels in the SIMATIC HMI range can be configured. With each higher edition, you expand the range of supported target devices and functionalities.

The configuration data is compatible.

- WinCC Basic
  - Configuration of the Basic Panels (KPxxxx Basic)
- WinCC Comfort
  - Configuration of the Basic Panels (KPxxxx Basic)
  - Configuration of the panels / Multi Panels (OP73, opxx7, tpxx7, mpxx7)
  - Configuration of the Mobile Panels (Mobile Panel xx7)
  - Configuration of the Comfort Panels (KTPxxx Comfort, TPxxxx Comfort, KPxxx Comfort)
- WinCC Advanced
  - Configuration of the Basic Panels (KPxxxx Basic)
  - Configuration of the panels / Multi Panels (OP73, opxx7, tpxx7, mpxx7)

  - Configuration of the Mobile Panels (Mobile Panel xx7)
  - Configuration of the Comfort Panels (KTPxxx Comfort, TPxxxx Comfort, KPxxx Comfort)
  - Configuration of the PC-based single user systems (HMI IPCxxxc, standard pcs)
- WinCC Professional
  - Configuration of all Basic Panels and PC-based single user systems
  - Configuration of PC-based SCADA systems (control systems) SCADA = Supervisory Control and Data Acquisition normally networked multiple user systems.



Comparison of engineering system edition

The SIMATIC Panels and PC-based HMI solutions with WinCC Runtime Advanced and WinCC Runtime Professional contain all of the important functions for operating and monitoring machines or plants. For an expanded task range, the functionality can be supplemented by means of additional options. In addition to the Runtime options, WinCC Runtime Advanced and WinCC Runtime Professional can be expanded by means of customer-specific controls. The WinCC Control Development option is required for the development of the controls.

Option	Panel Runtime (device-dependent)	WinCC Runtime Advanced	WinCC Runtime Professional
WinCC Logging	●	○	●
WinCC Recipes	●	○	○
WinCC Audit	○	○	–
SIMATIC Logon	○	○	●
Smart Server	○	○	–
WinCC Client	–	–	○

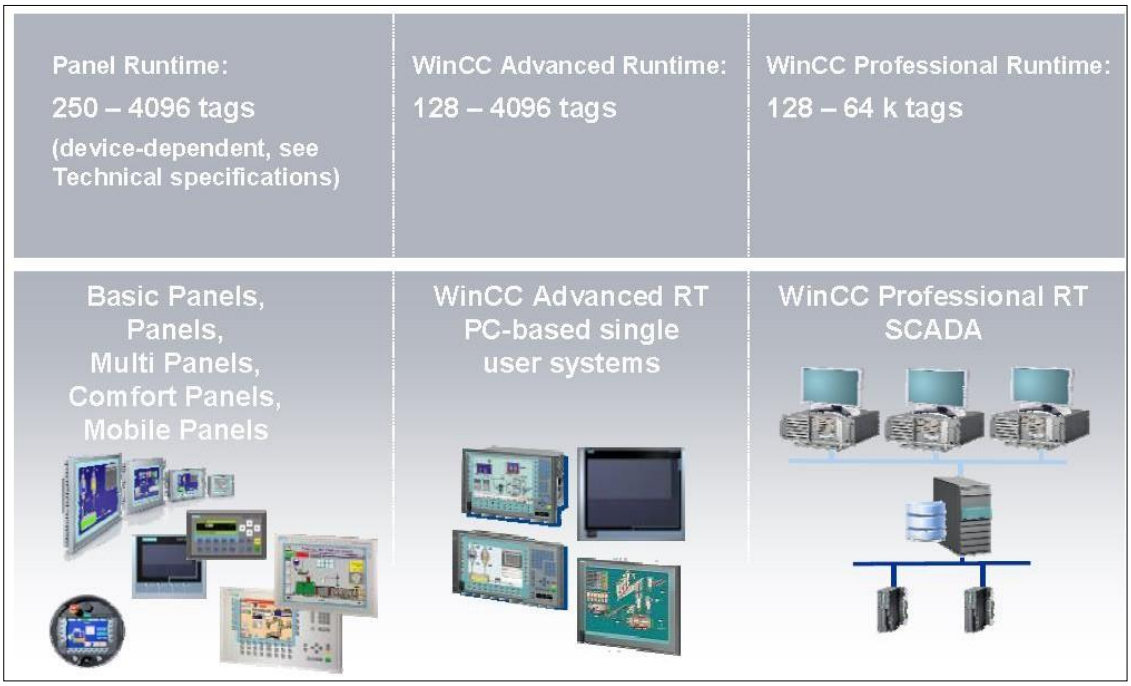
WinCC Server	–	–	○
Web Navigator	–	–	○
Data Monitor	–	–	○

- Included ○ optionally available

Operating systems

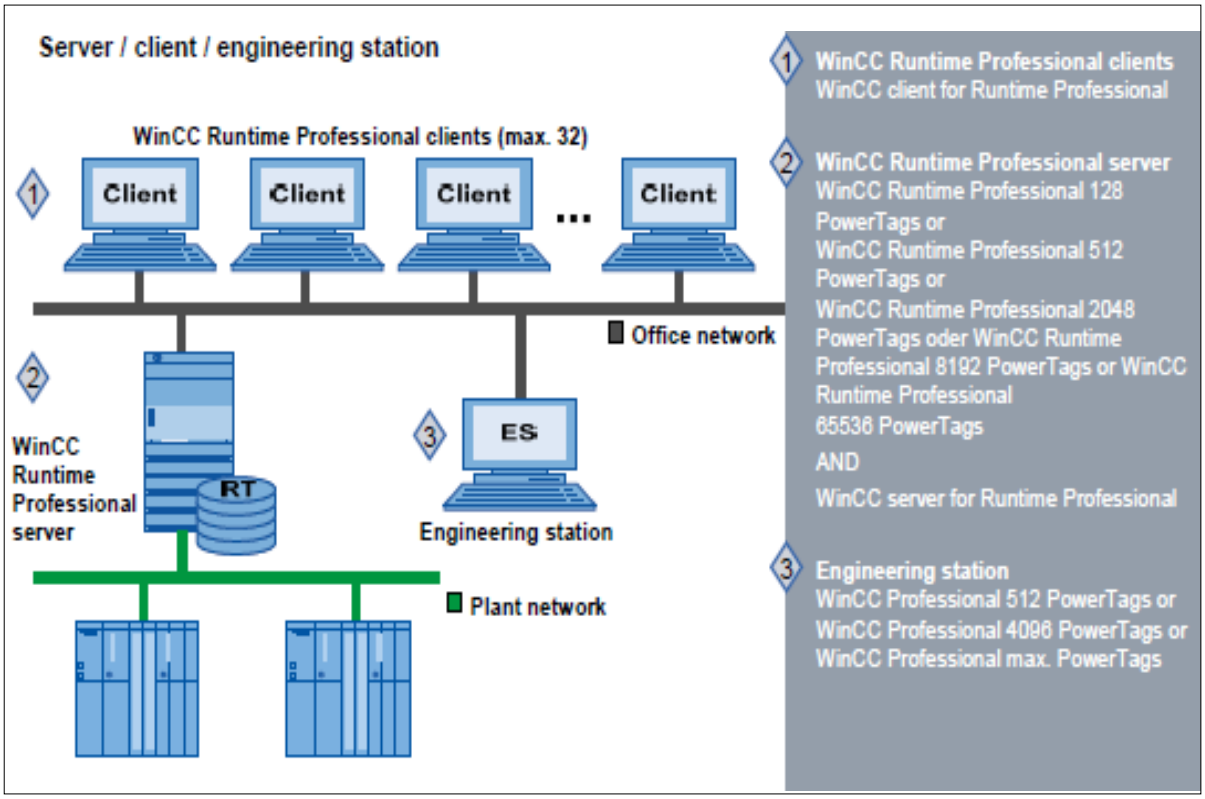
WinCC in the TIA Portal – Engineering software	
Supported Operating systems	• Windows XP Professional SP3 (32 bit) • Windows 7 Professional/Enterprise/Ultimate (32/64 bit) • Windows 7 Professional/Enterprise/Ultimate SP1 (32/64 bit)
	Additionally with WinCC Basic: • Windows XP Home SP3, • Windows 7 Home Premium (32/64 bit) • Windows 7 Home Premium SP1 (32/64 bit
	Additionally with WinCC Professional: • Windows Server 2003 Release 2 Standard Edition SP2 (32 bit) • Windows Server 2008 Standard Edition SP2 (32 bit) • Windows Server 2008 Release 2 Standard Edition (64 bit) • Windows Server 2008 Release 2 Standard Edition SP1 (64 bit)
WinCC in the TIA Portal – Runtime software	
Supported Operating systems	• Windows XP Professional SP3 (32 bit) • Windows Embedded Standard 2009
	• Windows 7 Professional/Enterprise/Ultimate (32/64 bit) • Windows 7 Professional/Enterprise/Ultimate SP1 (32/64 bit) • Windows Embedded Standard 7 (32 bit), • Windows Embedded Standard 7 SP1 (32 bit),
	• Windows Server 2003 Release 2 Standard Edition SP2 (32 bit) • Windows Server 2008 Standard Edition SP2 (32 bit) • Windows Server 2008 Release 2 Standard Edition (64 bit) • Windows Server 2008 Release 2 Standard Edition SP1 (64 bit)
	Windows Embedded operating systems only on specifically released platforms (e.g. Panel PC 477), contact your Siemens representative for information.

Runtime versions



Power Tag

A created process variable only becomes a power Tag when used in a configured display (I/O fields, parameters of graphic objects, trends etc.) or for logging process values.



Changing versions WinCC Runtime Advanced / WinCC Professional

The Runtime version must be upgraded when the maximum number of Power Tags has been exceeded. The existing configuration does not need to be adapted. You can change to a higher Runtime version with a "**Power pack".**

Licensing of engineering editions, Runtime versions and options

What is a license?

A license is a key that is supplied on a read-only USB stick. This license key releases a particular software component for use.

License model

All WinCC software requires a license. When purchasing a software package, a specific license contract is obtained in paper form. The license entitles the user to install and use the purchased software on one PC or HMI device. With the exception of WinCC Basic, a USB licensing stick belongs to each license contract. This USB licensing stick contains the corresponding license key. If client/server concepts or a service/diagnostics concept (Smart Server) are installed and configured, a license is required for each HMI device involved.

Licensing of software components

Utility: Automation License Manager

This license key must be transferred from the licensing diskette to the target device (engineering PG/PC or HMI device) using a utility program. This utility is part of the **engineering software**: For the procedure and more information, refer to the documentation of the utility.

Without licensing

When replacing a device (e.g. Because of a fault), it is possible that a suitable license is not immediately available.

Runtime - licensing principle

Without the required license, the use of the WinCC software is restricted: Messages (regarding the missing license) appear frequently on the screen/display and must be acknowledged. The HMI device nevertheless remains functional.

Engineering - licensing principle

When the engineering system is started, the automatic activation of a 14-day trial (time) is Offered. The Engineering software has an emergency licensing function that can be activated. This is intended to cover servicing periods.

Validity: **14 days!**

Restoring a license

Request a new license from support.

System requirements: hardware

For each system, Basic requirements are 2 GB RAM, Core 2 duo processor, 1280 ×1024 graphics minimum.

Detailed specifications are described below.

	Processor(Min)	**RAM**
Windows XP SP3	2.5 GHz P4 or comparable	2 GB
Windows 7	3.5 GHz P4 or comparable, Dual Core	2 GB
Windows Server 2003 SP2	3 GHz P4 or comparable	2 GB
Windows Server 2008 SP2	3 GHz P4 or comparable, Dual/Multi Core	2 GB

System limits

WinCC V11 Professional for medium sized projects

	WinCC V7.0 SP2	**WinCC V11 Prof.**
Tags	265 K	64K(1)
Archive Tags	80,000	5,000
Screens/ screen objects	Unlimited/unlimited	1,000(2)/3,000
Messages	150,000	20,000
Recipes/ recipe elements	Unlimited/3,000	1,000/3,000

(1)Max. 80,000 Tags per project (Incl. logging Tags)
(2)Max 1,000 screens per HMI devices, max 3,000 screens per project

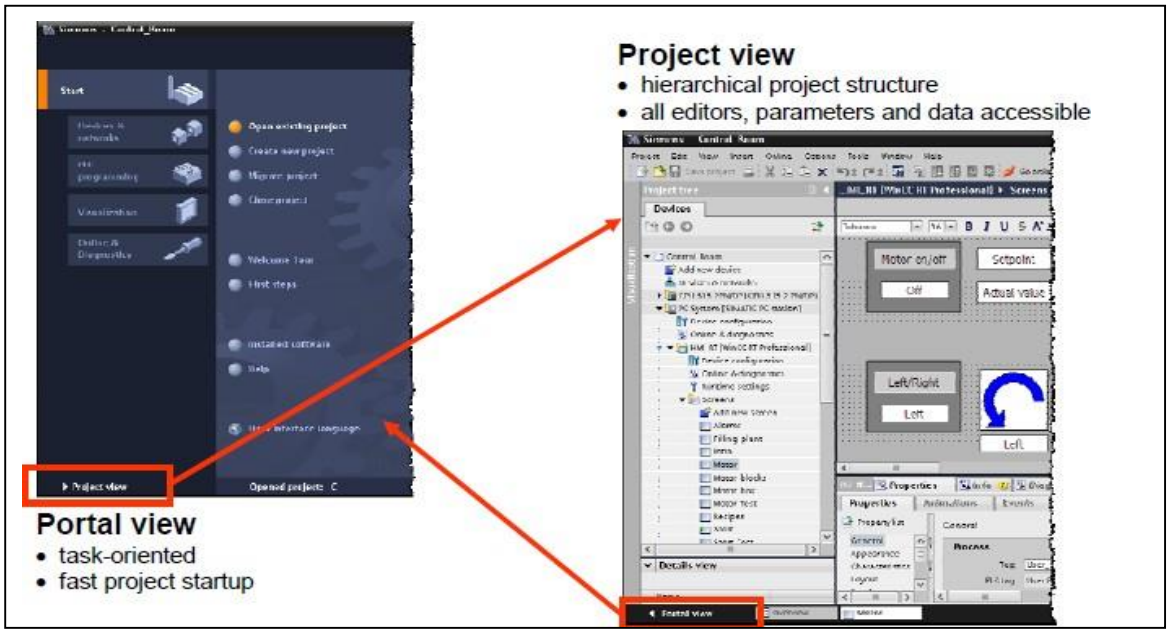
Configuration Interface of Engineering System

View of the engineering system

The TIA Portal framework manages the project (a database) for all components and project data of an automation solution.

It provides two views between which users can change at any time.

1. Portal view
2. Project view

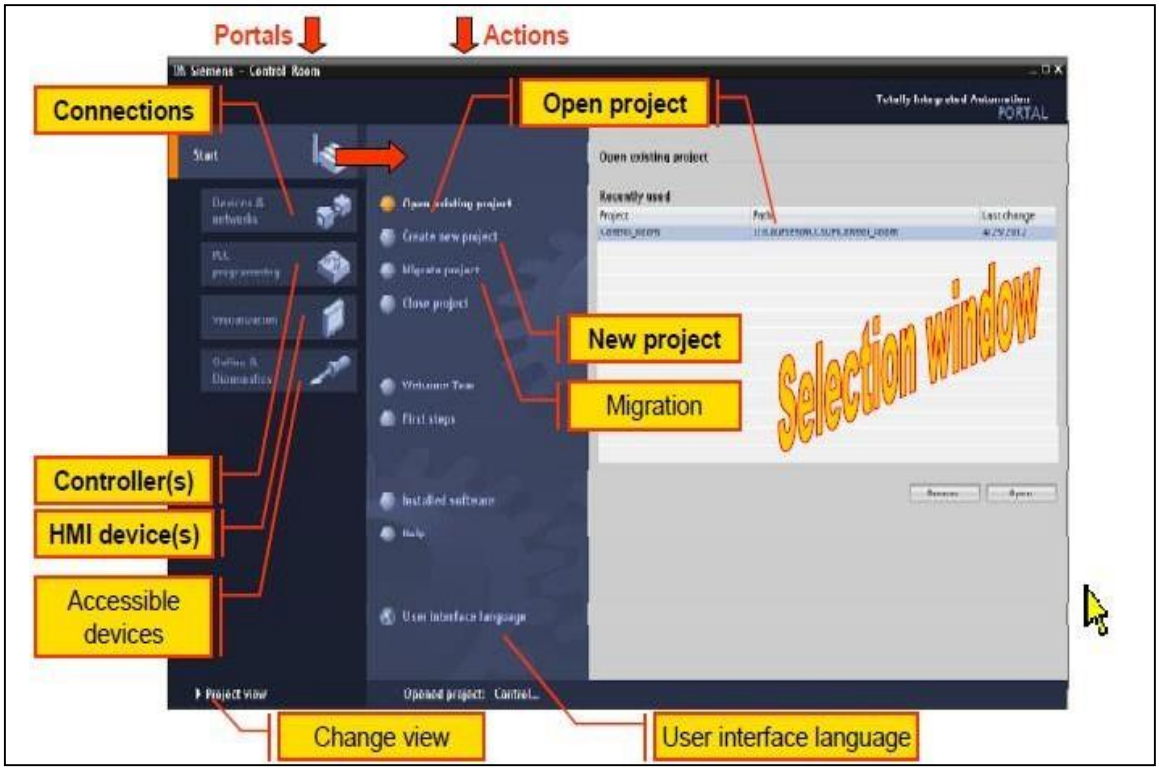


Following software installation, the TIA Portal engineering user interface opens with the basic Setting → **Portal view**, Start portal.

Project & Portal view understandings Project tree, task card & inspector window opening & working with an editor

1. Portal view

- Task-oriented mode of working
- Fast project start-up with user guidance



2. Project view

- Hierarchic structure of the project
- All editors, parameters and data are in one view
- The necessary editors open according to the task in hand.

Layout of the portal view

Portals (left) for the various tasks

- Actions (central area) for the portal selected on the left
- Selection window (right) for the selected action

Portals

Access to devices, components and their connections.

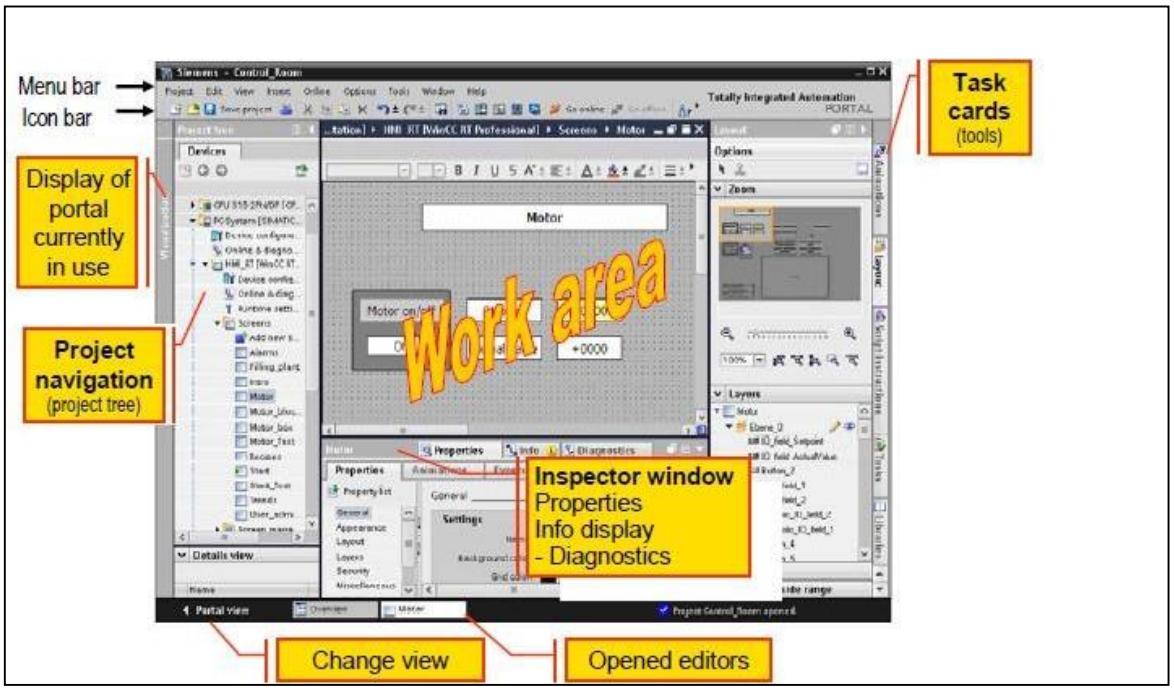
Actions

Depending on the selected portal, actions are available here that can be executed in the selected portal. Context-sensitive help is available in every portal.

Selection window

The selection window is available in all portals. The content of the window adapts to the currently selected portal and action.

Project view:



Project tree

The project tree is the central control point for the automation solution. All components and editors available in a project appear in the Project tree and can be opened there. Each editor is assigned a symbol with which the objects belonging to it can be identified. Only elements supported by the selected HMI device are displayed in the Project tree.

Work area

The central configuration area in which the objects of the HMI device can be edited using the active editor. Several editors can be opened simultaneously. If more than one editor is open at one time, these are displayed as tabs at the bottom.

Task cards

These provide tools for configuration/programming. The content of the task cards depends on the open editor and the selected object displayed in the work area. If a screen is open, configuration objects, templates for animation, layout tools and libraries, for example, are available. If a CPU is opened as the device, the hardware catalog, for example, is available as a task card. If a program block is opened, a task card with instructions is available.

Inspector window

Additional information on a selected object or on executed actions is displayed in the Inspector Window. The available properties of the selected objects can also be edited here (for The Inspector window displays all system messages from the engineering, for example those resulting from generating a project. On completion of generation, this window should always be checked for any errors and warnings.

Menu call:

> View > Inspector window

Details view

There is a help window in the Details view. Here, the elements of the configuration editor selected in the project window are displayed. These can be used in the active editor of the work area (by dragging them to the work area). This allows fast access to the required objects (for example tags).

Project tree and task cards

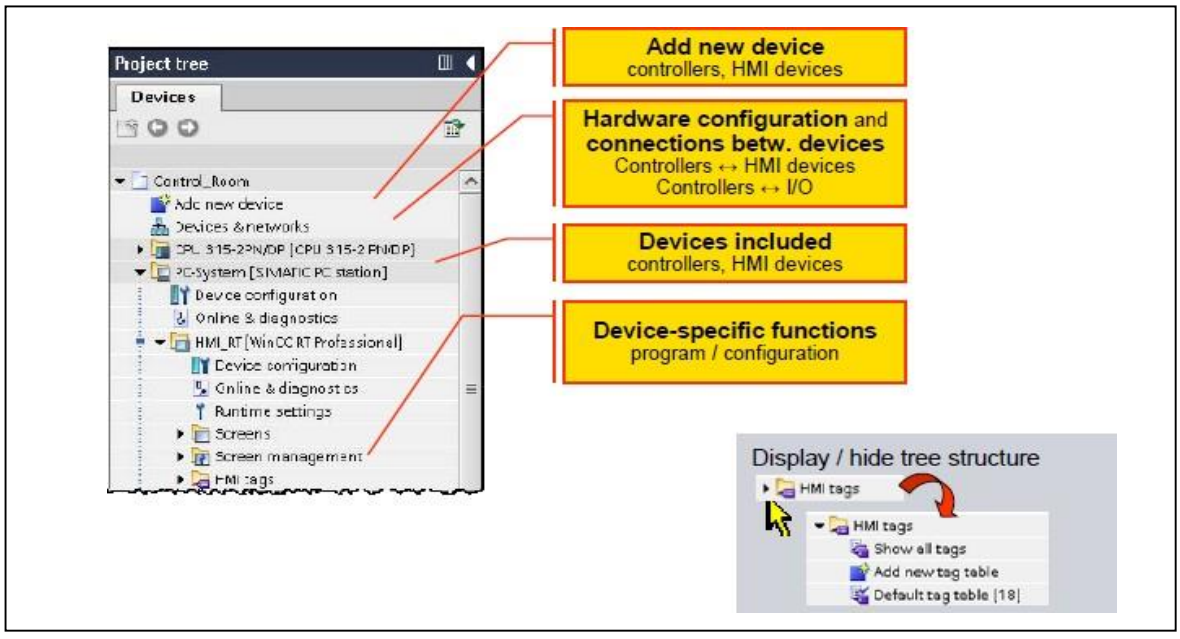
In terms of the range of functions, the contents of the project tree and the task cards adapt themselves during configuration to the HMI device currently being used. A different range of functionality is available in the task cards depending on the HMI device. This means that only functions supported by the selected HMI device can be configured.

Project tree

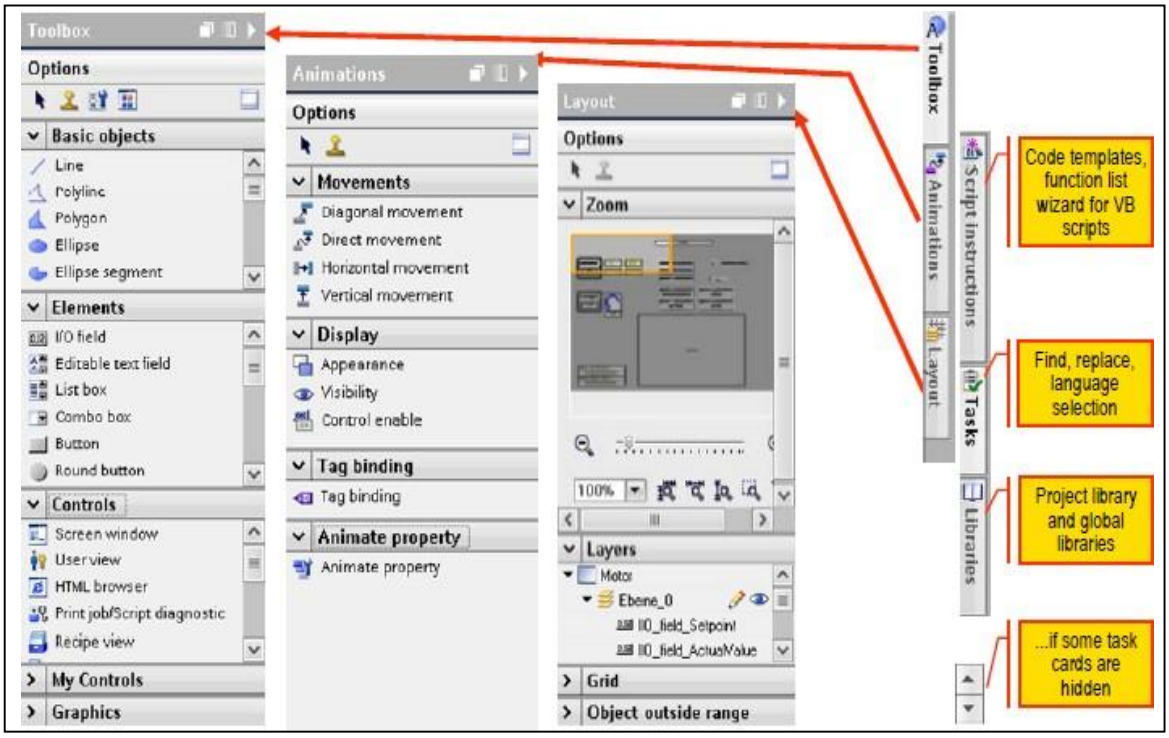
The Project tree provides access to all components and project data. All components and editors available in a project appear in the Project tree and can be opened there. Each editor is assigned a symbol with which the objects belonging to it can be identified. Only elements supported by the selected HMI device are displayed in the Project tree.

The following actions are available:

- Adding configurable or programmable components (controllers, HMI devices etc.)
- Editing existing components
- Querying and modifying properties of existing components
- Diagnosing accessible components



Task card



Depending on the opened editor, suitable task cards (additional windows) are available at the right-hand edge of the screen. Which task cards are available depends on the products that have been installed and the currently active editor. If they cannot all be seen, the task card bar can be moved to the bottom Right in the framework using the cursor buttons.

Screen editor

- Toolbox

Configurable screen objects (graphics, display and operator control objects) in different panes (basic objects, elements, controls, optional customized controls, graphics)

- Animations

Templates for dynamism of screen objects in different panes (movements, display, tag binding for dynamism). The "**Animations**" task card is not available in WinCC Basic.

- Layout

Tools for adapting the presentation in the editor (zoom, level assignment, grid alignment, objects outside the area)

Inspector window

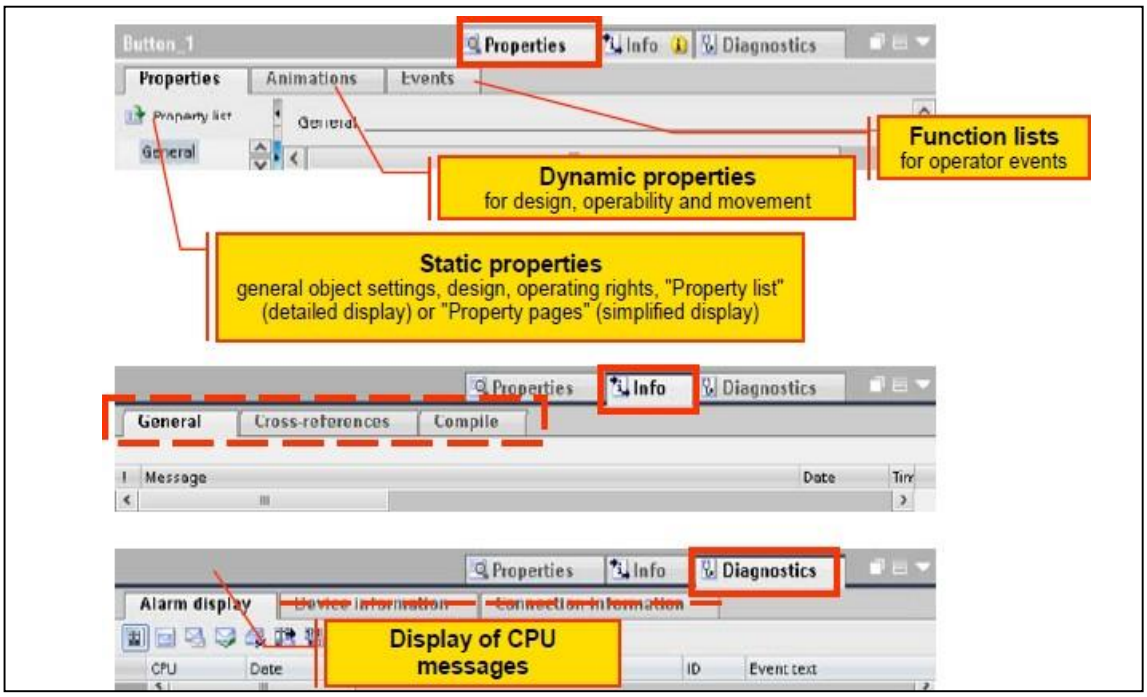
The Inspector window is divided into three information areas with a maximum of three underlying Areas each can be selected by clicking on the tabs

- "Properties" area

In this area, the properties of one or more objects selected in the work area are adapted. The contents of the area depend on the type of object. Each object has the "Properties" area. On the other hand, "**Animations**" and "**Events**" do not exist with all objects.

- Properties

Object settings and static properties are specified during configuration and assigned to the objects. The static properties always have a default value.



- "Info" area

Here you select the output area of the engineering.

"**General**" tab → general status output

"**Cross-references**" tab → display of the current locations at which the selected object is used

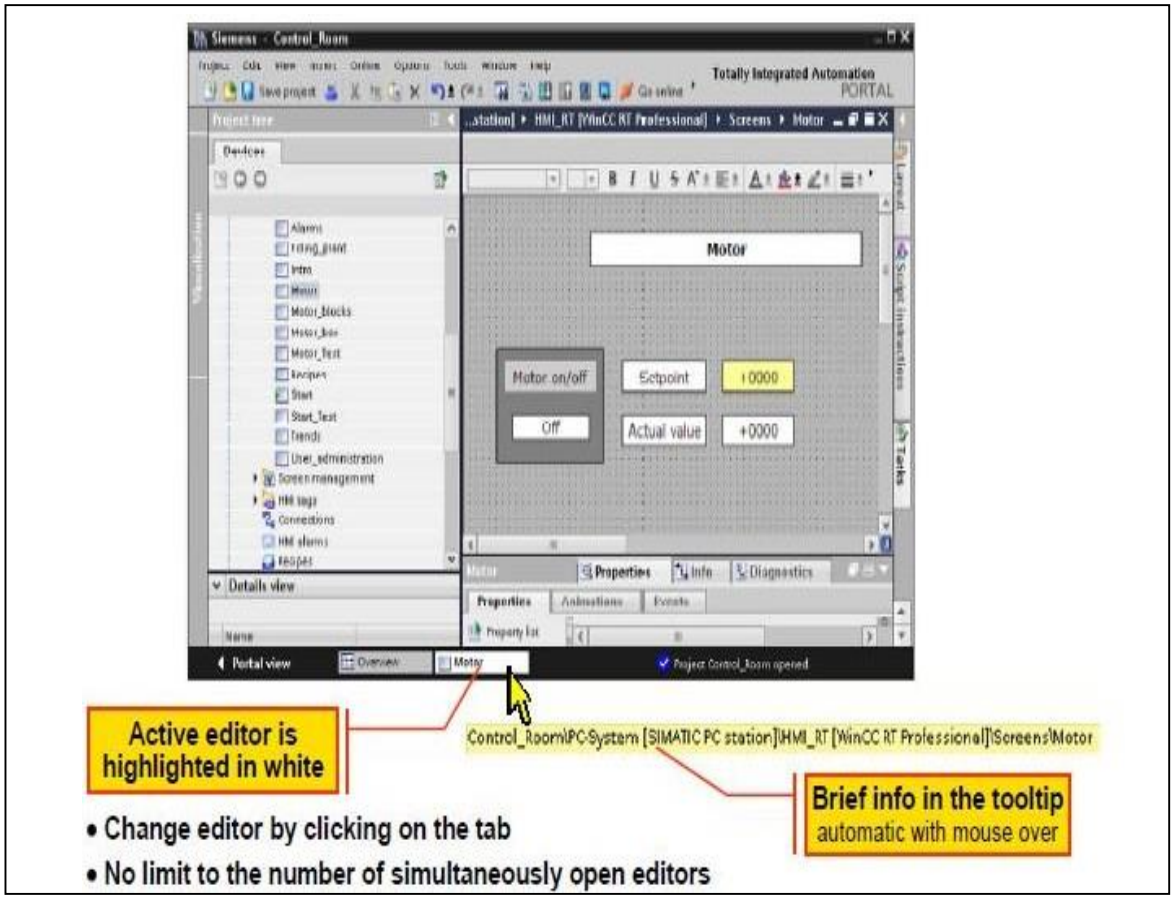
"**Compile**" tab → status display of compilation progress

- "Diagnostics" area

Here, the output area for diagnostics information is opened.

"**Alarm display**" tab → display of current CPU alarms (for example Alarms)

"**Device information**" tab and "**Connection information**" tab → only relevant for CPUs and similar components.



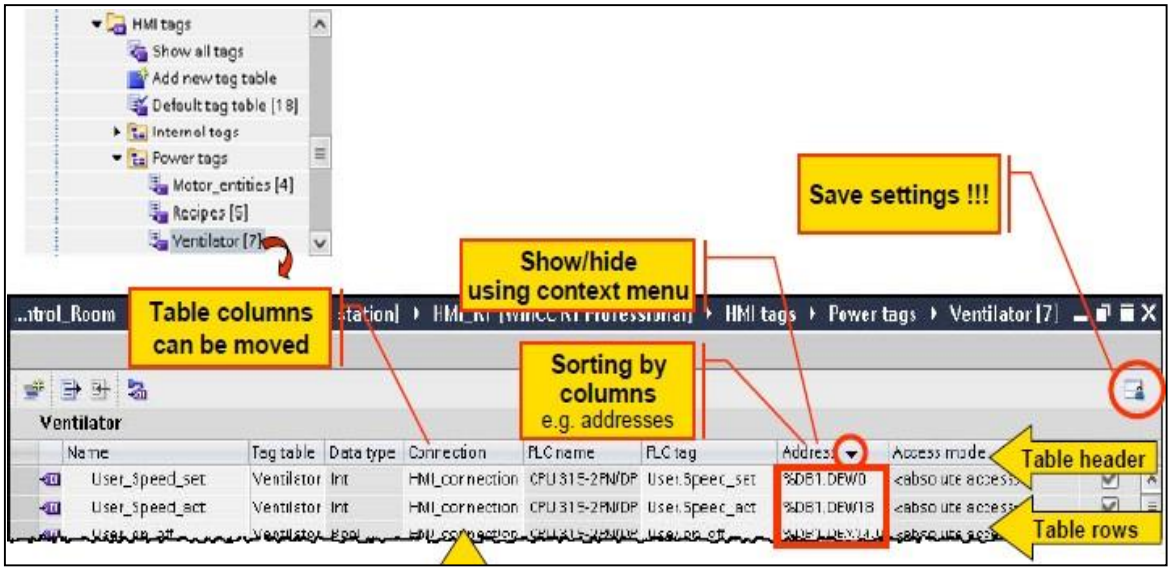
- Working with the editors

WinCC has a special editor for each configuration task. The currently opened editors are

Displayed along the bottom edge of the framework. A distinction is made between graphic editors (Screen editor) and table editors (e.g. Tag editor).

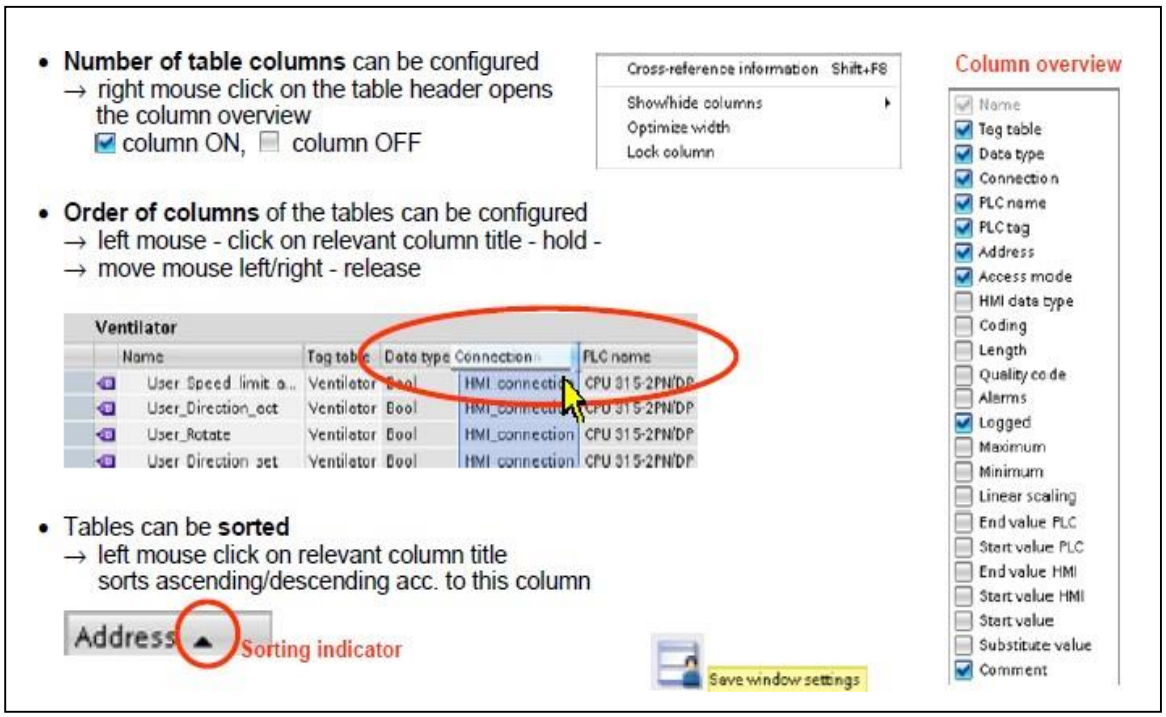
Table editor

The table editor provides useful support for fast and efficient configuration of table objects such as tags.

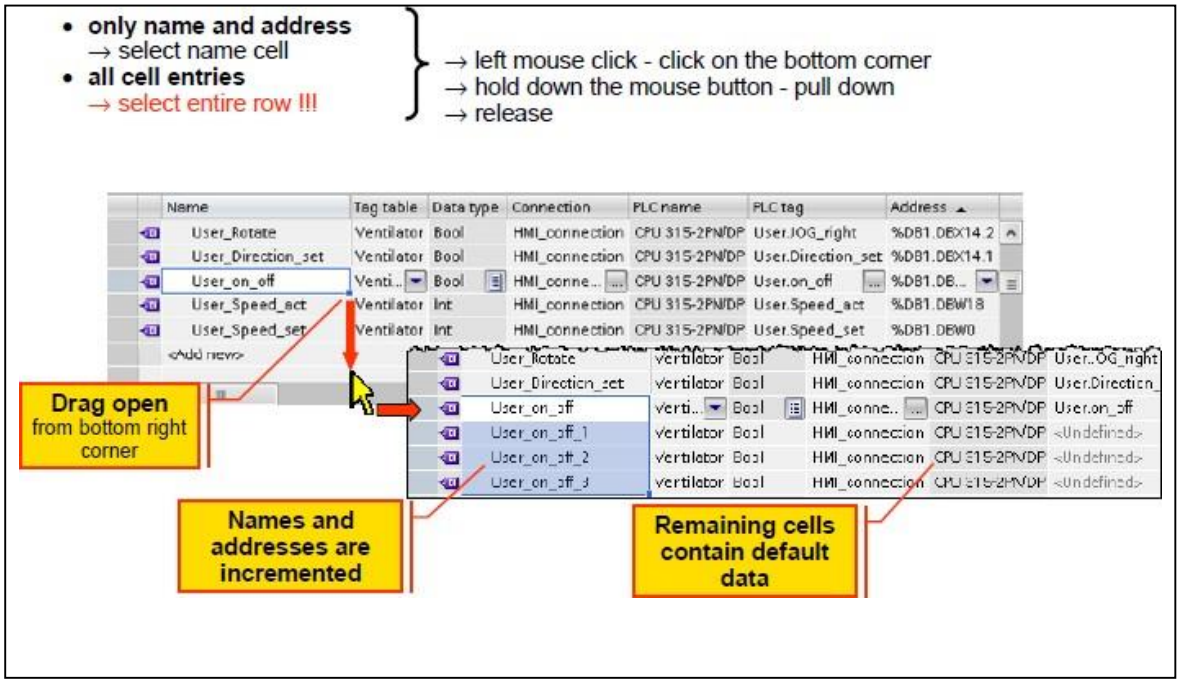


- **Design of a table editor**

The display columns, their sequence, and sorting of their contents in each table can be individually configured. The individual arrangement can be saved (in the Windows user profile) using the "Save window Settings" button (at the top right in the editor). Otherwise the adapted setting is lost again when you close the editor.



- **Input help for the Table editor: creating new rows**



A larger number of similar tags can be created within the table by "dragging rows". To do this, the left mouse button must be held down and then the lower right-hand corner of the selected area must be dragged down into the empty area in the table. When the entries are created automatically, the tag names with indices and the memory area According to the selected data type are incremented.

Depending on the selection, all other cells are reset to the default new entry (selection of name cell) or adopted in their entirety (selection of the complete row).

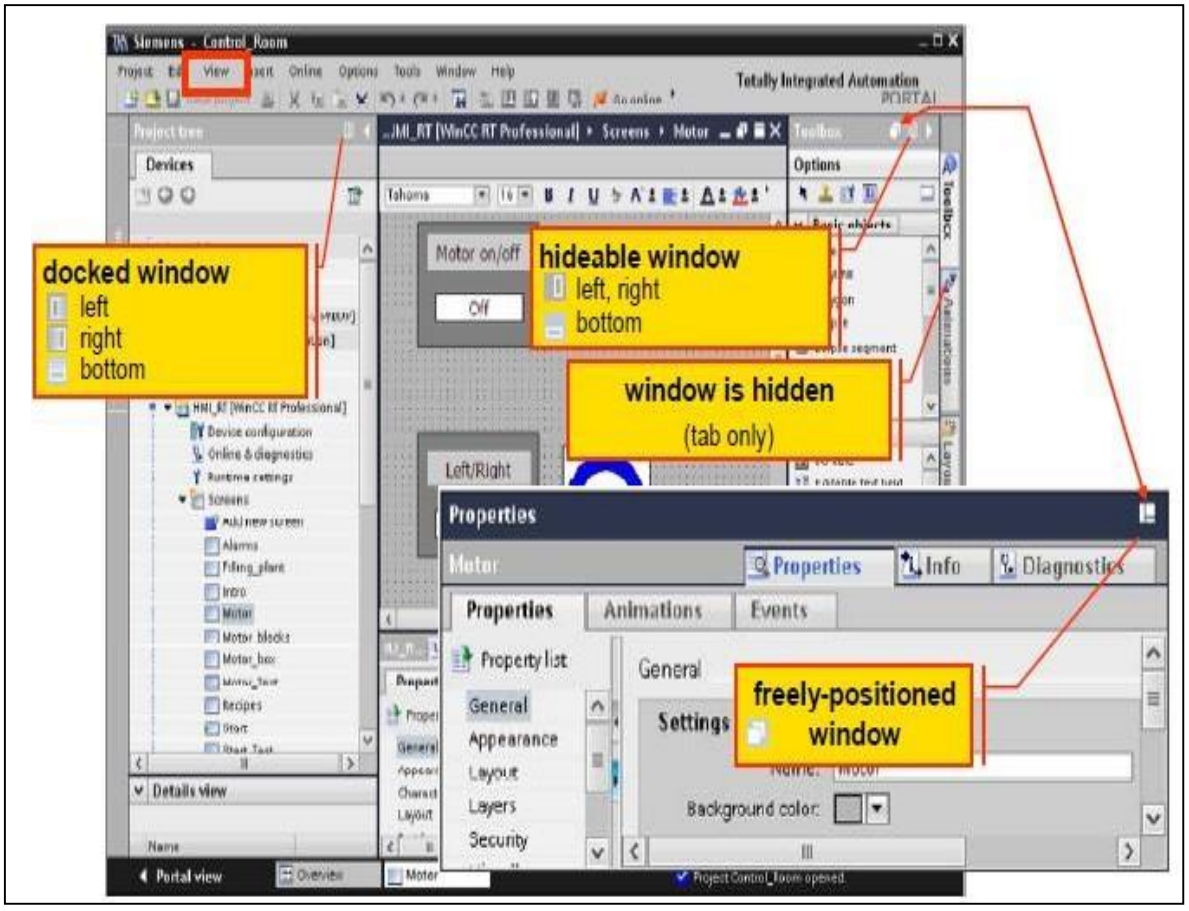
- **Selecting a row**

Clicking on the symbol  at the start of a row selects the entire row.

- **Individual design of the user interface**

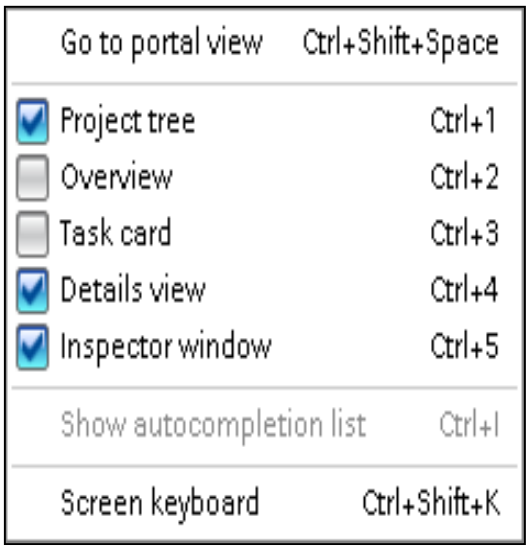
The positions and characteristics of windows can be configured individually. You can hide Windows that are seldom required and enlarge the surface of the work area. The current configuration of the engineering user interface is saved in the user profile of Windows.

On saving the project, the positions and characteristics of windows are automatically saved with it.



- **View menu**

Using the "**View**" menu function or assigned hotkeys, the windows listed below can be opened or closed or reduced in size.

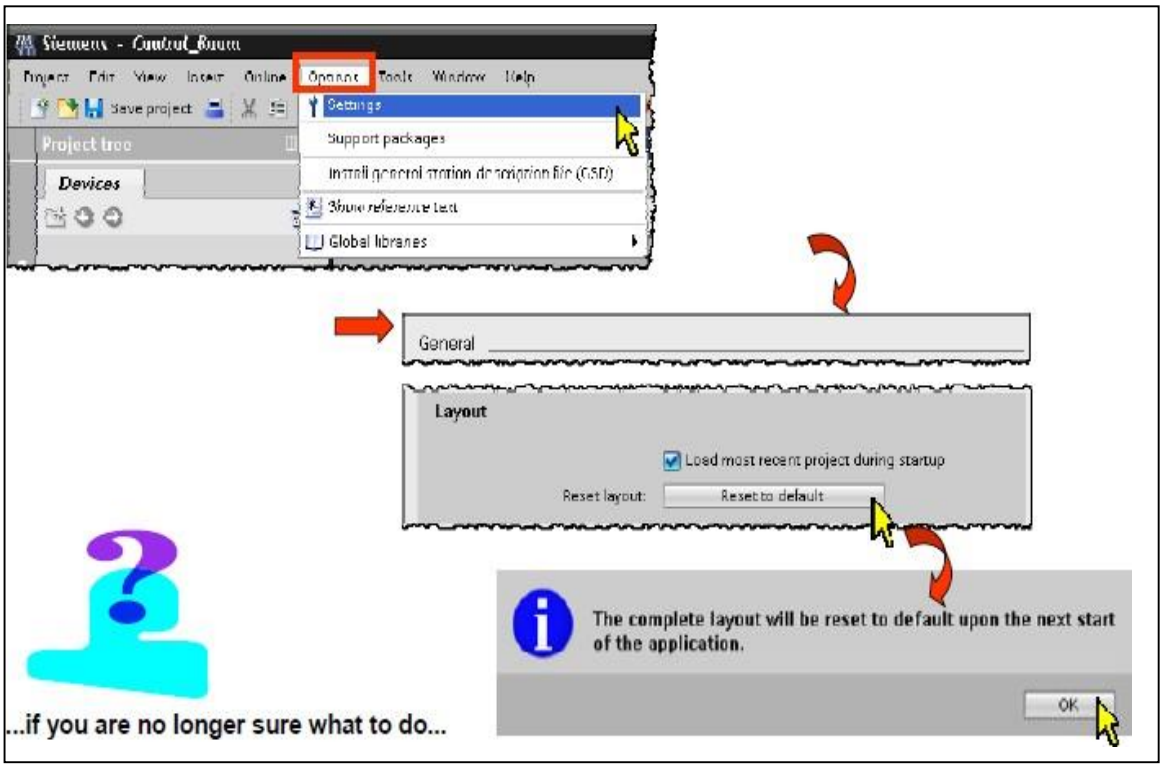


- **Window layout options**

  - Docked window
    - Fixed location and fixed size on user interface
    - Position possible at left, bottom or right outside the work area

    - Always open, reduces work area
- Hide able window
    - Hidden at edge of user interface
    - Position possible at left, bottom or right, superimposed on work area when open
    - Default status = window closed, and tab displayed at edge of the user interface
    - Mouse click on the tab opens the window
    - Closed automatically the next time there is a click outside the window area.
- Freely-positioned window
    - Can be positioned anywhere on the user interface
    - Permanently covers the user interface area underneath it clicking the functions in the window title bar changes between the mode freely positioned, docked and hidden. Hidden windows are opened by clicking on the tab and closed again by clicking outside the window area.

Reset user interface to defaults



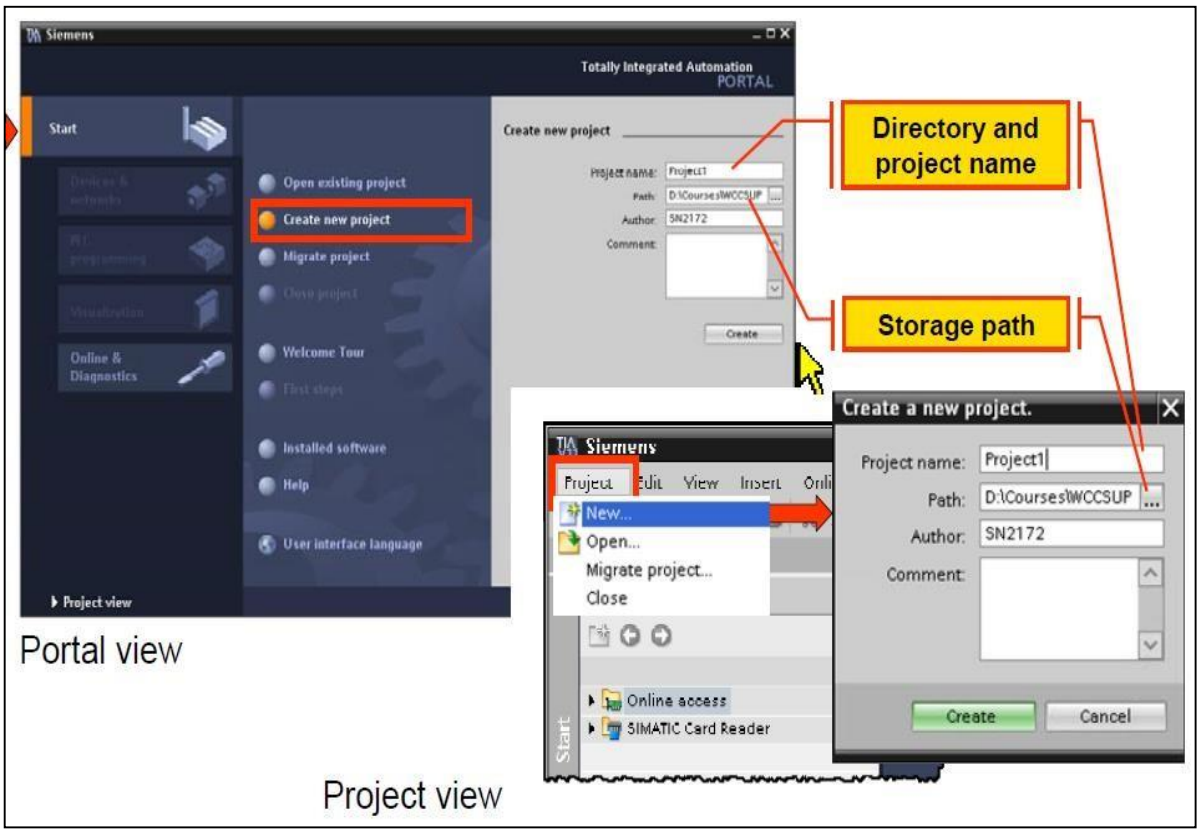
The next time the TIA portal is started, the window layout is then reset.

- Framework full screen
- Project tree docked left
- Details view opened
- Task cards docked right
- Inspector window docked at bottom

Project creation & basic exercises

Creating project

Go to **create new project**.



The basis for configuration of HMI devices a framework project. Here, the relevant devices can be inserted and configured.

There are two ways of doing this:

1. In Portal view

"Start" portal > **Create new project**

2. In Project view

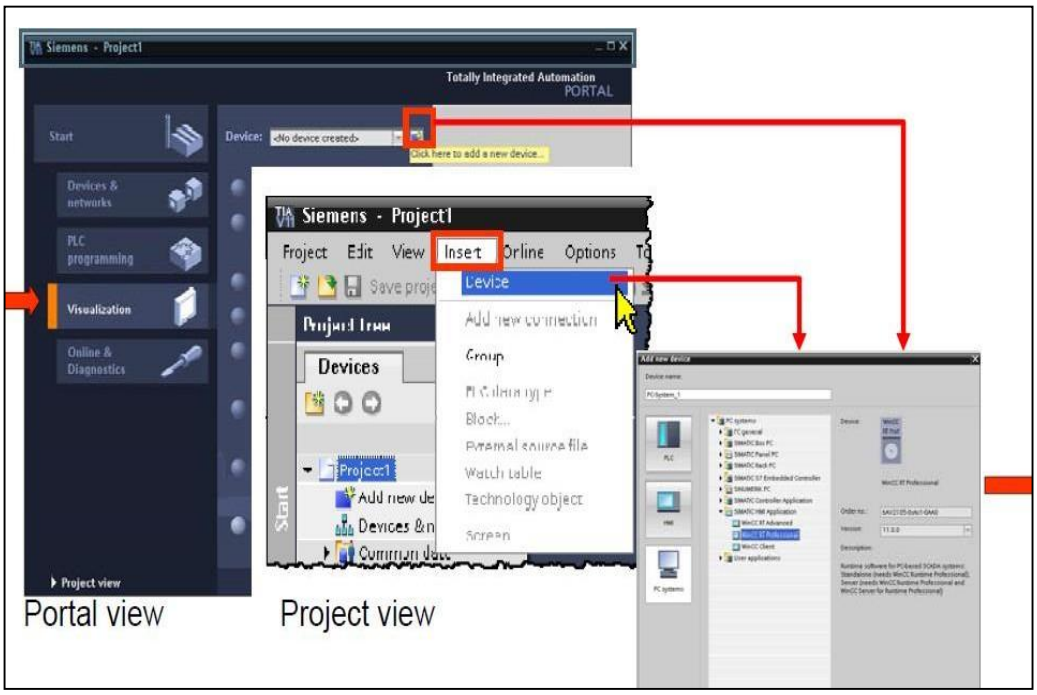
Project > **New**

Adding an HMI device

Procedure

There are two ways of doing this:

1. In Portal view

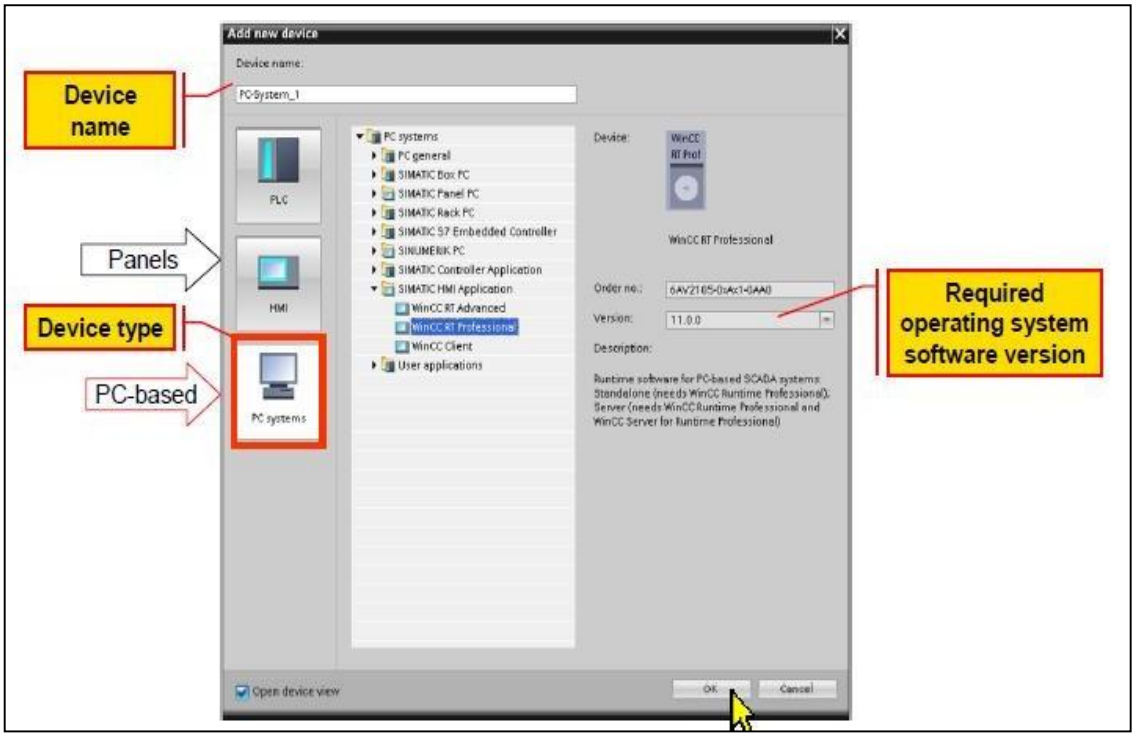


"Visualization" portal >

2. Project view

Insert > **Device** or shortcut menu (**right mouse button**) on the project name

The device dialog is then identical however it is opened.



The device type and the version of the target device must be specified for each HMI device. With PC systems, the software version of the WinCC RT software installed on the target system must be checked. The (panel) device version is the required version of the operating system image. This is displayed in the Loader menu during startup of the (panel) HMI device.

Example:

- Opening the ***TIA Portal*** project "***Control Room***"
- Add an ***HMI device*** → device type "***PC system***" with the SIMATIC HMI application "***WinCC RT Professional***"
- Setting up the engineering user interface **(*language*)**
- Inserting the first ***screen***
- Saving the ***project***

Establishing connection Setting Ethernet address

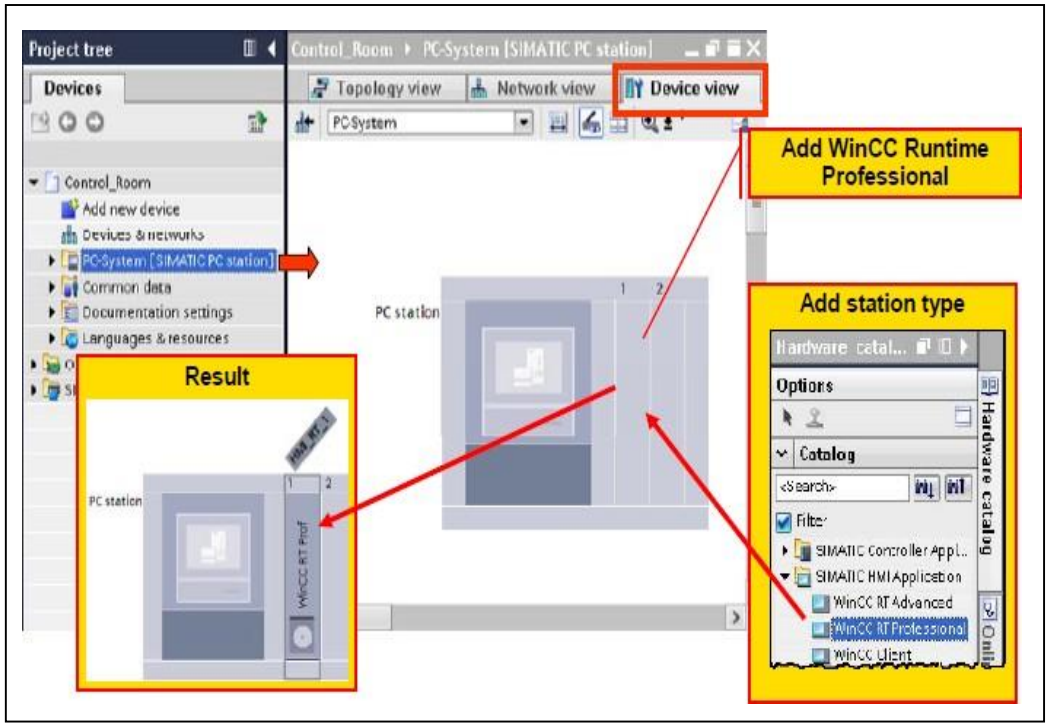
If you are adding any device with your PC, First ensure that which is your **PC's IP address**. For that please go to:

Control panel > open network and sharing center > Local Area Network-properties > ipv4 > IP address.

And also ensure that IP address of your PC and PC SYSTEM should be same.

Connection to PLC:

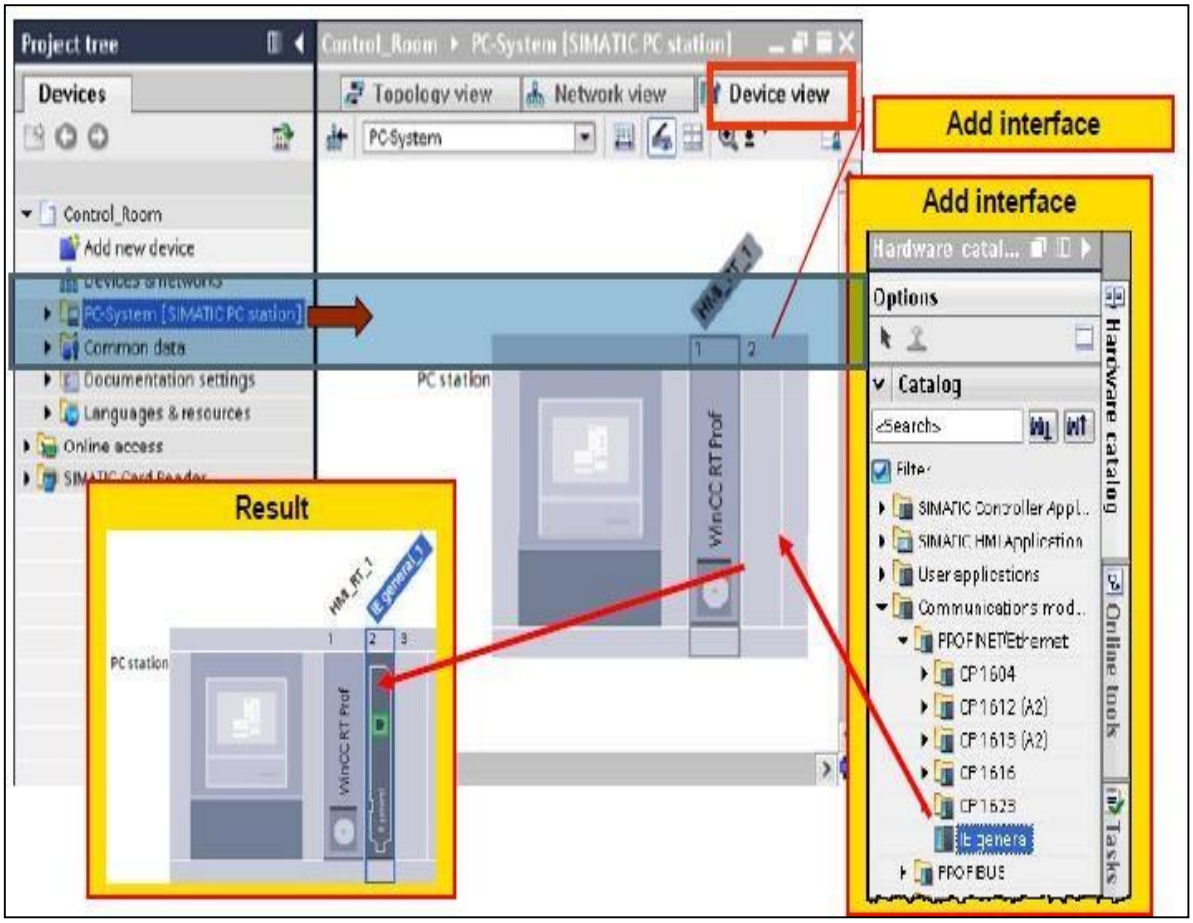
With PC-based WinCC systems, the assignment of the station type is made using the assignment of the HMI application from the hardware catalog.



"Hardware catalog" task card

From the module section "***SIMATIC HMI application***", take the "***WinCC RT Professional***" Component. The slot used in the HMI system is of no significance. The required communications interface (for example: ***IE general*** (Ethernet interface)) must also be added.

In the next step, the required communications interface (for example ***IE general*** (Ethernet Interface)) is added to the "***WinCC RT Professional***" component.

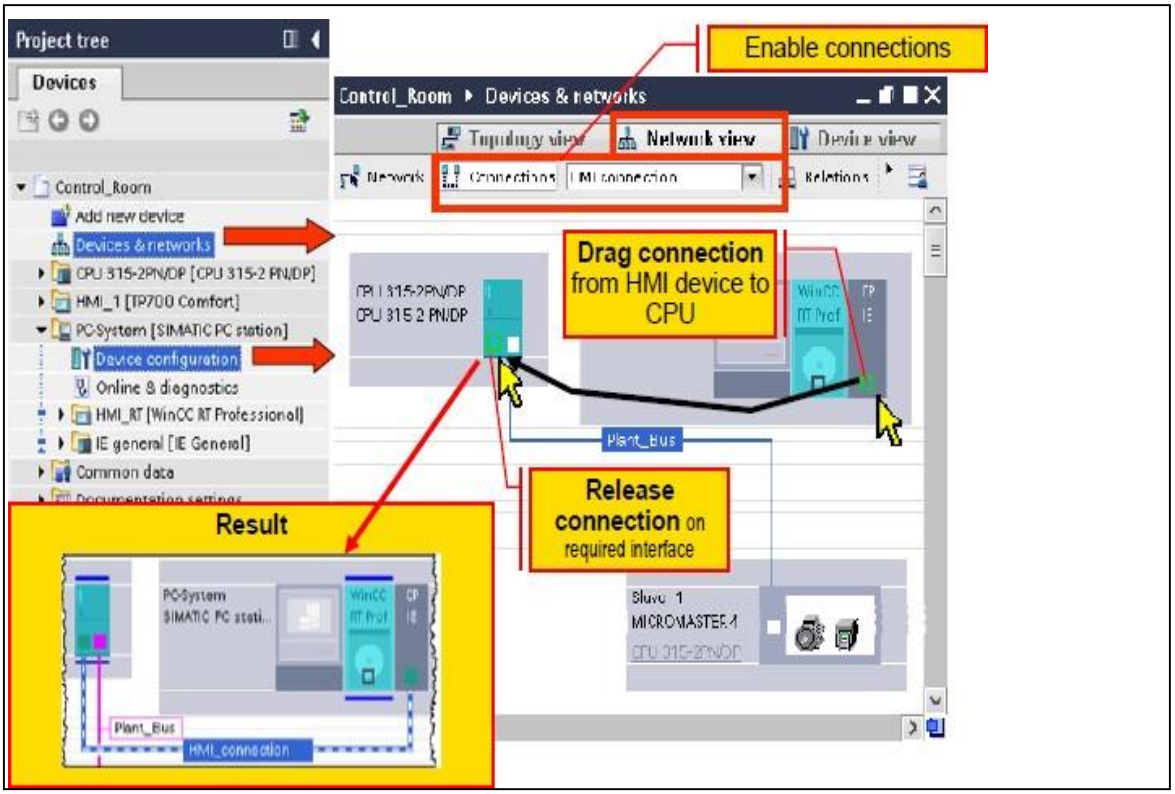


The communications interface is selected in the **hardware catalog** → task card "**Hardware Catalog**" in the modules section ***communications modules*** > ***PROFINET/Ethernet*** > ***IE general.***

Creating connections

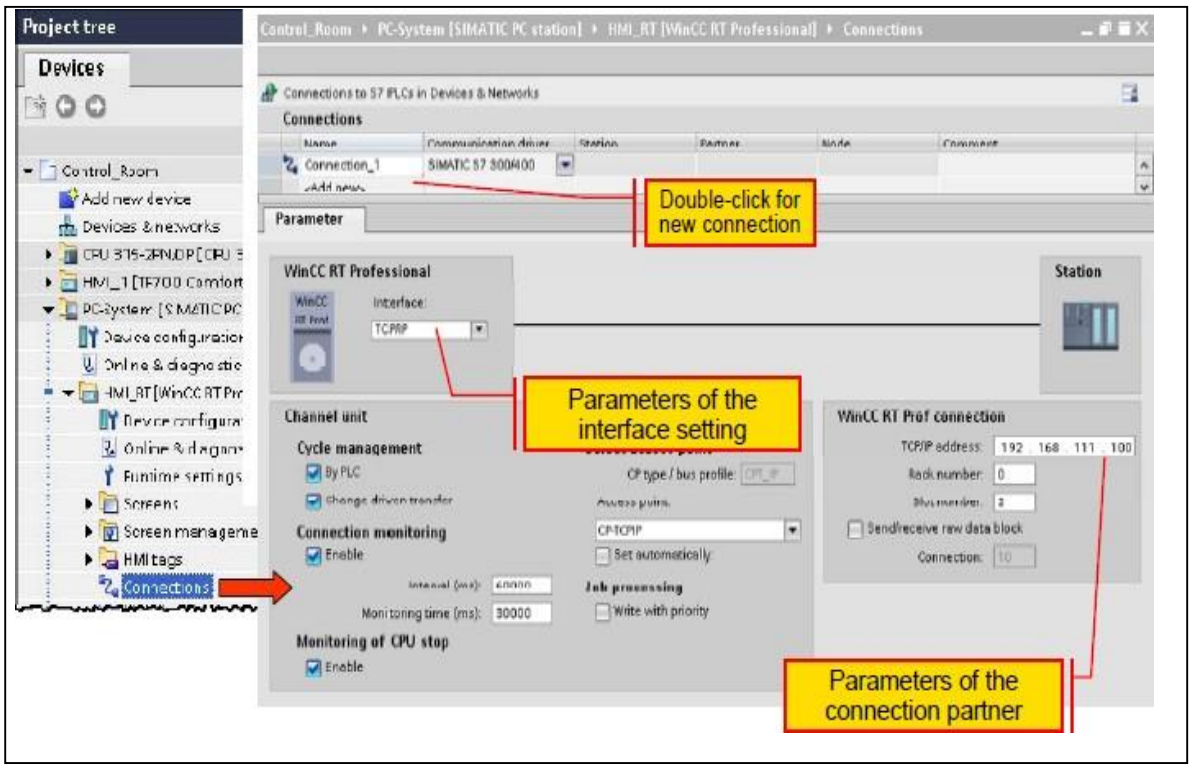
Networking and connections are configured in the Network view. The connection between the PC system and the **SIMATIC S7 CPU 315-2PN/DP** is established graphically by dragging a connecting line between the interfaces displayed in green (holding down the left mouse button).

Networking does not automatically produce a connection. When a connection is created, the networking is automatically created at the sometime.



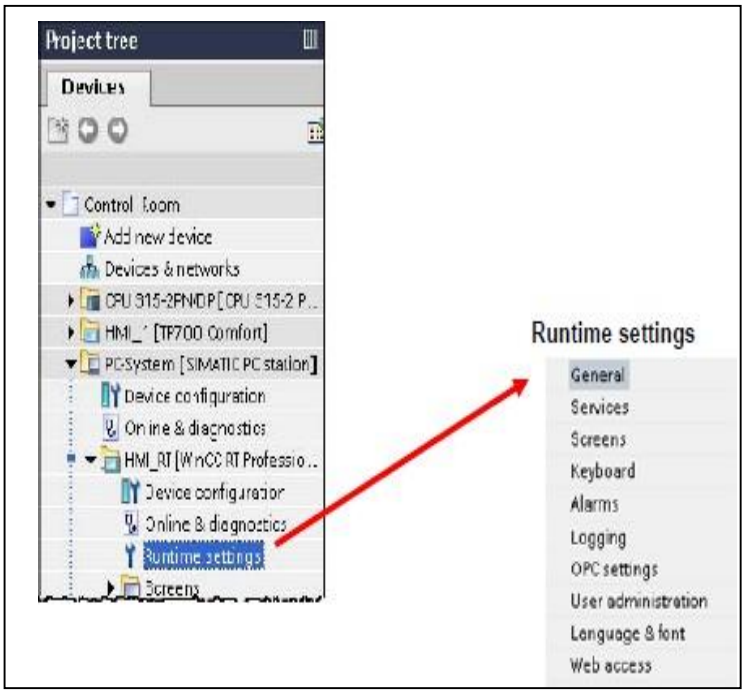
Connection to PLC not integrated

For non-integrated projects, all communications interfaces must be configured on the HMI device.



Runtime settings

The Runtime settings specify the parameters for the various components of a WinCC RT Professional project



General

Start screen

The start screen is the initial screen opened at the start of the project in runtime. A start screen must be defined for each HMI device. The first screen created with the "***Add new screen***" function ("**Screen_1**") automatically has the Start screen property

Services

The program modules required in WinCC RT Professional Runtime mode can be selected individually here. With client-server systems, for example, the program module "Screens in Runtime" is disabled on the unmanned WinCC server PC system to relieve the system software.

Screens

The properties of the WinCC RT Professional user interface (screens) required in WinCC RT Professional Runtime mode are selected here.

Keyboard

In this dialog, the keyboard entries for the WinCC RT Professional Runtime mode are selected, Such as disabled keyboard actions, keys for actions (hotkeys) etc.

Languages

The languages required in WinCC RT Professional Runtime mode are enabled here.

Runtime simulation

Simulation systems provide effective support with the development of programs and the actual application. A simulated test environment including controller and process reduces, for example, commissioning times and thus costs. Depending on the progress of the project, the functionality to be tested, and the degree of integration, three kinds of simulation are available:

- HMI tag simulation with tag table

  Testing the configuration without a connected controller and without a running process via the tag simulator cost effectively checks an HMI project for internal consistency. In the simulator, the configured tags are simulated; for example, do the configured colour changes agree with the value pattern of a tag? The configured tags can be manipulated, activated and deactivated in the course of simulation. The simulator provides various kinds of simulations for this (sine, random, increment, etc.).

- HMI project simulation with simulated controller (PLCSIM)

  To test the interaction of a WinCC configuration with the PLC, the project can be tested on the engineering PC in combination with a simulated controller (PLCSIM). The Power Tags are directly provided with the values of the simulated PLC program.

- HMI project simulation with a connected, real controller the simulation of a WinCC configuration in communication with a real controller provides maximum integration and accuracy. In this case, the WinCC simulation connects directly to the PLC in the plant via PROFIBUS, Ethernet, or PROFINET. The Power Tags are directly provided with the values of the real process in the plant.

Simulation of the Runtime system

1. Configuration
2. Simultaneous simulation in a Runtime environment

Configuring buttons and output

Procedure:

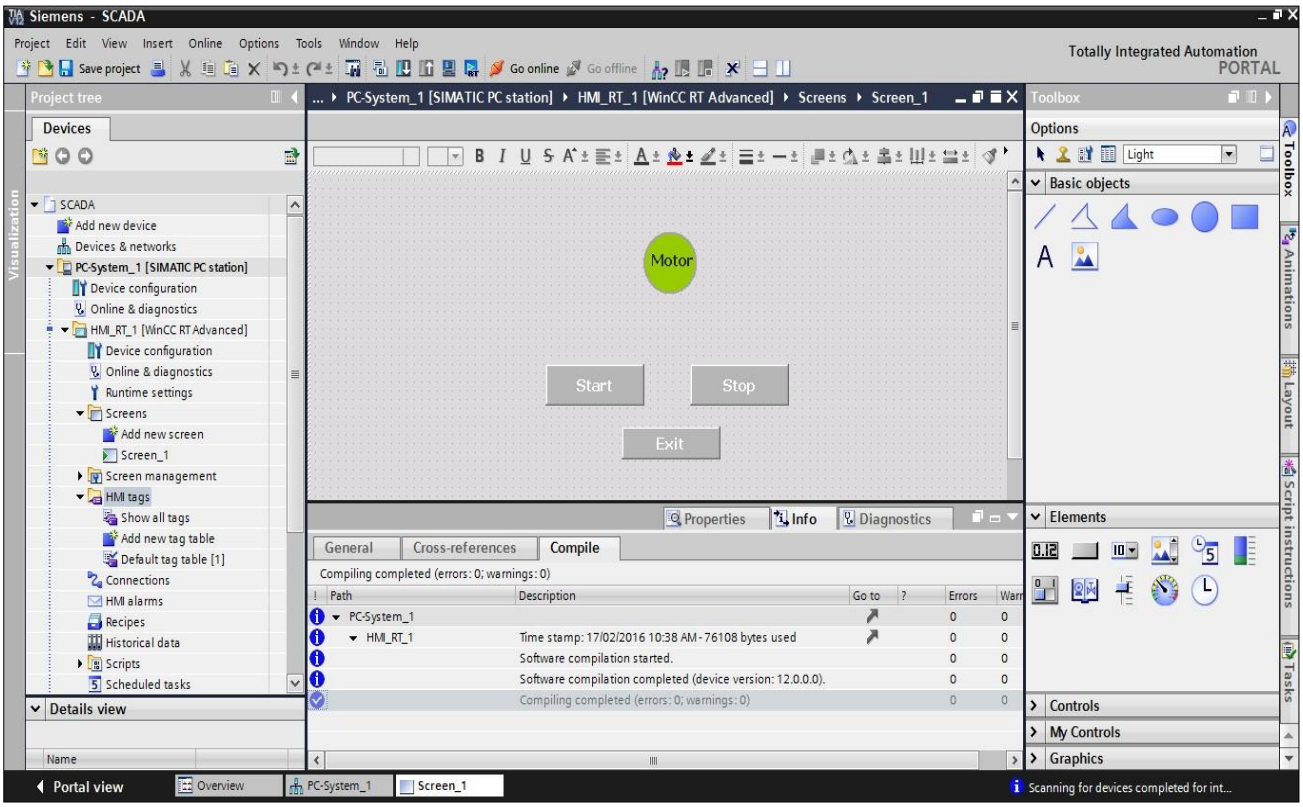
1. Open →**TIA portal v13**
2. Create a new project
3. Give project name, path, author, comment and click on create
4. Go to →project view
5. In project tree go to →"add new device"
6. Go to → PC system>Simatic HMI application>WinCC RT advanced
7. Click → ok.
8. Go to →

   Hardware catalog>pc system>communication modules>Profinet/Ethernet>IE general.

   Drag and drop to Simatic PC station in device view

9. Go to → project tree>hmi_rt_1 >screen>add new screen
10. Take three buttons from tool box> elements, and also take one circle from >toolbox > basic objects
11. Rename buttons to "Start", "Stop", and "Exit" and add text field below circle and give name to "Motor"
12. Go to → project tree>HMI tag>default tag table
13. Click on "add new" and add 'start tag", "stop tag", and "lamp" and data type "bool"
14. Now go to screen>screen_1>select "start button and click below its "properties"
15. In properties go to "events tab" >select "click" > double click on 'add function' and search for "set bit", after selection in place of "tag(input/output" assign "start tag".
16. Now select "stop button" properties >event>click> search for "invert bit"> assign "start" tag
17. Click lamp > properties> animation>appearance >dynamic colors and flashing
18. Add tag name" Start" in below table add range "0" and "1" set back ground color "red for "0", and "green for "1".
19. Now select "Exit" button properties >event>click> search for "stop runtime"
20. Compile the project.
21. Start "runtime"

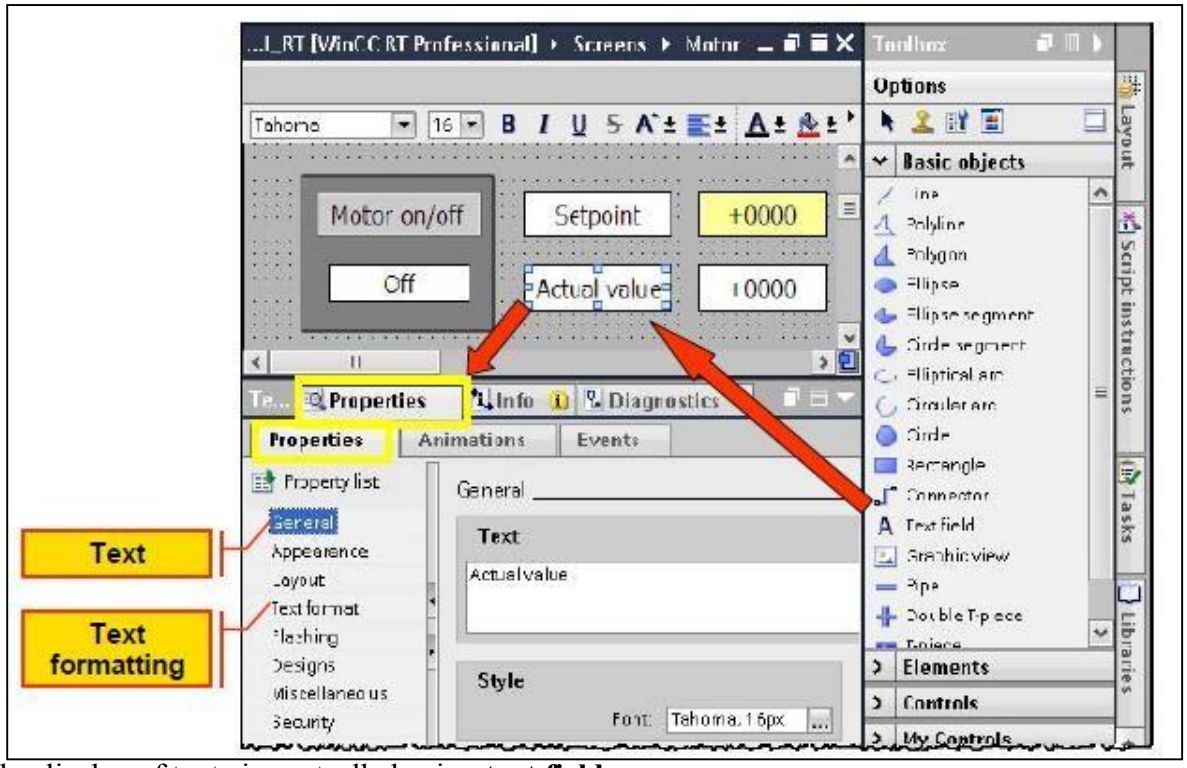
TIA portal Screen



Runtime Screen

Basic Configuration

Configuring text display



The display of texts is controlled using **text fields**.

Task card >> Toolbox > Basic objects > Text field

The display properties are always based on the entire text field. If texts with various display Properties need to be displayed, these must be configured in different text fields.

Input and output for alpha numeric display

The value display is configured using **I/O Fields**.

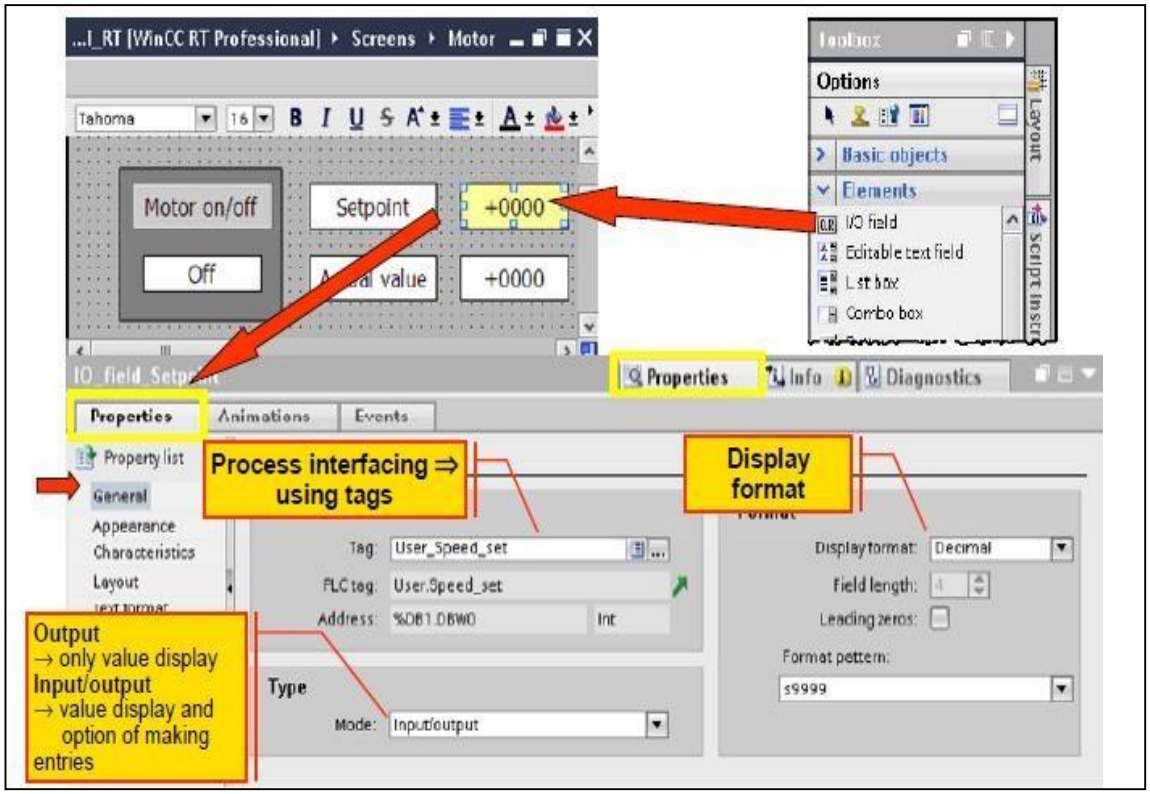
Task card >> Toolbox > Elements > I/O field

Properties

The **mode** specifies either write access or display only (input or output type or both the types)

The **display format** specifies in which format the tags will be displayed

The **Tags** specifies connection of I/O fields to the corresponding tags



Configuring text and graphic list

Text list

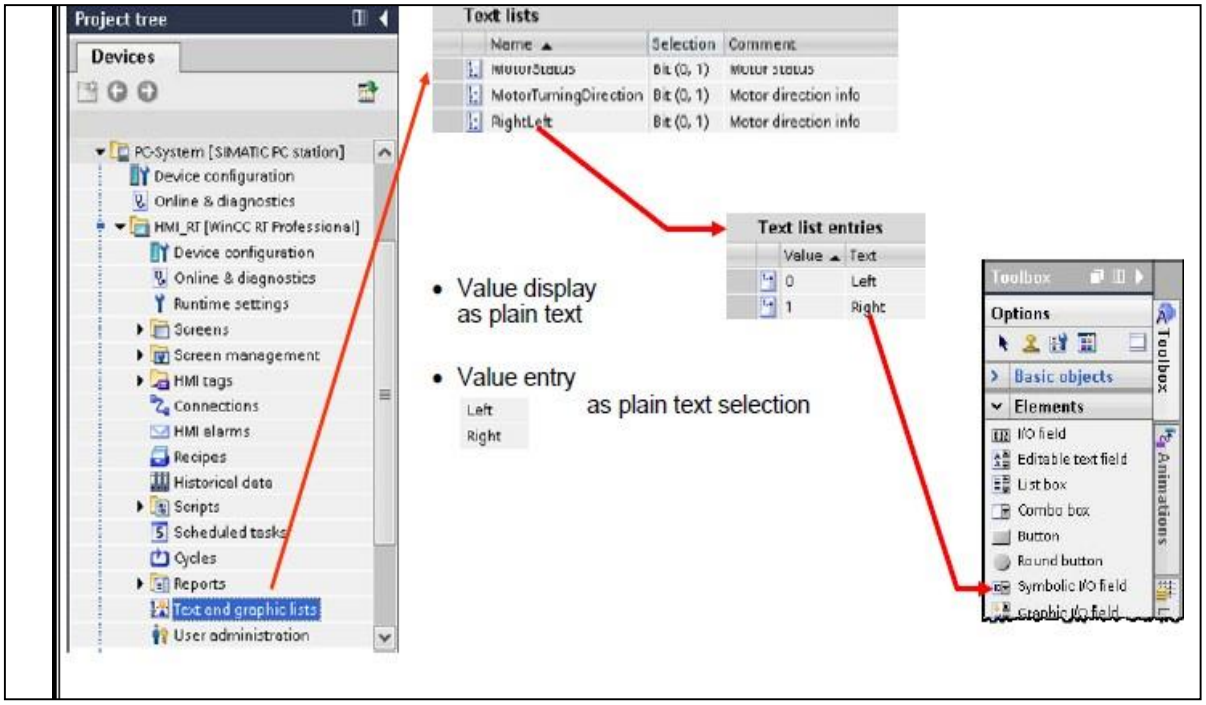
A value can also be represented as a plain text display. To do this, a suitable text list must be configured that allows the HMI device to assign the actual process values (numbers or number ranges) suitable texts for display.

Areas of application for text lists

Configuration of a plain text display or text selection list with a symbolic I/O field configuration of a text output for the value in a discrete alarm

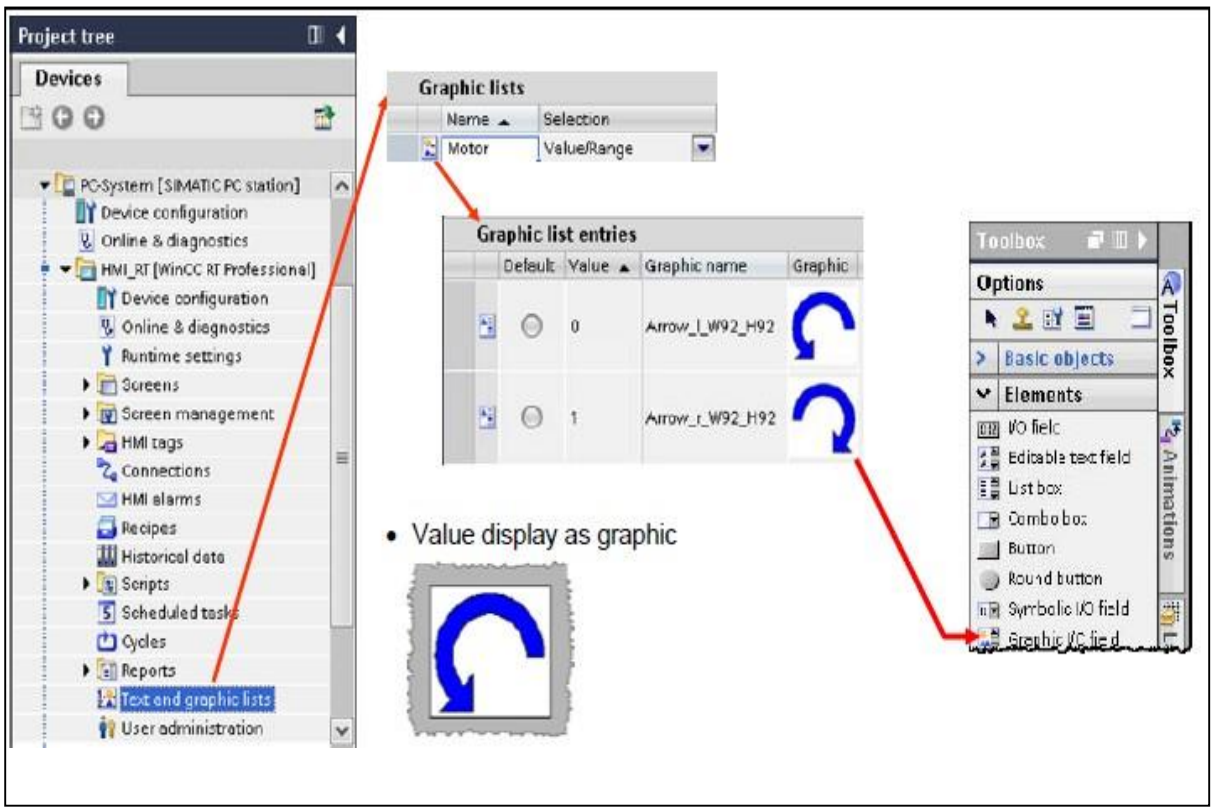
Graphic list

A value can also be represented as a graphic view. To do this, a suitable graphic list must be configured that allows the HMI device to assign the actual process values (numbers or number Ranges) suitable graphics for display.



Areas of application for graphic lists

Configuration of a graphic value display with a graphic I/O field



Movement and fill property

Movement

You can configure dynamic objects in such a way that they move on a certain track. The movement is controlled via tags. The object moves every time the tag is updated. You can only program one type of movement for each object.

Requirements

- You have created a tag.
- You have opened a screen which contains at least one dynamic object.
- The Inspector window is open.
- The toolbox window is displayed.

Procedure for movement

1. Select the screen's **object** you want to control dynamically.
2. The object **properties** are displayed in the Inspector window.
3. In the **Inspector window** select "**Properties** > **Animations**".
4. The animations available for the selected object are displayed.
5. Select "**Horizontal movement**" and click the button (you can select other motion also i.e. vertical, horizontal etc.)
6. The parameters of the animation are displayed.
7. A transparent copy of the object is shown in the work area, which is connected to the source object by means of an arrow.
8. **Select a tag** for control of movement.
9. Move the object copy to the relevant destination. The system automatically enters the pixel values of the final position in the Inspector window.
10. **Customize** the range of values for the tag as required.

Fill property

Fill property is used for Visual representation of tank filling in process industries, tank level continuously get updated using PLC program's tag.

Requirement

- You have created a tag.
- You have opened a screen which contains Bar elements (used for tank filling properties)
- The Inspector window is open.
- The toolbox window is displayed.

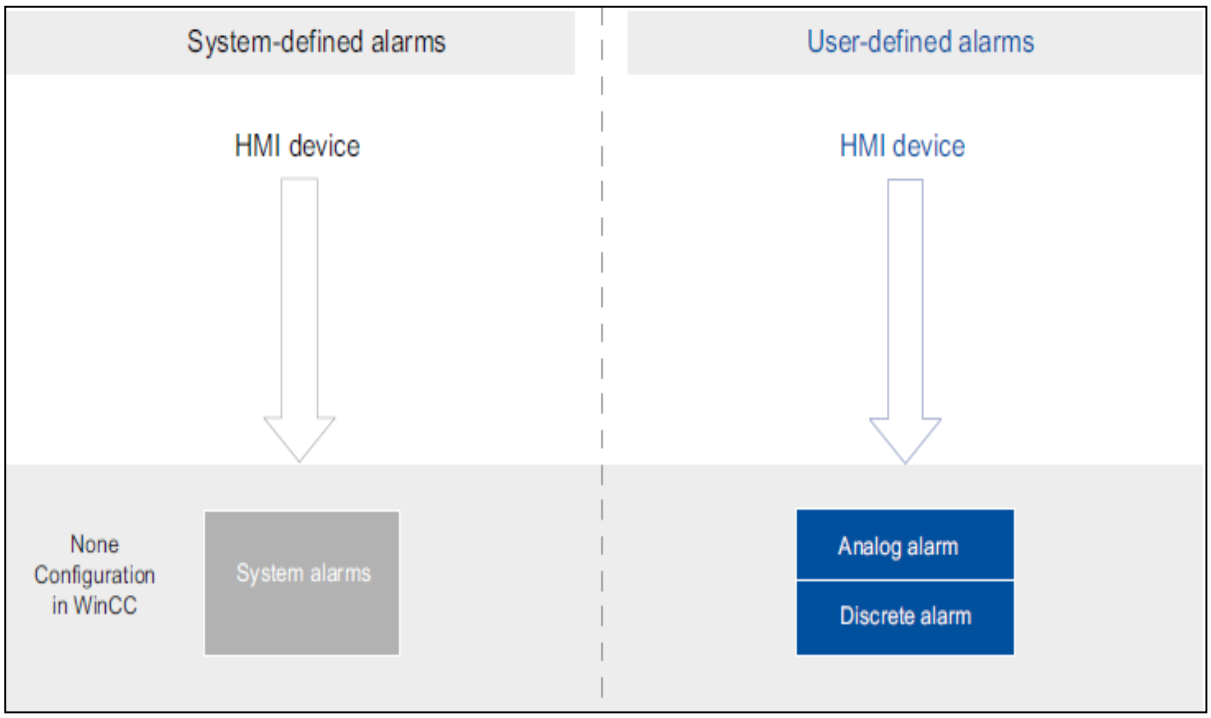
Procedure

1. Select Bar in the screen that you want to fill dynamically.
2. The Bar properties are displayed in the Inspector window.
3. In the Inspector window select "Properties > General".
4. Assign processing tag for tank filling (i.e. From Level sensor)
5. Set scaling for tank level.
6. Compile and test your program in Runtime.

Alarm Configuration

General Information

With the alarm system, events from the controller or from the monitoring function in WinCC (operating states, faults, user input etc.) Are displayed in the form of alarms, logged if necessary and reported and acknowledgements by the operator are accepted. To allow this, alarms need to be configured in alarm classes. To allow the history to be followed as well, the alarms are stored on the hard disk in a long-term historical alarm list on the local computer. The alarm system of WinCC Professional is based on the directions of DIN 19235. The alarm system of WinCC Professional provides discrete, analog, user and controller alarms.



Alarm logging processes various alarm procedures used by the PLC and the HMI device. The alarm procedures can be broken down into system-defined alarms and user-defined alarms:

- User-defined alarms serve to monitor the plant.
- System-defined alarms are used to monitor the HMI device or the PLC.

The detected alarm events are displayed on the HMI device. You can use the alarm logging system to log alarms from the ongoing process. Targeted access to the alarms combined with supplementary information about individual alarms ensures that faults are localized and cleared quickly. This reduces stoppages or even prevents them altogether.

Alarm types in WinCC

WinCC supports the following alarm types:

User-defined alarms

- Analog alarms

  Analog alarms are used to monitor limit violations.

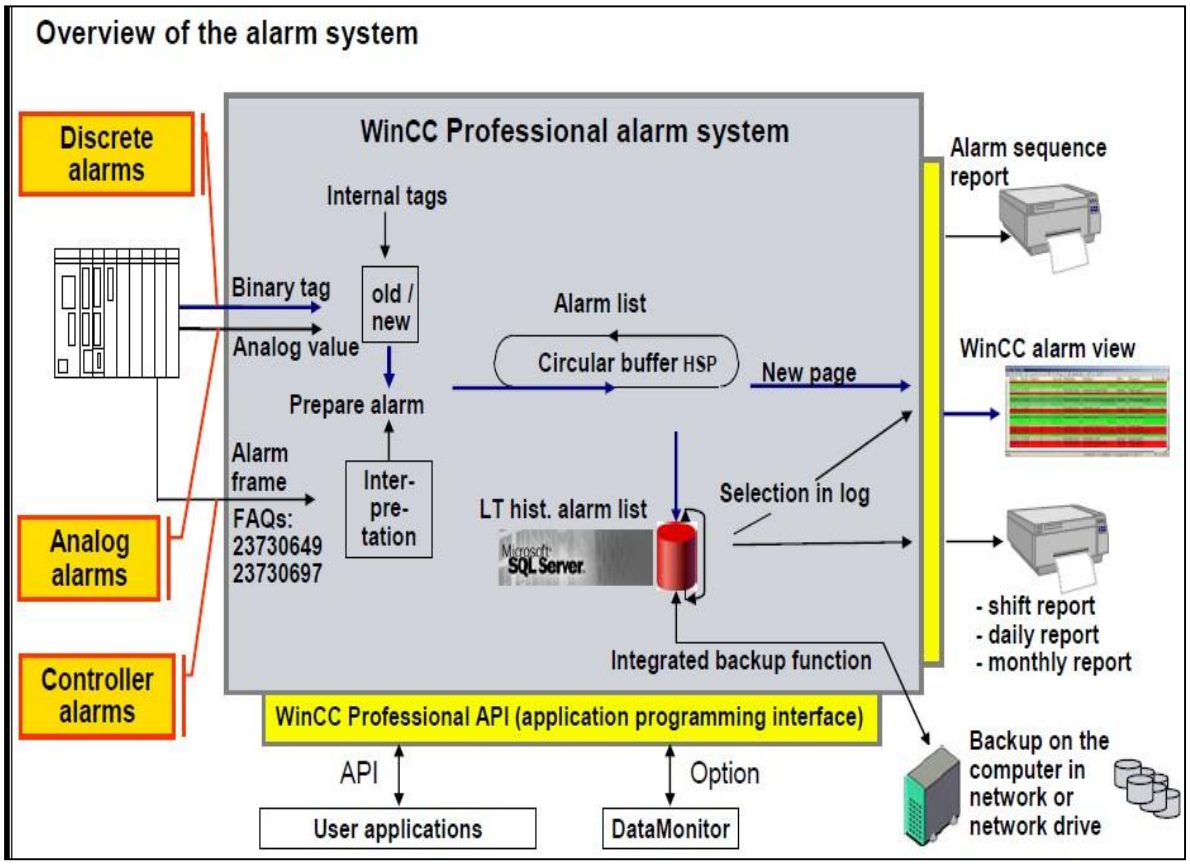
- Discrete alarms

  Discrete alarms are used to monitor states.

System-defined alarms

- System events

System events belong to the HMI device and are imported into the project. System events monitor the HMI device.



Message blocks

Minimize downtimes – by means of alarms and messages

SIMATIC WinCC records process signals and local events, saves them in archives and makes them available filtered or sorted, as needed. Messages can occur via the derivation of the individual bits of a Power Tag (max. 32), as a result of a chronological message frame directly from the automation system, as a result of analog alarms due to any number of limit value violations, or due to an operation (operation message). Each message can be configured in such a way that the operator has to acknowledge it. Now it is also possible to compare tags with a defined value. In doing so, hysteresis values can be set and ranges can be monitored easily.

User-definable message structure

Because the message structure is freely definable, it can be tailored to the special requirements of your plant. Dividing the structure into as many as 10 different text blocks (plant ID, fault location, text, etc.) Leads to greater clarity of the information and allows targeted analyses to be initiated in connection with the filtering or sorting function. Differentiation of as many as 16 message classes makes easy fault and status messages just as possible as the separate preparation of alarms, warnings, faults and errors for several areas of the plant. Within a message class (e.g. Alarm), up to 16 priorities can also be differentiated.

User-friendly message view

Messages are displayed on the screen via the user-configurable WinCC Alarm Control. Here, the display of the message information, for example, can be adapted exactly to the needs of the operator. The settings made can be saved in user-specific or global templates. WinCC Alarm Control for displaying current/historical messages Based on the contents of the individual message blocks, filtering, selecting and sorting is possible in the screen, e.g. Chronologically, according to priorities or fault location. The contents can then be exported directly as a CSV file or printed as a report. A freely definable toolbar function also provides a high degree of flexibility. In this way, for example, your own project-specific functions can be integrated. To maintain a clear overview of a large number of incoming messages, the operator can suppress unimportant operational messages on the screen display by means of Alarm Hiding. The messages continue to be archived in the background.

Archiving and logging messages

The Microsoft SQL Server is used for archiving message. This ensures complete recording of all events. Messages are archived in the case of so-called message events, for example, when a message occurs and when the status of the message changes. In the message sequence report, the messages can be (selectively) documented chronologically. All of the status changes (came in, went out, acknowledged) of all currently pending messages are output to a printer. In the message archive report, specific views of the archived messages can be generated.

Steps in configuring an alarm

Below is the procedure to configure the alarm in WinCC.

1. Configure alarm blocks

   - System blocks

     System blocks are system data, for example date, time, alarm number and status.

   - User text blocks

     User text blocks contain the alarm text with the description of the cause of a fault and additional texts with information, for example the location to help localize the fault.

   - Parameter blocks

     Parameter blocks are used to link the alarms to process values, for example, current fill levels, temperatures or speeds. Up to 10 parameter blocks can be configured per alarm.

2. Set alarm classes

   - Configure your own alarm classes.

3. Specify alarm settings

   - Display suppression.
   - Back transfer of alarms.

4. Configure trigger tags

   - Bit string for discrete alarms.
   - Threshold tags for analog alarms.

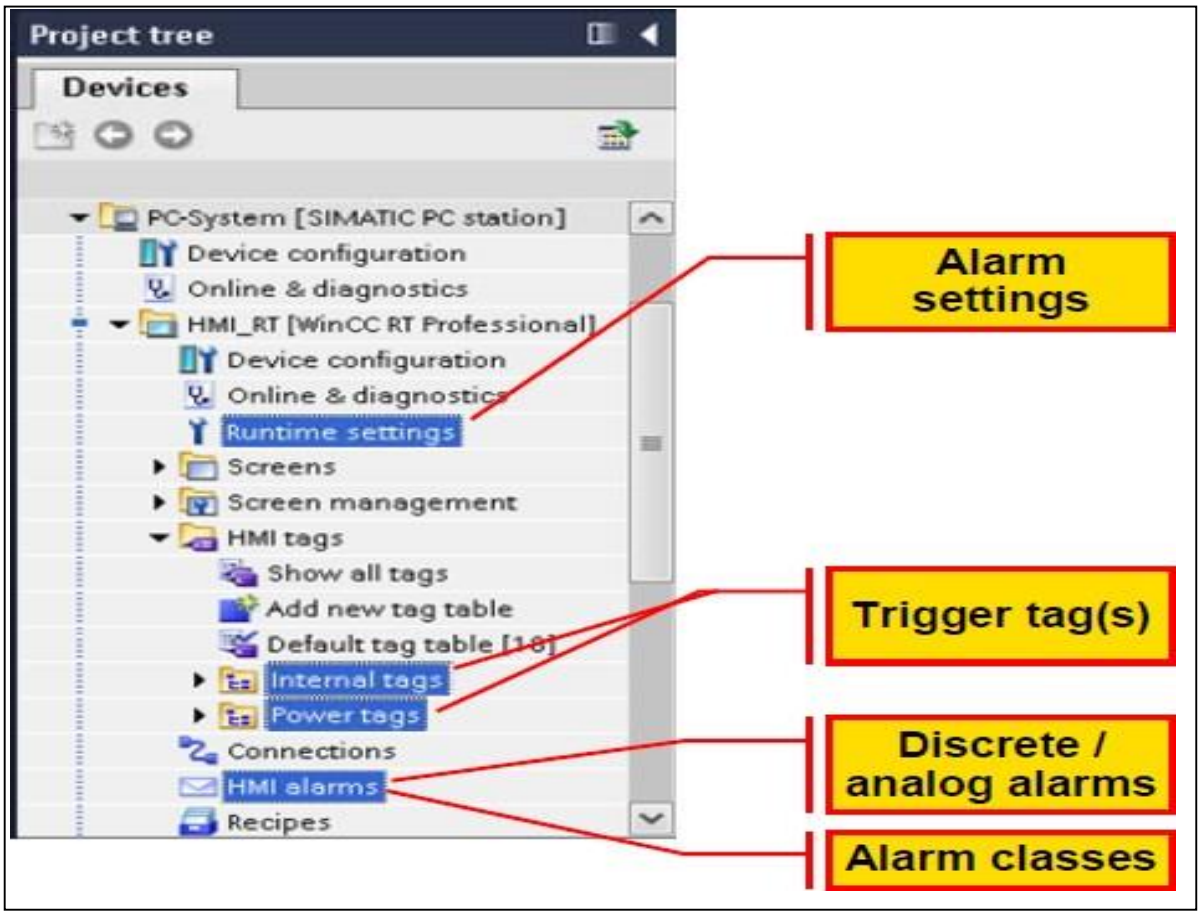
5. Configure discrete alarms

   - Configure analog alarms/ discreet alarm.

Procedure for Configure discrete alarms

1. Select **HMI alarms** in project tree.
2. Choose **discrete alarm**
3. Assign **ID** (i.e. 1, 2, 3, .etc.)
4. **Alarm text** Write text that you want display on alarm screen when it will be executed (example “tank limit exceed!”).

5. **Alarm classes** : define class of alarm (i.e. Error, acknowledge, warning)
6. **Trigger tag**: Assign HMI tag that you have chosen for HMI alarms (first you have to create HMI tag Assign HMI tag's memory bit that can be used as Triggering Bit to triggered the alarm.
7. Take **Alarm View** from **Toolbox > control. (**you can also set **appearance** for Alarm view**)**
8. **Compile** and test your program in **Runtime**.



Displaying alarms

So as not to overload the plant operator with information, the display of alarms can be suppressed automatically or manually. If alarms are suppressed (hidden), they are logged but not displayed.

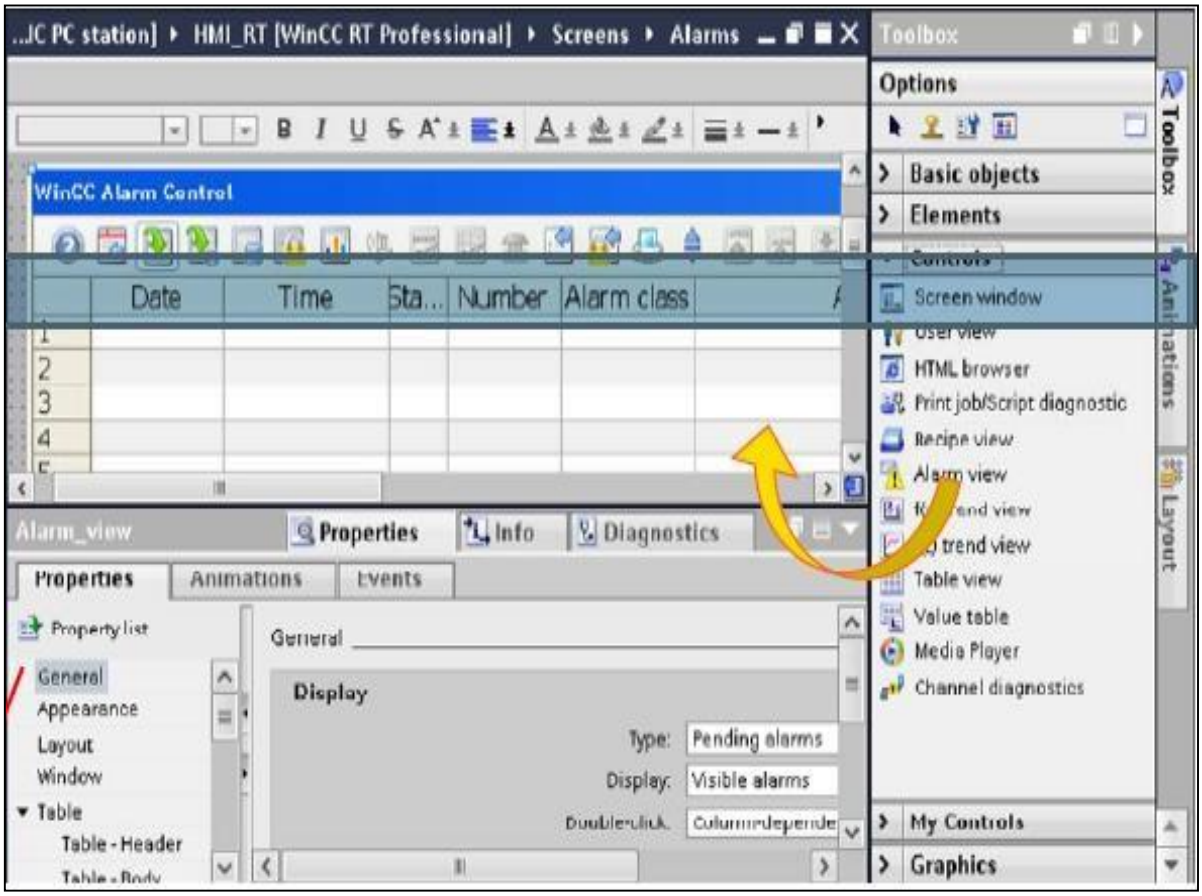
There are two ways of suppressing the display of incoming alarms:

Automatic suppression of the display

The alarms are not displayed depending on a certain system status. To do this, alarm suppression must be configured for certain plant statuses, for example plant startup or plant shutdown.

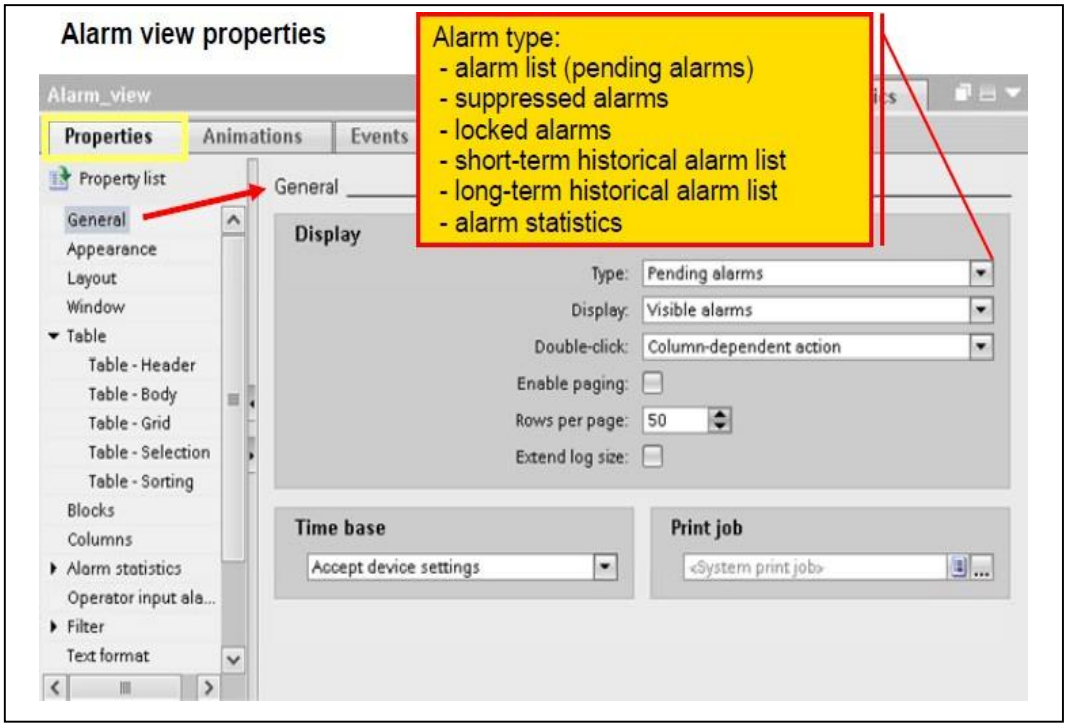
Manual suppression of the display

The user suppresses the display of an alarm in runtime when necessary using the "Hide alarm "button in the alarm view. When necessary, the user can display the alarm again using the "Show alarms" button.



Alarm View properties

The settings for the position, geometry, style, color and font of the object are made in the Inspector window. The following settings need to be selected for an alarm list display.



General/Display:

Type:

Pending alarms

Display:

All alarms

Double-click:

E.g. Loop-in alarm: Specifies that double - clicking on an alarm, triggers a configured screen change.

General/Time base:

Accept device settings. Here, you specify the time base with the alarms are output.

Appearance:

Set the appearance for Alarm view.

Layout:

Here, the position and size of the alarm view is defined.

Window:

Settings title: e.g. "Blast furnace-W"

Table:

Activate display of the column and row headers. Specifies the properties (color, column labeling, and row numbers) of the table in the alarm view.

Blocks:

Adopt project settings with these settings, the blocks created in the alarm configuration are selected for further use in the column configuration.

Columns:

Specifies the columns of the table in the alarm view. Each column corresponds to an alarm text block. Here all available alarm text blocks should be selected. Here, only the alarm text blocks selected under "Blocks" are displayed.

Toolbar:

At least the following buttons should be available in the taskbar: **Alarm list, short and long-term archive, acknowledgment** (single/group acknowledgment), **selection dialog (filter), and auto scrolling** and for navigation in the alarm lists: First alarm, next alarm, previous alarm, latest alarm

Status bar:

Specifies the elements of the status bar. In runtime mode, these elements show, for example, information about the number of pending alarms.

Additional functionality with WinCC Runtime Professional

Alarms can occur via the derivation of the individual bits of a Power Tag (max. 32), as a result of a chronological message frame directly from the automation system, as a result of analog alarms due to any number of limit value violations, or due to an operation. In doing so, hysteresis values can be set and ranges can be monitored easily. Since the alarm structure is freely definable, it can be tailored to the special requirements of a plant. Dividing the structure into as many as 10 different text blocks (plant ID, fault location, text, etc.) Leads to greater clarity of the information and allows targeted analyses to be initiated in connection with the filtering or sorting function. Differentiation of as many as 16 alarm classes permits the separate preparation of alarms, warnings, faults and errors for several areas of the plant. Within an alarm class (e.g. Warning), up to 16 priorities can also be differentiated.

Based on the contents of the individual alarm blocks, filtering, selecting and sorting is possible in the screen, e.g. chronologically, according to priorities or fault location. Then the contents can be exported directly as a CSV file or printed out as are port. A freely definable toolbar function also provides a high degree of flexibility. In this way, for example, your own project-specific functions can be integrated. To maintain a clear over view in the event of a large number of incoming alarms, the operator can suppress unimportant operational messages on the screen display (Alarm Hiding). The alarms continue to be logged in the back ground. The Microsoft SQL Server is used for logging of alarms. This ensures gap-free recording of all

events. Alarms are logged when alarm events occur. In the alarm sequence report, the alarms can be (selectively) chronologically documented. In the alarm log report, specific views of the logged alarms can be generated.

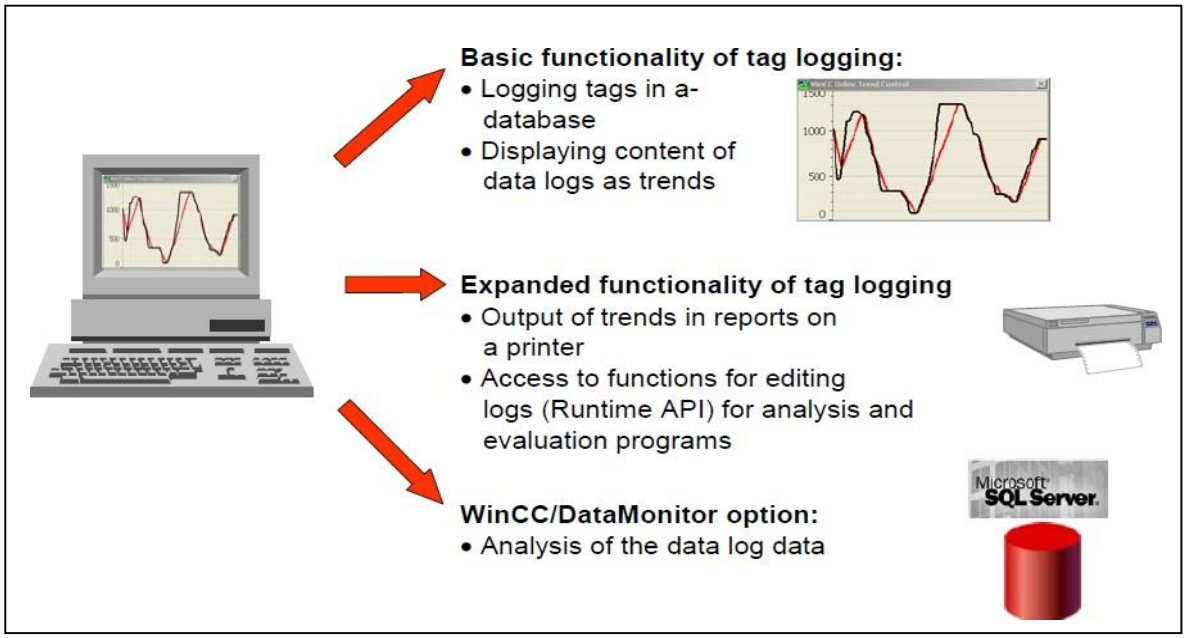
Reporting and logging system

The integrated WinCC report system prints data acquired during the runtime in configurable and page-based layouts by means of various types of reports, ranging from alarm sequence reports, system alarm and operator input reports, to user reports. Of course, these logs can also be configured in multiple languages. The report output can also be started time-controlled or event-driven or by operator input. The printer can be selected online via a printer selection dialog. The contents of a report can be determined dynamically during runtime. WinCC reports can contain data from the database and external data in CSV format as a table or trend. Data from other applications can be integrated as tables or graphics via customer- Specifically developed Report Providers

Data logging and trend configuration

Purpose

- Saving current tag values in a database for display with the WinCC trend view and for later analysis.
- Depending on the configured acquisition mode, the current tag value is entered with the associated time stamp in a database
- Acquisition modes: on demand, on change, cyclic and cyclic selective.
- Dead band: Process values outside the dead band are not logged.



For a data log, the storage location must be specified; in other words, whether data is stored on the hard disk or in main memory.

In contrast to storage in the logging database, process values logged in main memory are only available for as long as Runtime is active. Storing in main memory has the advantage, however, that the values can be written and read out very quickly.

Types of process value logging

When you configure logs, the system distinguishes between the following log types:

Process value log

Stores process values in logging tags. When configuring the data log, the process tags to be logged and the storage location must be selected.

Compressed log

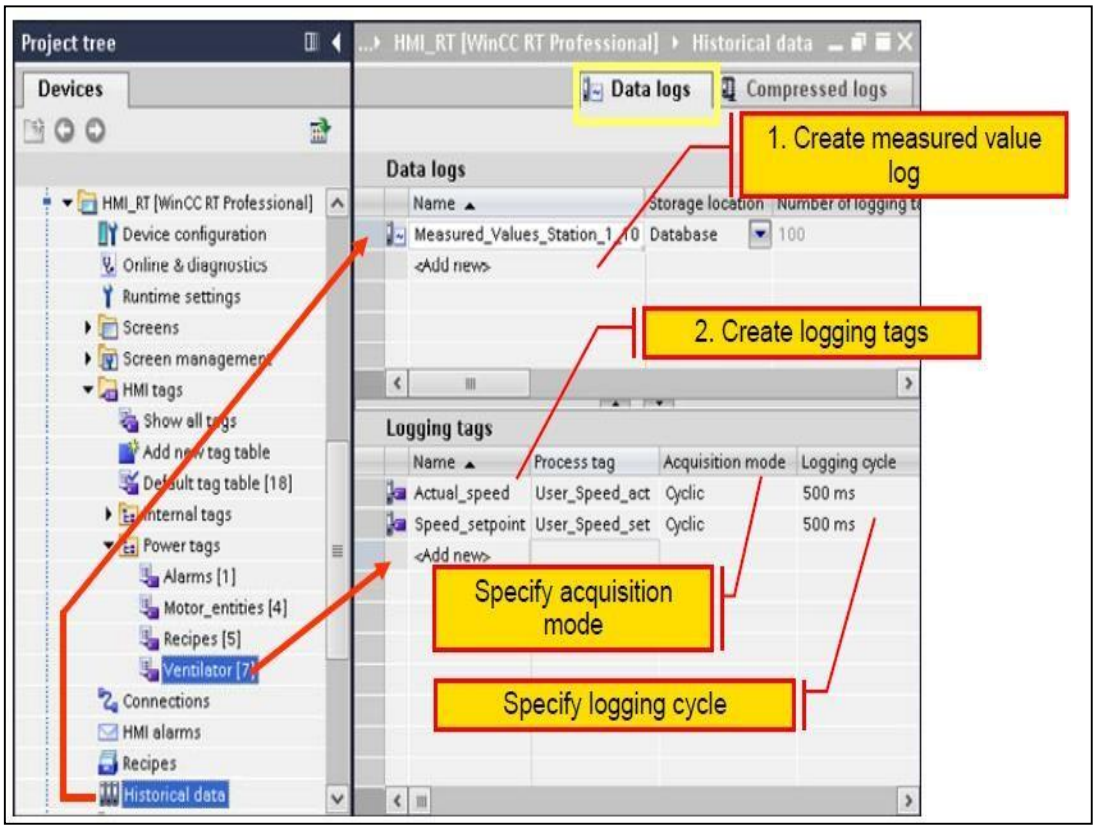
Compresses logging tags from tag logs. The compressed log can also further compress existing compressed logs. During configuration of the compressed log, a calculation method and the compression period need to be selected.

Procedure for configuring data logs

In the "Logging" area navigation in the runtime settings, you configure the logging settings for the fast and slow data logs (tag logs). Here, you specify the period for which the data of "fast data log and slow data log" are logged and the point in time as of which the data can be overwritten. In addition to this, the backup function is also enabled here and the path for swapping out the log segments for backup is entered.

The procedure for the configuration of a tag log consists of the following steps:

1. Go to → > Project tree > HMI device > Historical data > Data logs tab
2. Create new tag log and select storage location. (By clicking on the <Add new> entry in an empty table row, a new log is inserted in the configuration. Following this, the log name and storage location for example "Database" and the status at the start of runtime must be selected.
3. Select the tags for logging.



- Inspector window > Properties tab (Properties) > General:
  - Name: Specify the log name
  - Storage location / Storage location: e.g. select "Database".
  - Status: Disable the "Locked" option so that logging is enabled when runtime starts.

As an alternative, the properties of a log can be configured directly in the table of the "Data logs" editor. Columns that are not displayed can be shown using the shortcut menu of the column title.

Properties of the process value

Use the following settings for the WinCC value table (Properties page):

General / Display:

- Inspector window > Properties tab + **Properties** > **General** > **Display**
- Data source: f (t) **trend view_1**
- Mode: Ruler
- Automatic: enable

Layout / Position & size:

- Inspector window > Properties tab+ **Properties** > **Layout** > **Position** & **size** X: 150, Y: 430, width: 1020 and height: 320

Window / Settings:

- Inspector window > Properties tab+ **Properties** > **Window** > **Settings**
- Title style: "**Default**"
- Title: "Blast furnace-W."
- Moveable, Closable: Sizeable: deactivated

Table:

- Inspector window > Properties tab+ **Properties** > **Table**
- Column headers: Show column headers: enabled
- Row headers: Show row headers: activated

Blocks:

- Inspector window > Properties tab+ **Properties** > **Blocks** >
- Length for X value/time stamp under Blocks": set to 28

Columns:

- Inspector window > Properties tab+ **Properties** > **Columns** >
- **Columns**: - Y value and X value/ time stamp: enabled

Toolbar:

- Inspector window > Properties tab+ **Properties** > **Toolbar** >

General:

- Toolbar / Show tooltips
- **Buttons**:
  - Adopt defaults - enable all buttons
- **Status bar:**
  - Inspector window > Properties tab+ Properties > **Status bar** >
- **Status bar - General:**
  - Display status bar/ Show tooltips activated
- Status bar - Elements:
- Show all elements

Log Editing logging tags

The data log in which the logging tags will be inserted needs to be selected in the "**Data logs**" editor.

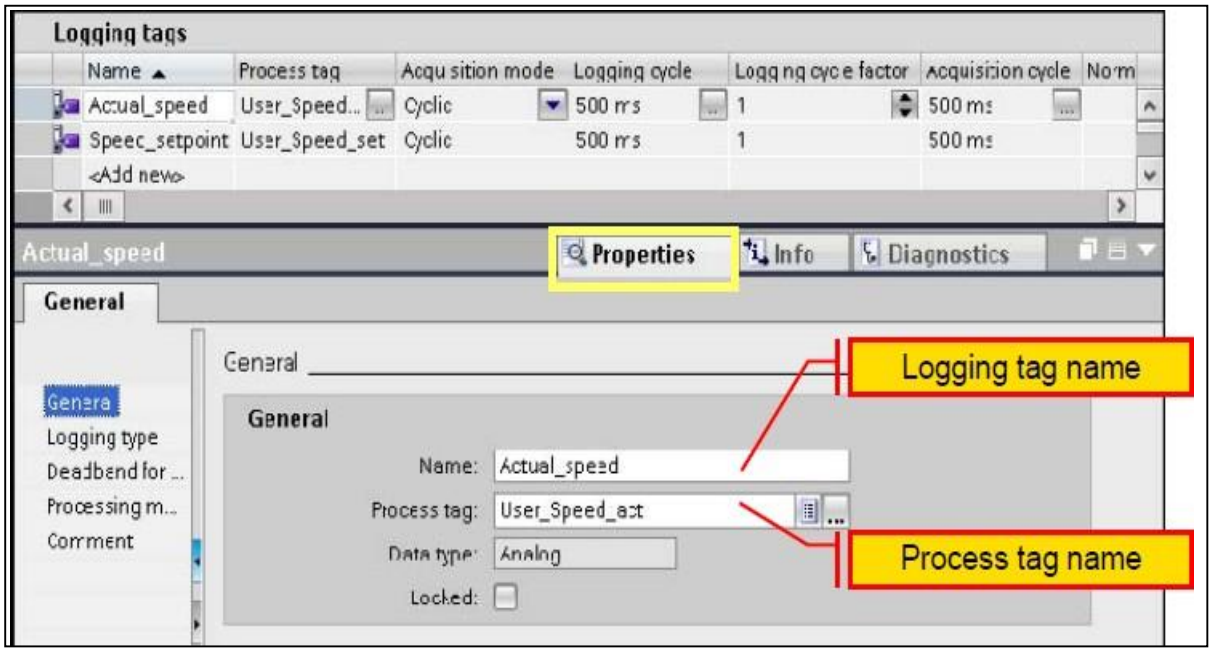
1. Project tree

   > HMI device > Historical data > Data logs tab

2. By double-clicking on "<**Add** >" in the "Name" column of the "Logging tags" table, a new logging tag is created.
3. The logging tag requires further parameters:

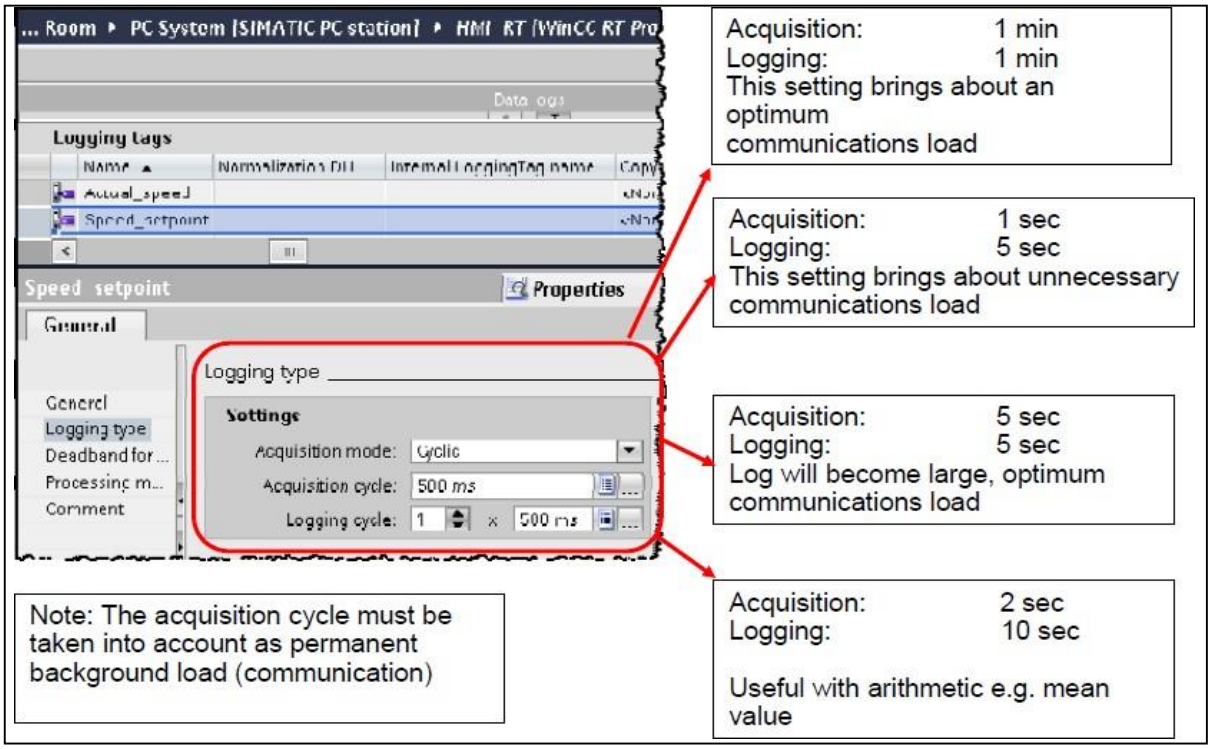
   Properties > General

4. Enter a unique name for the logging tag in the "Name" box.



5. Select the analog tag to which the logging tag will be connected in the "Process tag"

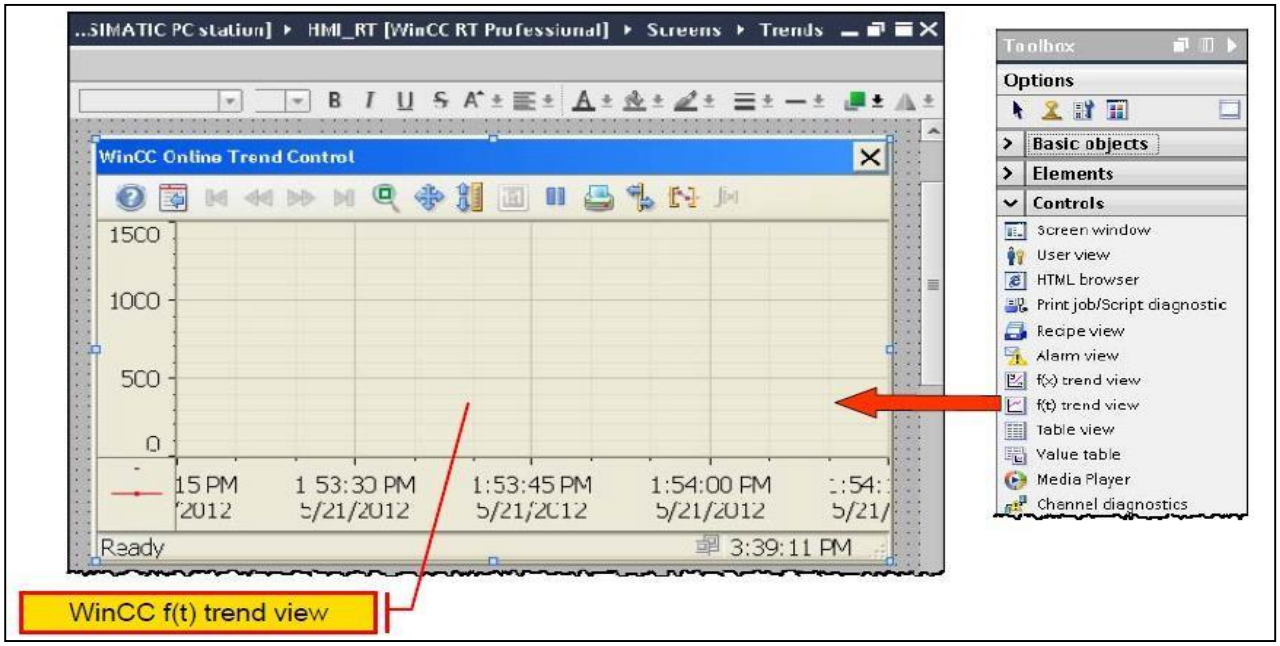
Cyclic selective tag logging:



Event-controlled, continuous process value logging, for example for monitoring a process value within a specific period of time. The system provides various acquisition modes. The default setting for the acquisition mode is "Cyclic" The same value should be used for the acquisition and logging cycle if, for example, "Current" was selected as the processing method so that the system is not subjected to unnecessary load.

Displaying the data log trend view

With the help of the WinCC f (t) trend view, tag values can be displayed graphically in the HMI system. The trend view shows the current or logged values within a particular time



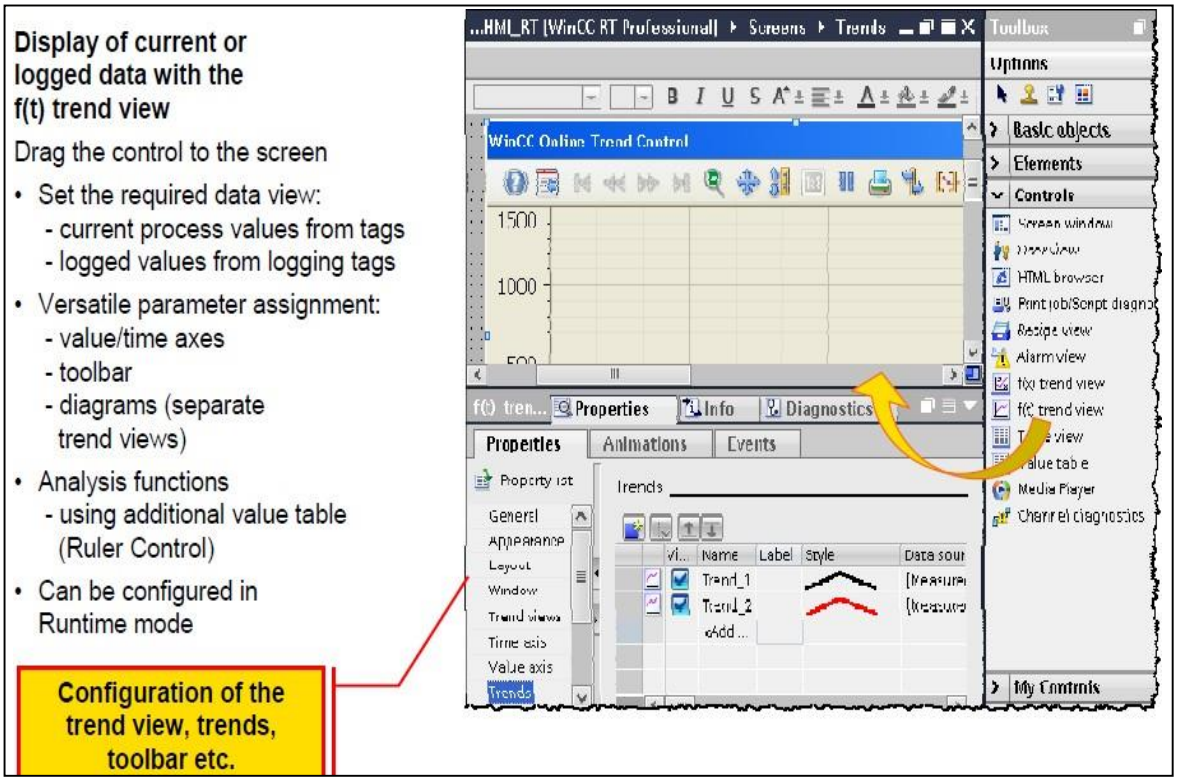
window. When displaying data logs, the operator can shift the time window on the HMI device to view the required information from the log.

The value assignment (**data source)** to the trends is made in the properties of the **WinCC f (t) trend view**

> Inspector window > Properties+ Properties tab > Trends

Several trends can be output in a view regardless of where they are stored (in the same log or in different logs).To achieve this, the source of the data is selected when defining the trend.

To improve clarity when displaying multiple trends, these can be displayed differently due to their names and labeling and different line types and different colors (Style).



Properties of trend view

General / Display:

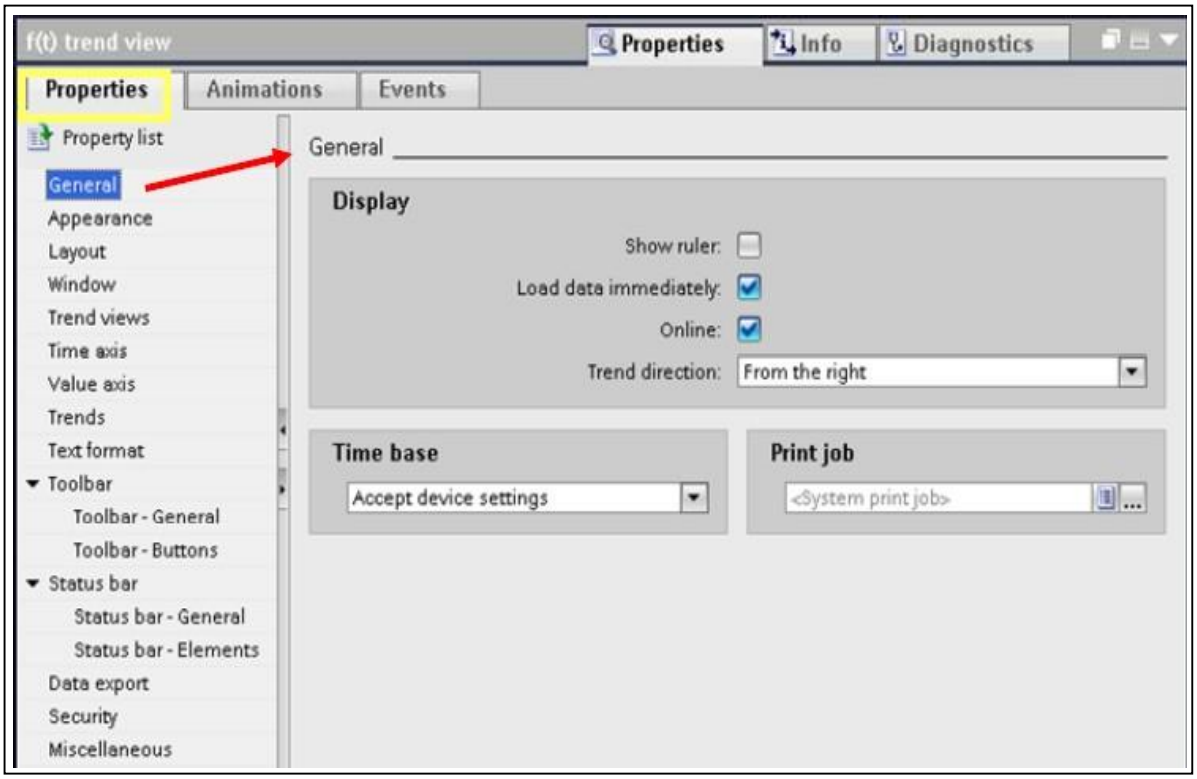
The behavior of the trend view when it is displayed must be defined:

>Inspector window > Properties tab+ Properties > General > Display

Among others, the following settings can be selected for the **trend view**:

- **Show ruler**: Show ruler to display an additional value table (that needs to be configured) below a ruler
- **Load data immediately**: Load data from the data log
- **Online**: Continuous data display

- **Trend direction**: Direction of the displayed trends, for example "From the right"



Time axis:

The behavior of the time axis must be adapted to the particular requirements.

> Inspector window > Properties+ Properties tab > Time axis

The following settings are available:

- **End time**: The time range is specified with a start point and an end point.
- **Measuring points:** The time range is specified using a start point and a number of measuring points

Value axis:

The value axis provides various parameter assignment options.

> Inspector window > Properties tab +Properties > Value axis

The "Value range" of the value axis specifies whether the "Range" of the value axis is adapted automatically or whether the value axis has a fixed start and end value (e.g. 0-1600).Value axes can be arranged to the left and/or right in the **WinCC f (t) trend display**. When displaying more than one trend, it is also possible to set different scales .Several value axes can be used on the left (and/or on the right) in the **WinCC f (t) trend display**. This makes it possible, for example to link five trends for pressure measured values with a value axis for pressures and a further five trends for measured temperature values with a further value axis for temperatures.

Trends:

To display the trends, a data source must be specified:

> Inspector window > Properties+ Properties tab > Trends > Data source

The following data sources are available for a trend:

- "**Logging tags**": The trend view is supplied with values from the data log.
- "**User-defined**": The supply of values for the trend view is user-defined.
- "**Tags":** The trend view is supplied with the values of a tag.

Buttons:

> Inspector window > Properties+ Properties tab > Toolbar > Buttons

The most important buttons when inserting the WinCC f(t) trend view already exist. Additional buttons, for example for data export or connecting/disconnecting backup can be enabled if required.

Recipe Management

Recipe

Recipes are a collection of data that belongs together, e.g. Machine parameter settings or production data. A recipe has a fixed data structure. The recipe structure is defined during configuration. A recipe contains recipe data records. These differ in terms of their values, but not their structure. For SIMATIC Panels and WinCC Runtime Advanced, recipes are saved on the HMI device or on an external storage medium. For WinCC Runtime Professional, they are saved in an SQL database. The various data records for the recipes can be created as early as the engineering phase of the project. This means that they are available directly after the transfer of the configuration in Runtime. This data can function as the initial basis for an optimization during runtime or it even can be the result of such a process. During operation, the data records can be conveniently created, deleted, or modified across different views. In addition, recipe data can be imported into or exported from Runtime Configuration.

In addition to the option of using recipe data exported from the HMI device on other HMI devices, it can also be re-imported into engineering. This data can then either be logged with the project or reused for other projects. Configurable screen objects are used for displaying and managing the data at runtime.

Using recipes

Recipes can be used in the following situations:

- Manual production

The desired recipe data is selected and displayed on the HMI device. The data can be modified and saved on the HMI device as needed. The recipe data is then transferred to the controller.

- Automatic production

The control program initiates the transfer of the recipe data Between the PLC and HMI device. The transfer can also be started from the HMI device. Production is thus implemented adapted. It is not essential to display or modify the data.

- Teach-In mode

Production data is optimized manually on the system, e.g. Axis positions or filling volumes. The values thus determined are transferred to the HMI device and saved in a recipe data record. You can then transfer the saved recipe data back to the controller at a later date.

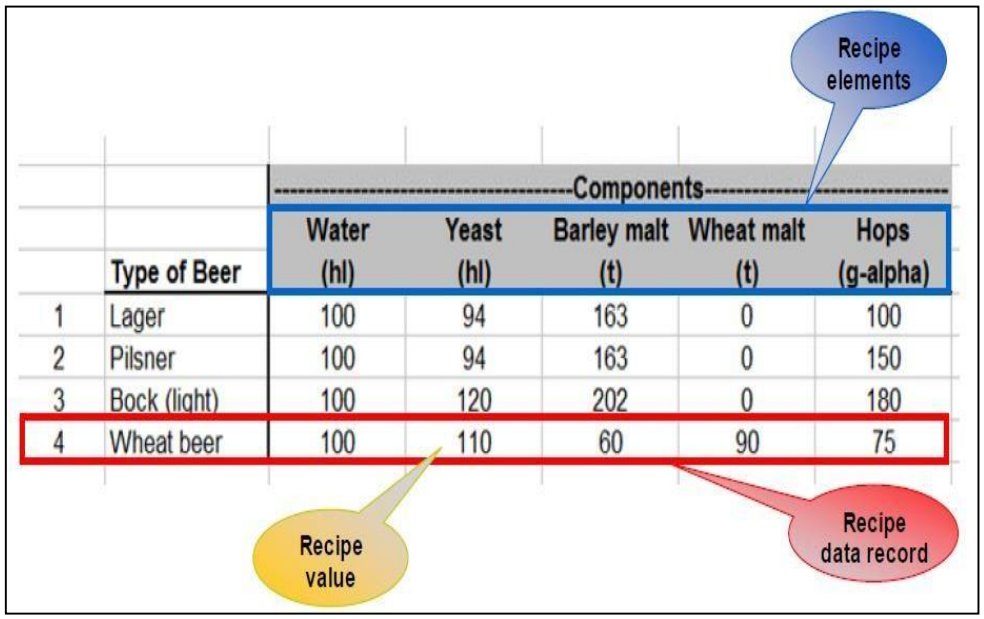
Structure of recipe

Recipes represent a grouping of data that is structured in the same way

Recipe components

- **Recipe name**

  In the HMI device, all recipe data records of a recipe are collected in a database and saved under the recipe name in the database.



- **Recipe data records**

  These are the contents of the recipe. Each recipe data record contains the component values of an individual variant, i.e., one recipe data record is saved per variant.

- **Recipe elements**

  These form the recipe structure. Recipe elements are the components of a data record.

Comparison

A recipe can be conceived as a table saved in a file. The table header is the recipe structure and consists of the individual recipe elements, i.e., each column is a recipe element. The rows of the table are the individual recipe data records (variants).

Example

In this example, the production data of a brewery is stored in a recipe. The following components are required for beer production: water, yeast, hops, and malt. Different quantities of these are required for the individual products. One recipe data record is saved for each beer type that can be produced.

This results in the following recipe components:

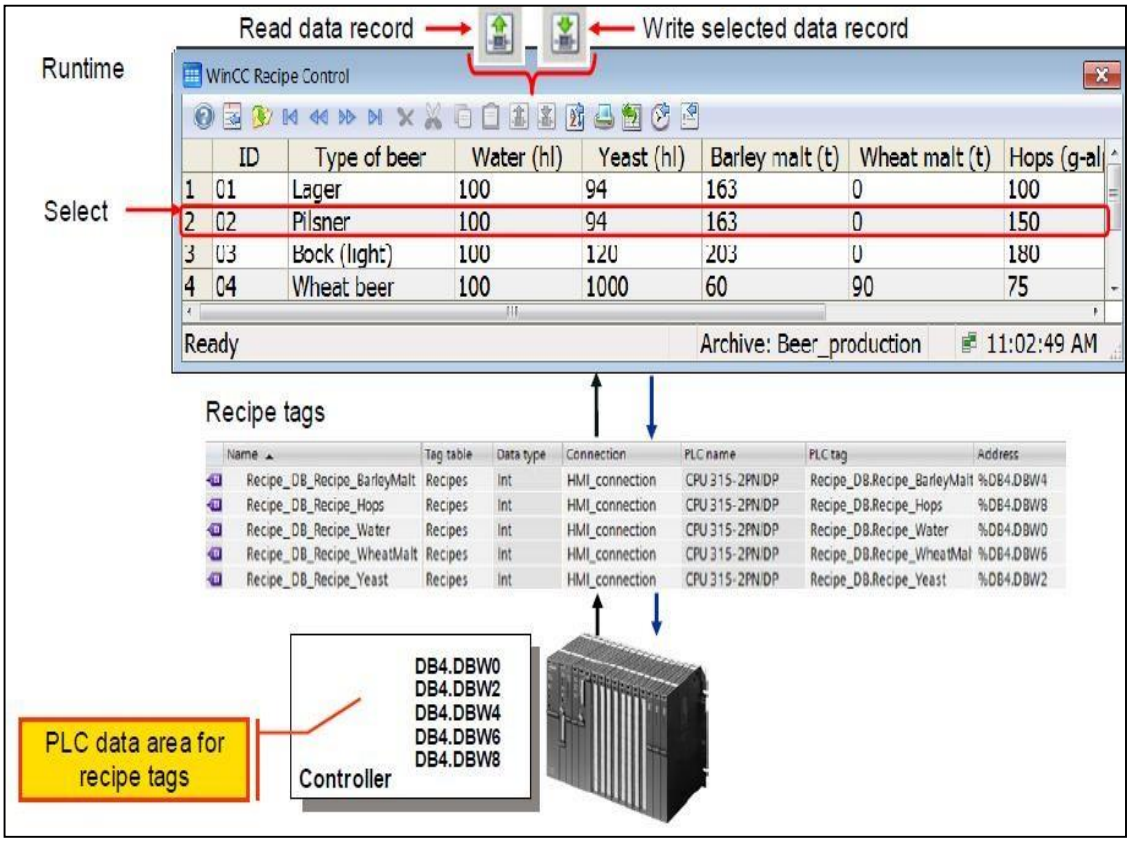
- **Recipe name** = Beer production
- **Recipe elements** = 5 recipe elements Water, yeast, barley malt, wheat malt, and hops

- **Recipe data records** = 4 recipe data records for the products: lager, pilsner, bock, and Weiss beer

Communication with the controller using tags

Recipe data is always transferred between the HMI device and the connected controller record by- record using configured recipe tags (process tags). Each recipe element is assigned a control tag. These recipe tags are then the communication area for exchange of a recipe data record between the HMI device and controller.

Settings for data transfer



The following options are available:

No data transfer

The recipe is not used for transferring recipe data to the controller but as a data memory and user log: You store values that belong together in data records, which you can access in various ways.

In the simplest case an operator switches between the recipes in a recipe view. The respective recipe data is loaded into the recipe tags (internal tags) and processed further there.

"Tags" communication type

Recipe tags transfer the recipe values to the controller. Conversely, the values from the controller are written to a recipe data record via recipe tags. In so doing, the data exchange is started by the operator using the buttons (Read tag/Write tag) in the recipe view or controlled

via control tags. The following jobs are started automatically by making the desired entry in the corresponding control tags.

- Write data records to the recipe tags thereby transferring them to the controller
- Read data records from the recipe tags thereby retrieving them from the controller
- Delete data records
- Create new data records

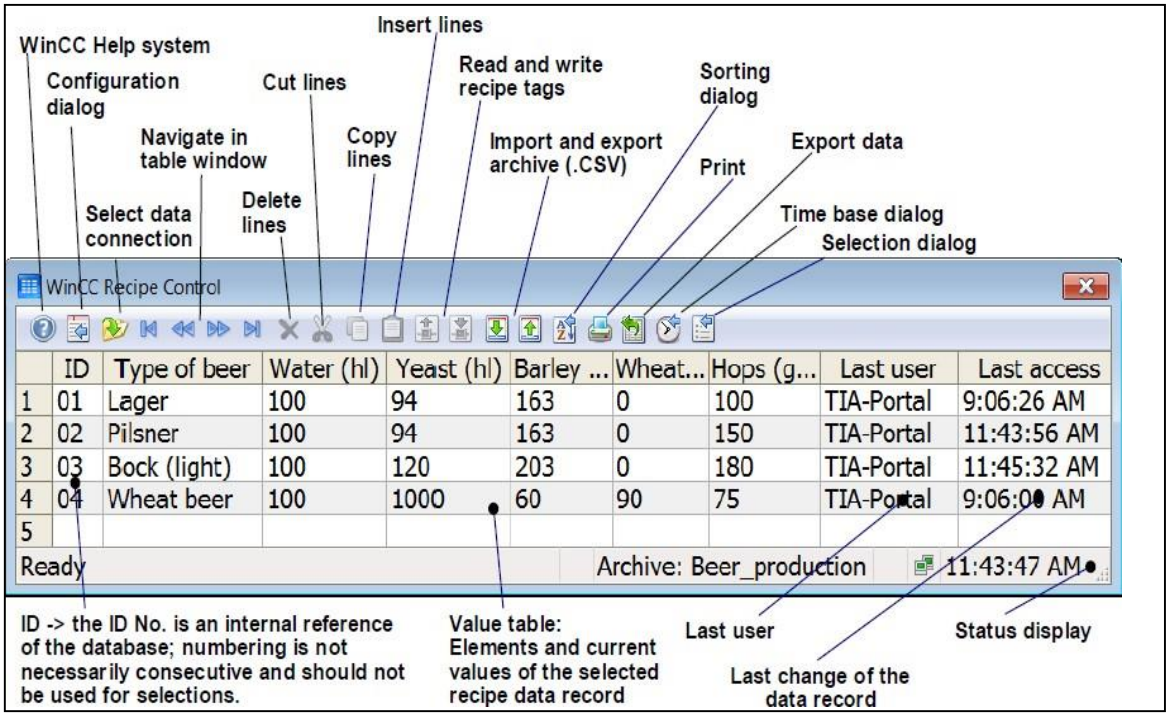
Note:
A complete data record is always transferred in the "**Tags**" communication type. Transfer of recipe data from a recipe query is not possible.

Table representation of WinCC recipe view:

The recipe view has a fixed structure, which can be configured during the configuring step. The advantage is that configuration is simple and fast. However, the basic structure of the recipe view cannot be changed. It is only possible to show/hide display elements (properties such as table columns, toolbar buttons, and the status bar) and change the size.

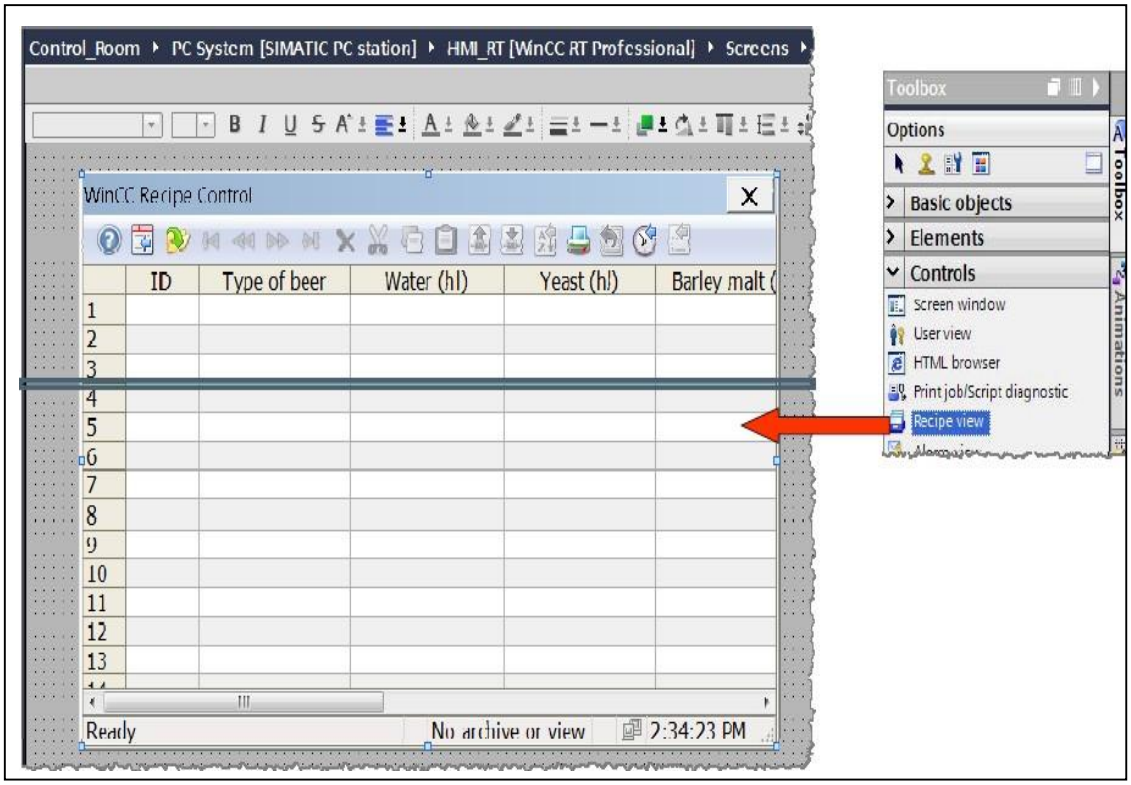
The recipe view is a configurable object that supports the handling of recipe data in the HMI device:

- Recipe display and selection
- Recipe entry and modification



	ID	Type of beer	Water (hl)	Yeast (hl)	Barley ...	Wheat...	Hops (g...	Last user	Last access
1	01	Lager	100	94	163	0	100	TIA-Portal	9:06:26 AM
2	02	Pilsner	100	94	163	0	150	TIA-Portal	11:43:56 AM
3	03	Bock (light)	100	120	203	0	180	TIA-Portal	11:45:32 AM
4	04	Wheat beer	100	1000	60	90	75	TIA-Portal	9:06:00 AM
5									

Ready Archive: Beer_production 11:43:47 AM



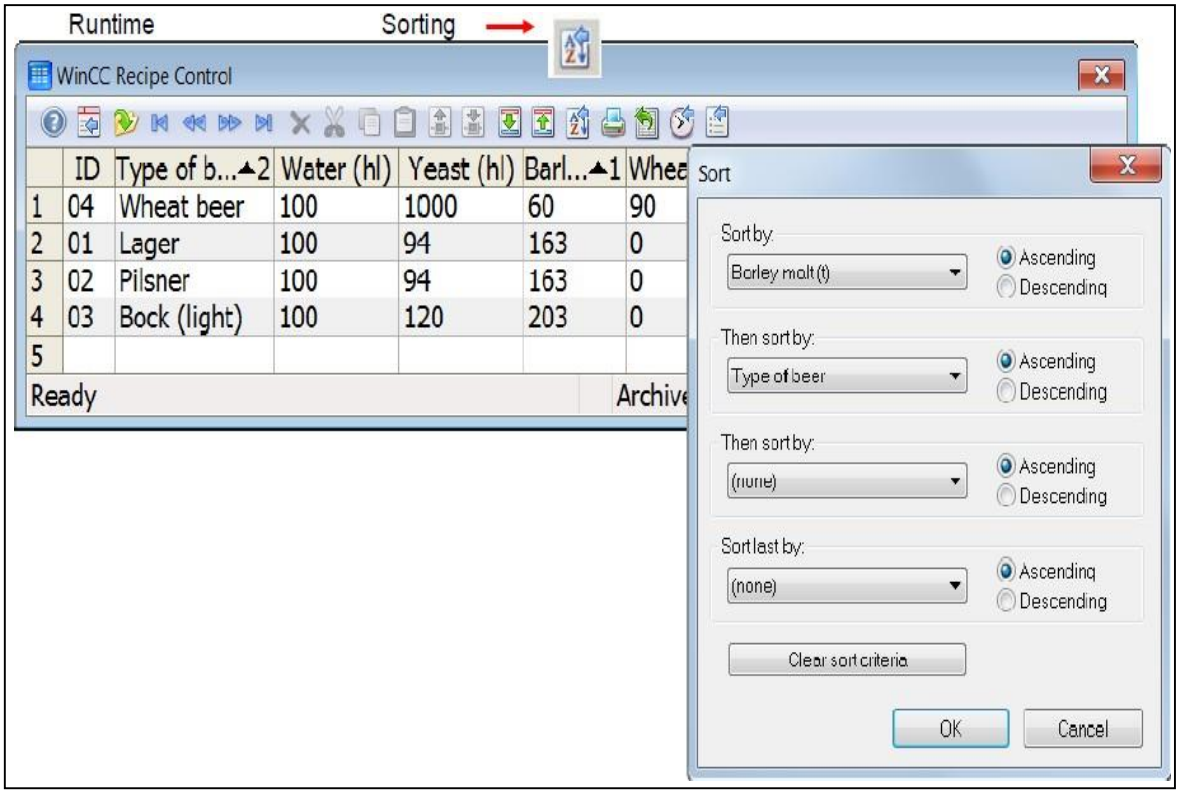
The recipe view is located in the "**Controls**" section of the **"Toolbox**" task card. Depending on the application, the recipe view can be adapted by means of its properties. The most important properties of a recipe view can be found in the inspector window in the sections:

- General
- Appearance
- Columns
- Symbol bar
- Label

These properties should always be checked and configured.

Sorting in the recipe view

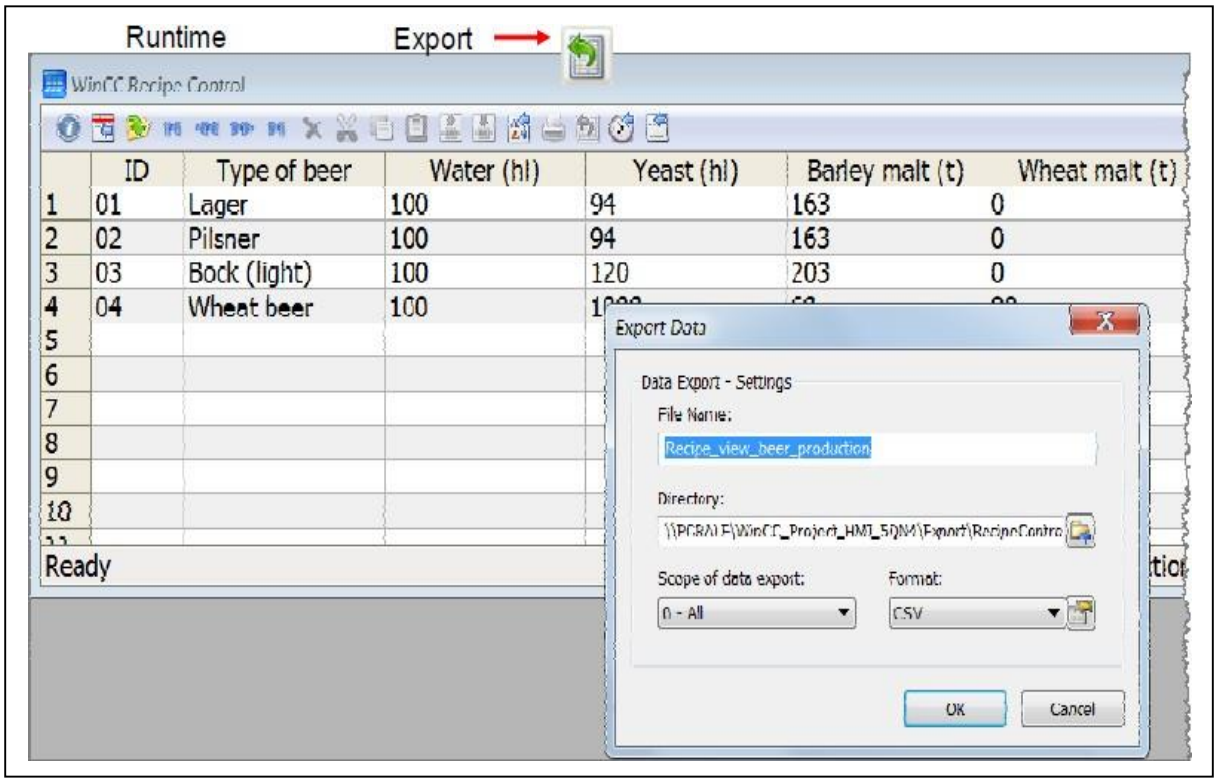
Opens a dialog box for entering a user-defined sorting of the displayed columns. We can sort the recipe as per ascending and descending order.



Exporting in the recipe view

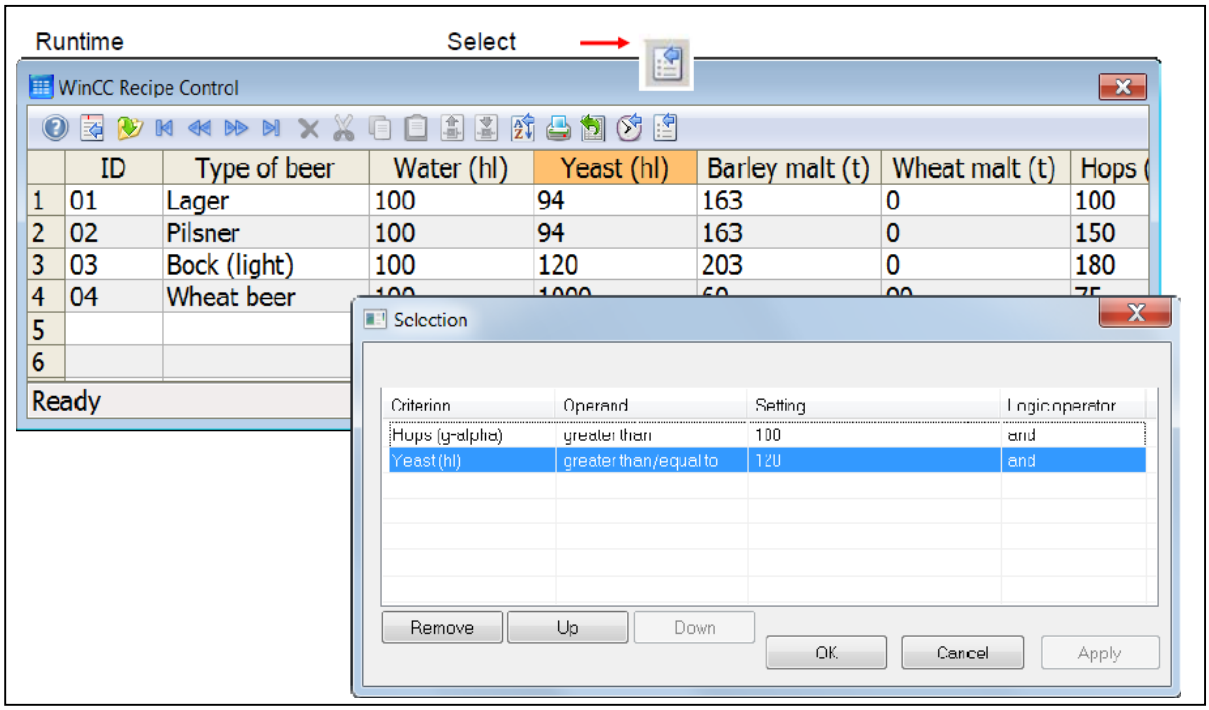
Export data

This button is used to export the selected recipes. The export is made with the current runtime data to a CSV file. The export settings are selected in the "**Export data**" dialog.



We can export the data in any format like excel file or CVV file or Trend view. Below is the screen how exporting for recipe view looks.

Selection dialog

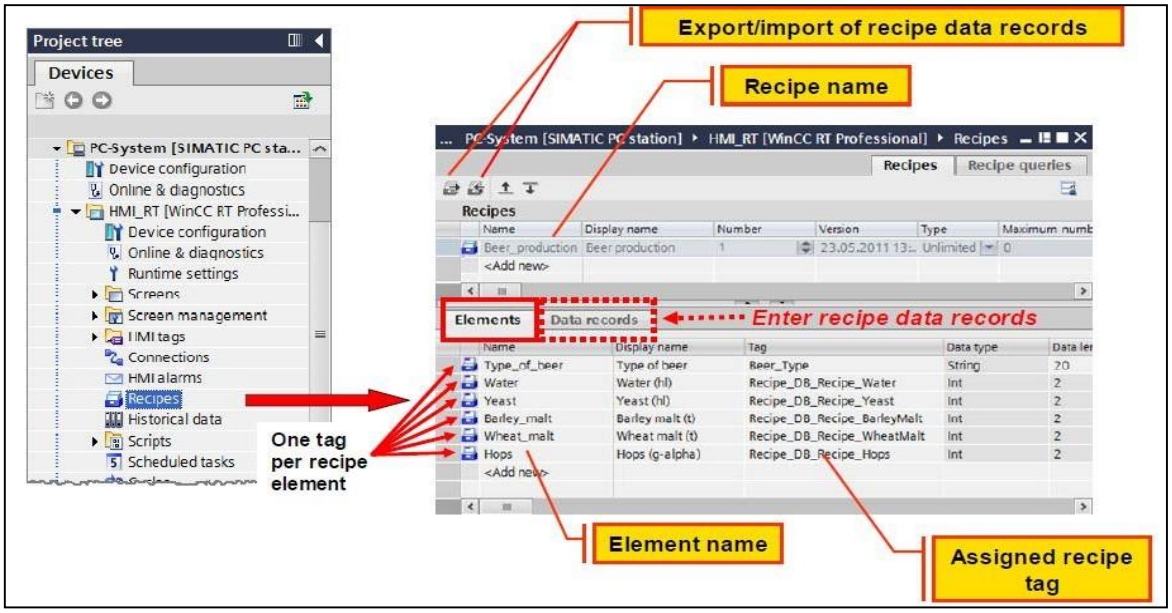


Selection criteria can also be entered in the **SQL** database language. The **SQL** string can be transferred, e.g., via a tag or script for the "**Filter/Filter SQL** statement" property of the recipe view.

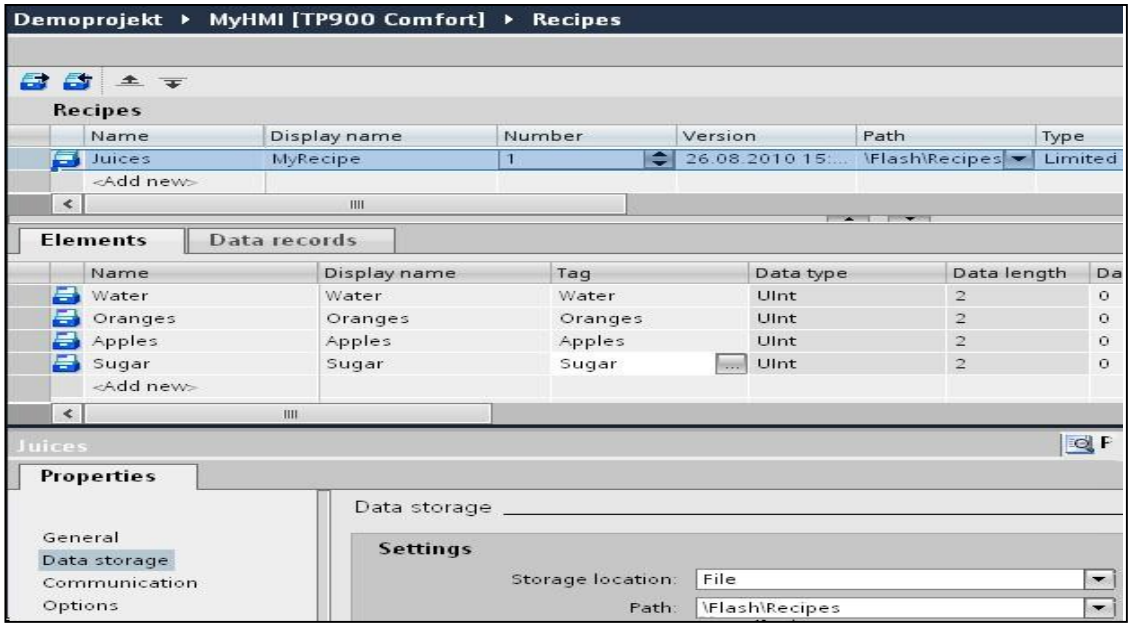
Procedure for configuring a recipe

The following steps must be performed when configuring recipes:

1. Project tree > **HMI device** > **Recipes**.



2. By double-clicking on "<**Add new** >" in the "Name" column of the “recipe” window, new recipe will be created.



3. Now select “**Elements**” in below tab (i.e. shown in figure) and in place of name enter elements for your recipes (for example: for TEA you needs elements are Water, milk, tea powder).
4. Now add “**display name**” whatever you would like to display in your recipe window in SCADA
5. Assign appropriate **Tags** for all the elements
6. You Can also use “**data record**” function to record your recipe data.
7. Take one recipe window from **toolbox** > **control.**
8. Go to its **properties** > **general** > Assign your recipe.

Recipe editor

Recipes are created, configured, and edited in the **"Recipes"** editor. Recipe data records can also be entered in the "Recipes" editor.

The "**Recipes"** editor is opened by double-clicking in the **Project tree**.

1. Name (recipe name):

   - Identifies the recipe uniquely within the project
   - Key words or reserved words of the SQL database language must not be used for the recipe name (see Section "SQL Keywords" in the TIA-Portal information system).

2. Display name:

   - Appears on the HMI device, e.g., in the recipe view
   - Multilingual configuration is possible

3. Recipe number:

- Identifies the recipe uniquely within the project

4. Version:

   - Identifies the date and time of the most recent change made to the recipe.

5. Name (recipe element name):

   - Identifies a recipe element uniquely within the recipe
   - Assign descriptive names or designations that you can explicitly associate with the element, such as axis designations on a machine, or ingredients such as "yeast".

6. Display name (recipe element display name):

   - Appears on the HMI device, e.g., in the recipe view
   - Multilingual configuration is possible

7. Tag:

   - Each recipe element is assigned a recipe tag in which the corresponding value for the recipe data record is stored.

8. Default value:

   - This is used as the default entry when a new recipe data record is created.

9. Tooltip:

   - Help text on the recipe element that provides information to the operator on the HMI device.

Entering of recipe data

Recipe entry:

Recipe data records can be entered at the time the recipes are configured (in the "Recipes "editor). These recipes are then transferred to the HMI device along with the configuration.

Export/import of recipe data records:

The export function can be used to export individual recipe data records or the entire recipe. In the case of import, all recipe data records of the export file are imported. For importing, it can be selected whether existing recipe data records in the project are to be overwritten.

Setting recipe properties:

Recipe properties for a recipe are defined in the properties tab of the Inspector window.

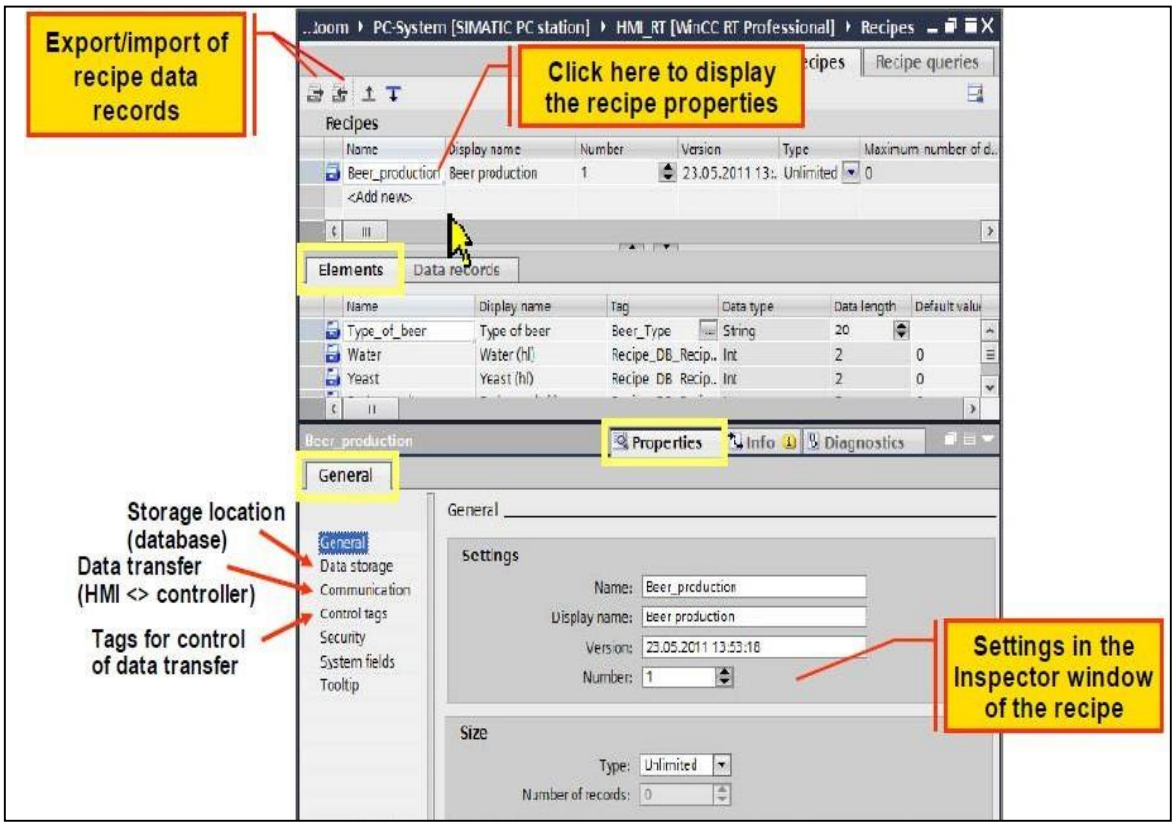
Recipe property – Communication:

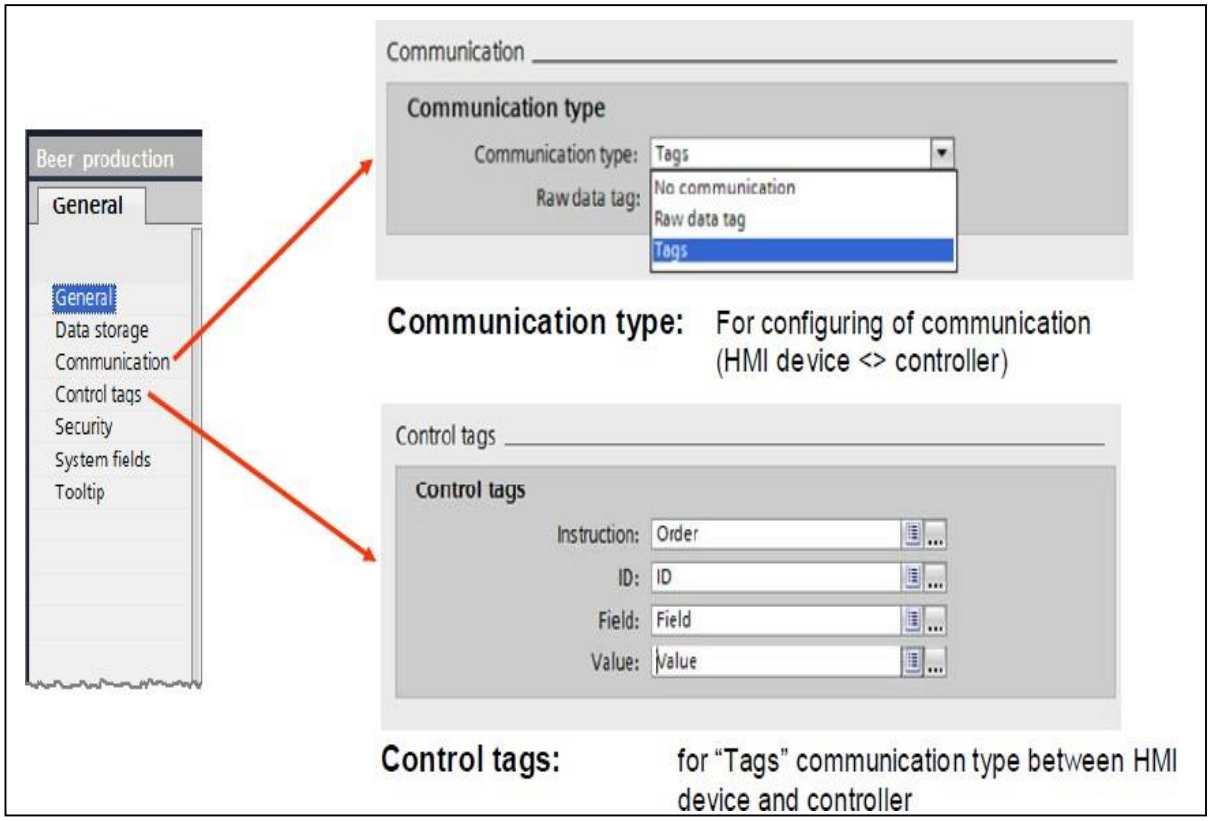
In WinCC Professional, recipe data is always stored in the database of the HMI device. Data are transmitted in two ways between recipes and the controller By means of recipe tags via user-actuated buttons in the recipe view or with the aid of control tags By means of message frames with the aid of raw data tags

"Tags/control tags" communication type:

In this case, the data exchange is started by the operator using the buttons (Read tag/Write tag) in the recipe view or organized via control tags. The control tags must be created separately for each recipe. A complete recipe data record is transferred to the recipe tags and vice versa via control tags. The recipe tags (process tags) have a communication connection to the controller.

Communication type **"Raw data tag"** The controller and the HMI device exchange raw data tags as message frames with a fixed structure.





Raw data tags offer the following advantages:

- Exchange of multiple data records between controller and HMI device with one job
- Transfer of portions of a recipe data record
- Use of the same raw data tag for various recipes

Configuring the recipe view

The recipe view has a fixed structure, which can be configured during the configuring step. The advantage is that configuration is simple and fast. However, the basic structure of the recipe view cannot be changed. It is only possible to show/hide display elements (properties such as table columns, toolbar buttons, and the status bar) and change the size. The recipe view is a configurable object that supports the handling of recipe data in the HMI device:

- Recipe display and selection
- Recipe entry and modification

The recipe view is located in the "Controls" section of the "Toolbox" task card. Depending on the application, the recipe view can be adapted by means of its properties. The most important properties of a recipe view can be found in the Inspector window in the sections:

- General
- Appearance
- Columns
- Symbol bar
- Label

User Administration

Introduction

The user administration controls access to data and functions on the HMI device during runtime in order to protect the HMI device's data and functions from unauthorized manipulation. The user administration function is configured in the engineering system for this purpose, and transferred to the HMI device. Not all of the functions of a machine or plant may be carried out by every user. Many tasks require special qualifications or are restricted by the process to special user groups. Carrying them out requires rights that are assigned to special user groups and users. WinCC supports the user in creating and managing user groups and users and in assigning the required rights in engineering and during runtime.

The separation of authorizations and users allows efficient user administration with reduced engineering effort. In the engineering system, user groups are defined which group together the configured authorizations in a task-oriented way. For example, the user group "Production planning" can change recipe data records, set system parameters, and log process values. The necessary authorizations are assigned to the corresponding objects in the project.

The actual user can then be accepted in the user administration with a user name or user ID and password even during operation and then be assigned to a user group without any further changes to the configuration. In this way, the unambiguous identification of the users – e.g. For Audit Trails – can be managed with minimal engineering effort.

All local operator stations are included in the user administration, as well as the standard and Web Navigator or Data Monitor clients for a SCADA system on the basis of WinCC Runtime Professional. If system-wide user administration is required, the SIMATIC Logon central user administration system can be activated as of Comfort or Multi Panels. In this case, SIMATIC Logon takes over the user administration of the local operating systems in cooperation with Windows. If communication to the central component SIMATIC Logon is interrupted, the users are then only checked locally on the HMI system. Depending on the target system, SIMATIC Logon can be installed on the HMI system itself or on another remote PC in the network or a domain controller. When SIMATIC Logon is used, the use of a chip card reader for user authentication is also supported.

Structure of the user administration

The user administration is divided into:

- Administration of users, user groups and authorizations
- Assignment of the corresponding authorization to the individual configuration objects

Operator authorization:

Authorizations describe the rights to access an object or to carry out a specific action on the object (e.g. "Change input value", "Select screen", "and Edit recipe").

An authorization can be assigned to each accessible object (IO fields, buttons etc.).

User group:

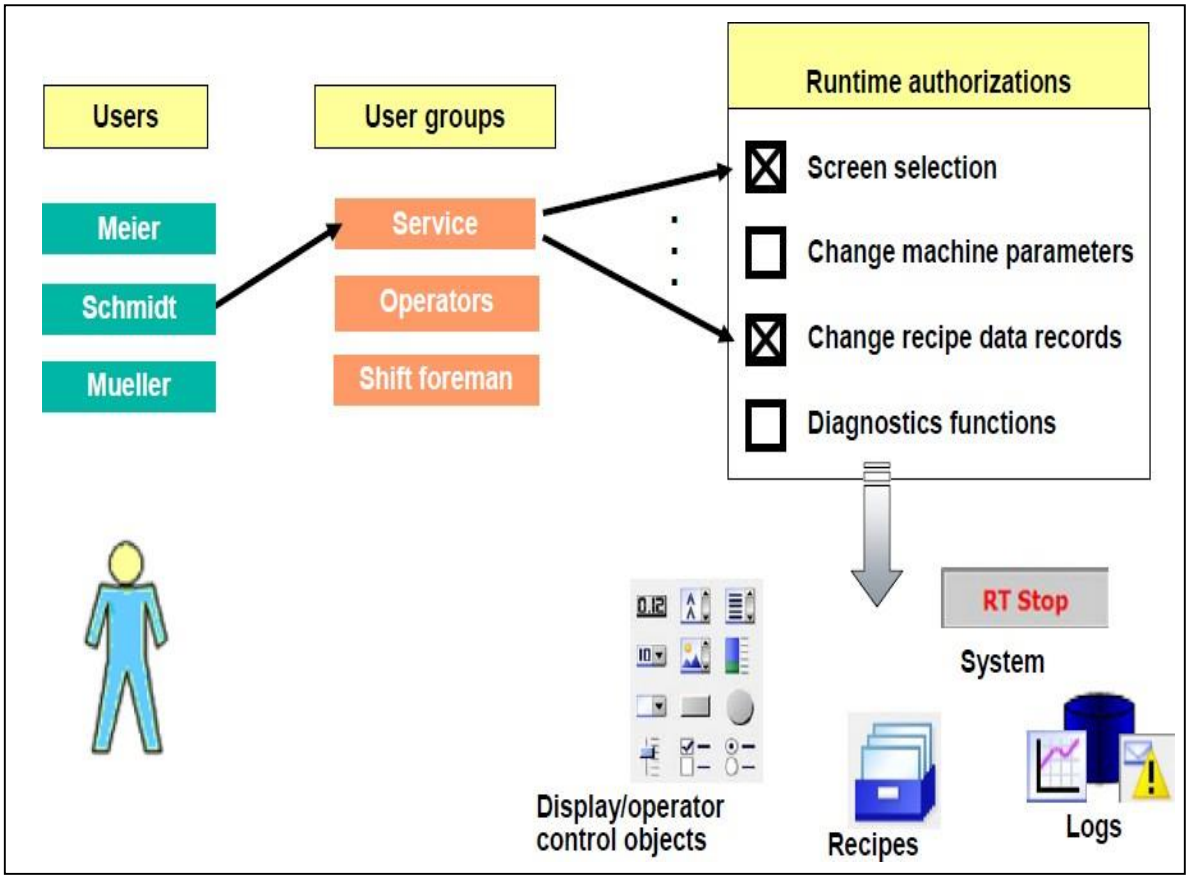
A user group combines authorizations means all users within a group have the same authorizations. Different operator views can be mapped in user groups, e.g.

- Organizational view: commissioning engineers, operators, shift I, shift II
- Technological view: axis control, tool changers, Plant North, Plant South

User:

"Users" is the generic term for operators. Each operator is assigned to an associated user group and thus receives its authorizations. Operators may therefore only access objects for which they possess the authorization.

A user then logs on at the HMI device with the user name and associated password.

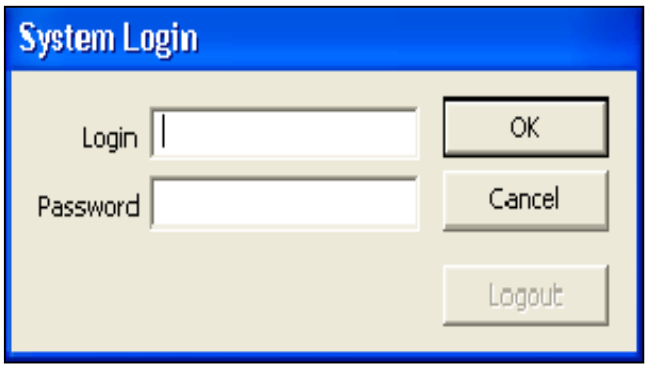


How to Access Protection Works

A user accesses an object (e.g. clicks a button). WinCC checks in runtime whether access to the object is protected.

- If access is not protected, the function configured for the object is executed.
- If access is protected, WinCC determines the user group to which the logged-on user belongs. The authorizations of the user are derived from this.

If the logged-on user possesses the necessary authorizations for the object, the configured function is executed. Otherwise, a "logon dialog" automatically appears for the user to log on.



Login dialog appears automatically:

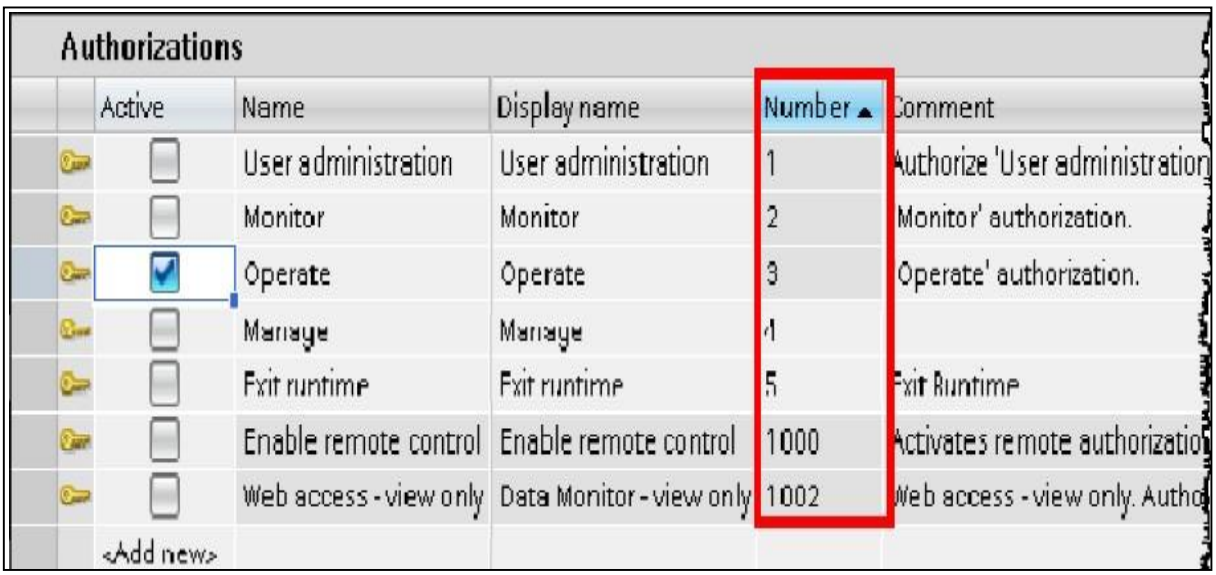
If a protected object has been accessed and the logged-on user does not have the required authorization or if no user is logged on, the "**Login dialog**" is automatically displayed. After a successful logon, WinCC once again checks whether or not the logged-on user has the necessary authorization.

- If it is, access to the object is enabled: When the user clicks the button again, the configured function is executed.
- If not, a corresponding system message is displayed indicating that the user does not have the required authorization. Although the user is logged on in runtime, the configured function is not executed.

Steps in configuring user administration

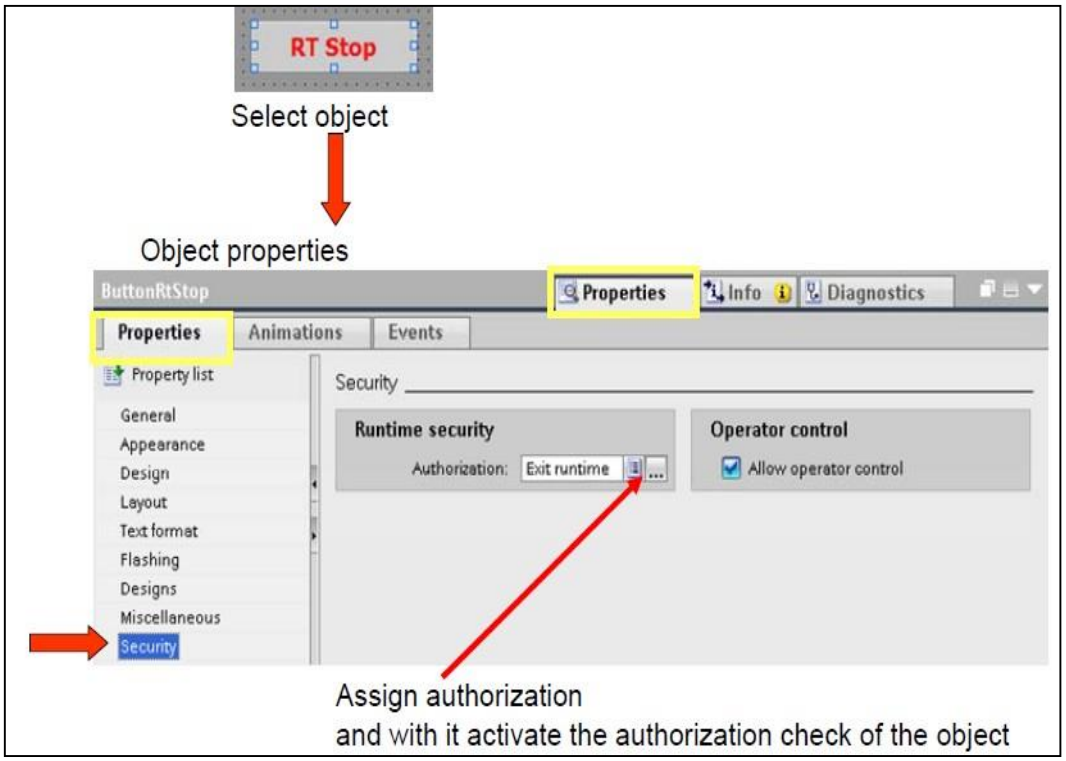
1. Structure authorizations

   Find out which groups of people and authorizations are required for access protection.



Example: *create three screen (1 for manager, 2nd for engineer and 3rd for operator, manager can access all screen, engineer can access 2nd and 3rd screen and operator can only access 3rd screen) and all screen only can access by individual password.*

2. Go to Project tree > HMI device > User administration > "User groups" tab
3. Create 3 groups i.e. 1. Admin, 2.operator, 3.engineer.
4. In below tab "Authorization" give Authorization for individual group(i.e. For manager assign all access for engineer give authorization of operating and monitoring and for operator give authorization of monitoring only.
5. Also Assign **display Name** for your reference
6. Now go to "**user"** and create 3 user (i.e. **Manager, engineer** and **operator**) and **passwords** for each.
7. Now go to below tab "**group**" and assign individual group for individual users.
8. Now go to screen and take "**user administration icon**" from **toolbox** > **control** > **user administration**.
9. Also assign security to all buttons from which you will access your screens
10. ***Click buttons > properties > security > click authentication > assign user.***



(Who can access screen from this button), follow same procedure for all buttons that you have created and assign individual protection for each.

Creating Authorizations

The configured authorizations are initially only names without reference to a particular function. This changes only when an authorization is assigned to a configured object. The names can be assigned as desired (free text).The name should be oriented on the function to be executed. A consecutive number is automatically assigned to each authorization. This is the unique identification feature. The authorizations in the range 1000 to 1099 are system

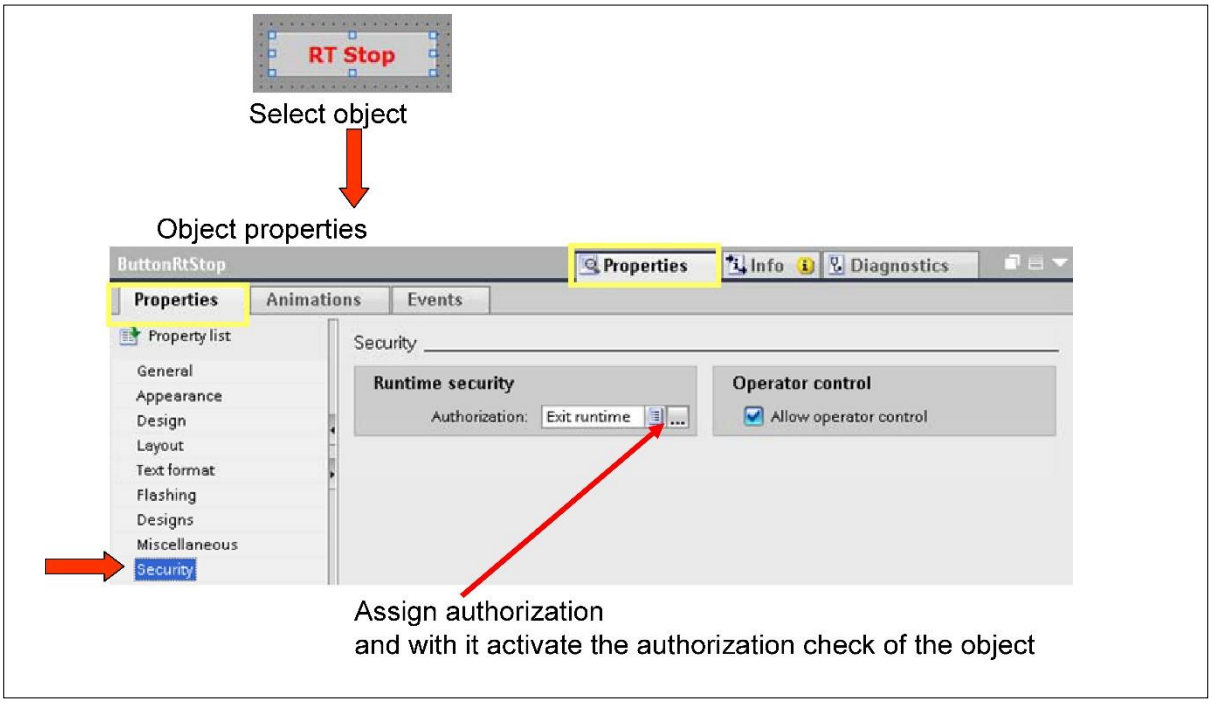
authorizations and cannot be changed by the user. Please only define authorization names of an HMI device once to avoid assignment errors.

The authorizations "**User administration**", "**Operate**", "**Monitor**", "**Remote control**" and "**Web access** - view only" are predefined and are always present. In contrast to their numbers, the names of the authorizations "**User administration**", "**Operate**", "**Monitor**" can be changed.

The Runtime authorization "**User administration**" (authorization number = 1) is always automatically assigned to the "**User view**". The authorizations "**Operate**" and "**Monitor**" are initially unused.

A new authorization is created by clicking in the next empty line in the "**Name**" column. The name of the new authorization (e.g. "**Exit RT**") can then be entered. A detailed description of the authorization can be entered under Comment. All entries can also be made in the Inspector window

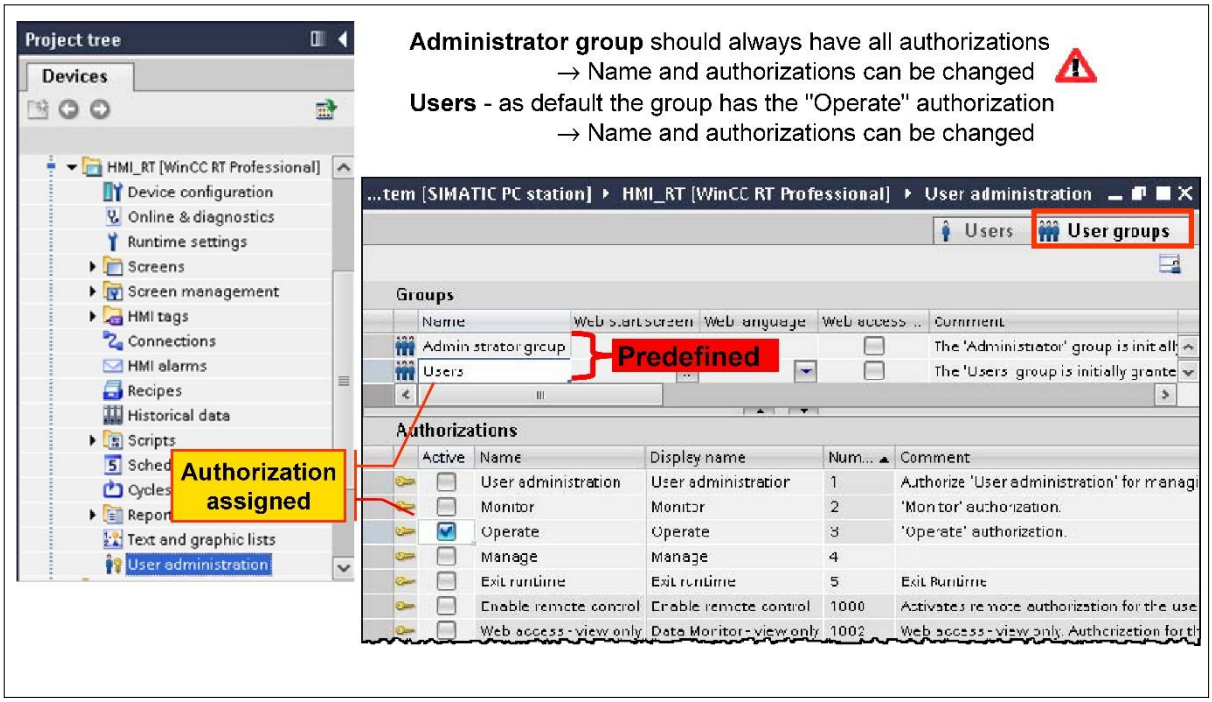
Assigning authorizations to objects



Each accessible object has the section "Security" in the properties. If an authorization is configured and activated there, access protection is active during Runtime.

Configuring user groups

The authorizations are combined in user groups. The "Groups" table shows the existing user groups.



The group names must be unique within an HMI device. A consecutive number is assigned automatically by user administration for the user group.

User group display name

The "**Display name**" of the group is language-dependent (can be translated like a display text) since the users can also be administered and edited on the HMI device in runtime and each user must be assigned to a user group. The display name in the corresponding language is then shown for the user group on the HMI device in runtime.

The "Administrator group" and "Users" groups

These are already predefined and always present. Initially, the "**Administrator group**" has all predefined authorizations.

Newly created customized authorizations are not assigned automatically. The assignment of authorizations can be modified as necessary. This means that users in the "Administrator group" do not automatically have unlimited access to all operator control functions on the HMIdevice → projectable.

Only the authorization "Operate" is assigned as standard to the "Users" group. However, the assignment of authorizations can be changed as required. A group name and user name can only be assigned once.

Create new group

A new group can be defined by clicking in the next empty line in the "**Groups**" table. The name of the new group (e.g. "**Service**") can then be entered. In the Comment field you can enter a detailed description of the user group. All entries can also be made in the Inspector window.

Processing of user groups / assignment of authorizations

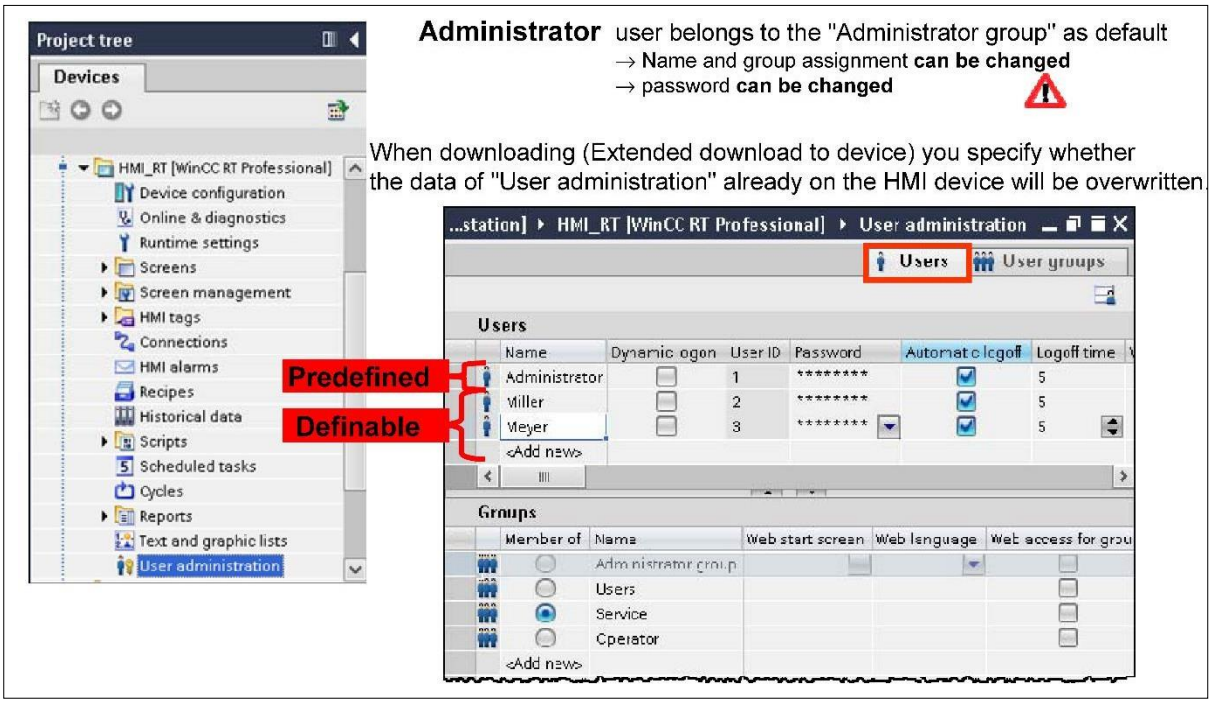
If a user group is selected in the "**Groups**" table, the "**Authorizations**" table shows the authorizations assigned to the selected user group. By clicking on the check box in the "**Activ**e" column of an authorization, this can be assigned to the selected group or canceled.

Users can also be copied and functions can be assigned to the user group in advance that are then the defaults when setting up a new user.

Configuring user:

Users can already be created during the configuration phase and loaded on the HMI device along with the configuration.

Users created during the configuration phase are always transferred as a completely new user data set to the HMI device, and overwrite the users present on the HMI device. This means that you must take care when updating the configuration that the users are not transferred again, otherwise all modifications made by operators since the last configuration transfer will be lost!



When downloading (Extended download to device), you specify whether the data of "User administration" already on the HMI device will be overwritten.

User "Administrator"

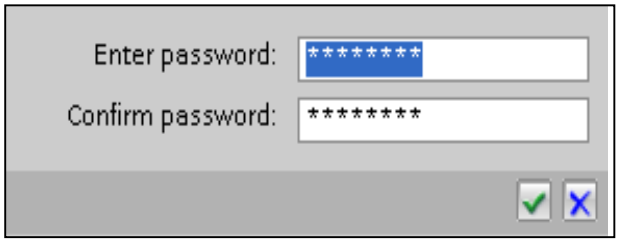
The "Administrator" user is predefined and is always present. This user is assigned as default to the "Administrator group".

Newly created customized authorizations are not assigned automatically to the "Administrator group". The group assignment can also be modified as necessary. This means that the "Administrator" user does not automatically have unlimited access to all operator control functions on the HMI device.

The default password is the word "administrator".

Password

Both the name and a password are used to identify a user.



- Minimum length password = 6 characters
- Maximum length of user = 24 characters
- Invalid special characters: " \ “and " ’ "

The password must be entered twice when configuring users.

Logoff time

If no actions are carried out by the operator within the set logoff time, the logged-on user is automatically logged off by the HMI device, and no operator is then logged on with the HMI device. The automatic Logout can also be disabled.

- Standard logoff time 5 minutes

Create user

A new user can be set up by clicking in the next empty line in the "**Users**" table. The name of the new user (e.g. "Smith") can then be entered.

A user group is not automatically assigned to a newly created user.

A password is also not assigned automatically. This must always be configured!

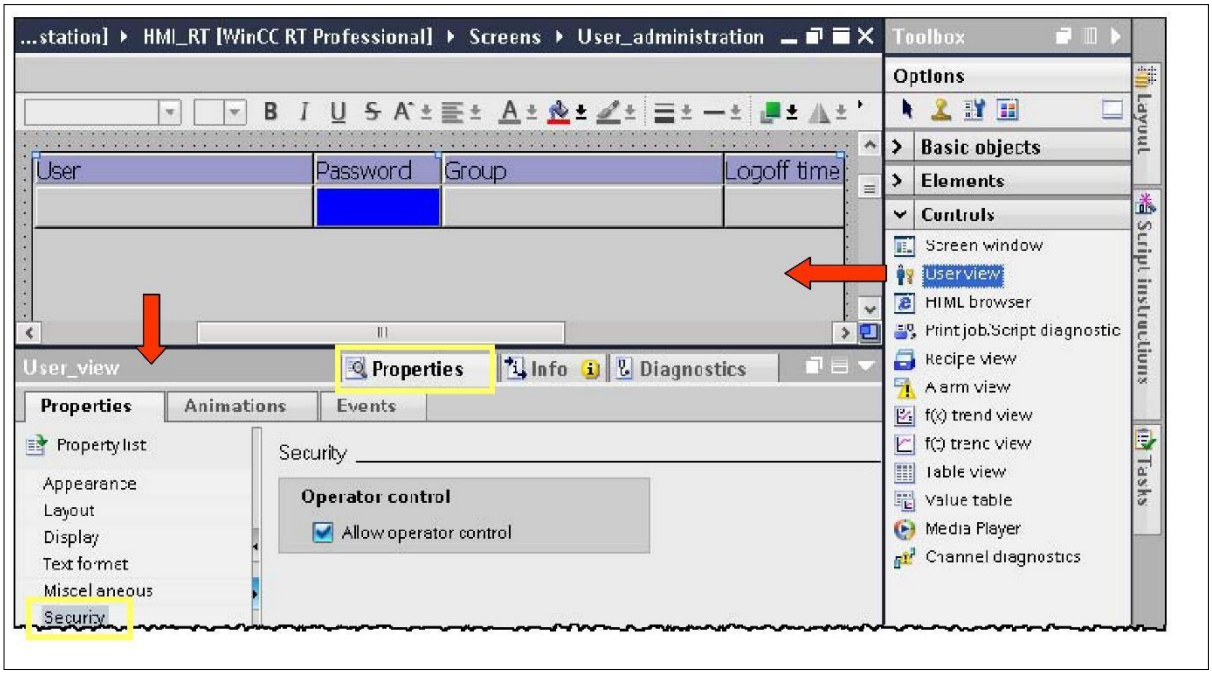
- User name: max. 24 characters

Editing users / assignment to group

If a user is selected in the "**Users**" table, the "**Groups**" table shows the group to which the selected user has been assigned. By clicking the radio button of a created group, the selected user can be assigned to it.

Configuring the user view

Users and passwords are administered using the "**user view**" in runtime. This is located on the "**Toolbox**" task card in the "**Controls**".

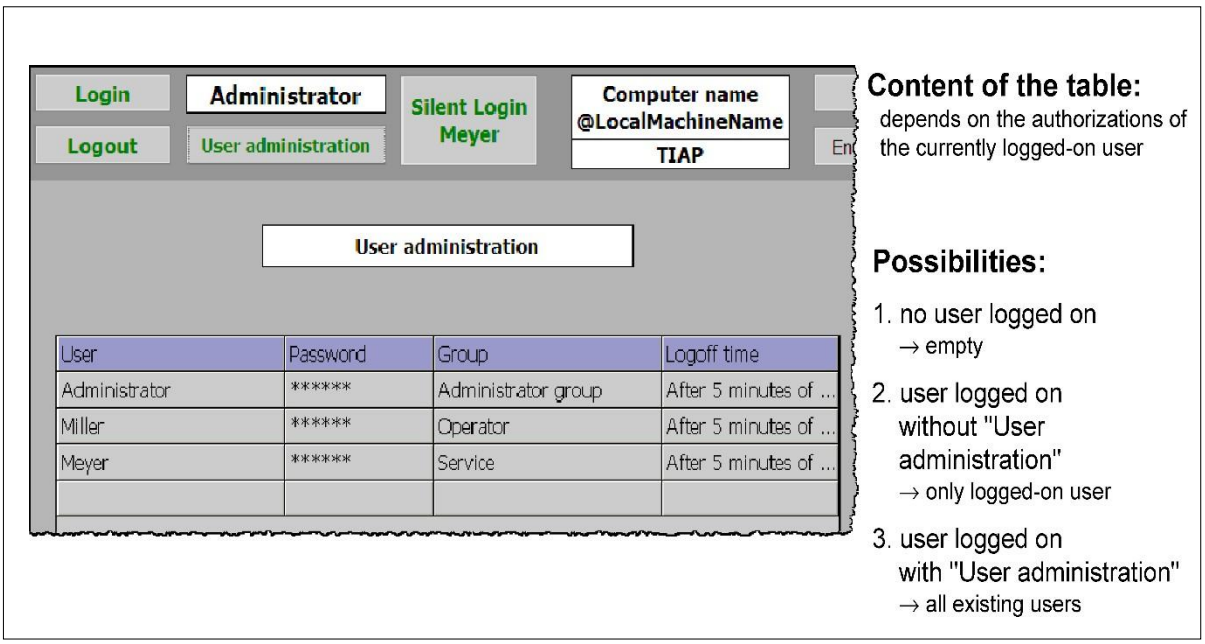


The "**User administration**" authorization (or authorization with the number =1) is always assigned automatically to the "**User view**" and controls the number of users displayed or managed in the user view on the HMI device in Runtime.

Administering users in runtime

A configured "**User view**" allows administration of the users on the HMI device in runtime. The user view shows different contents depending on the authorizations of the logged-on user:

- If no user is logged on the user view is empty.



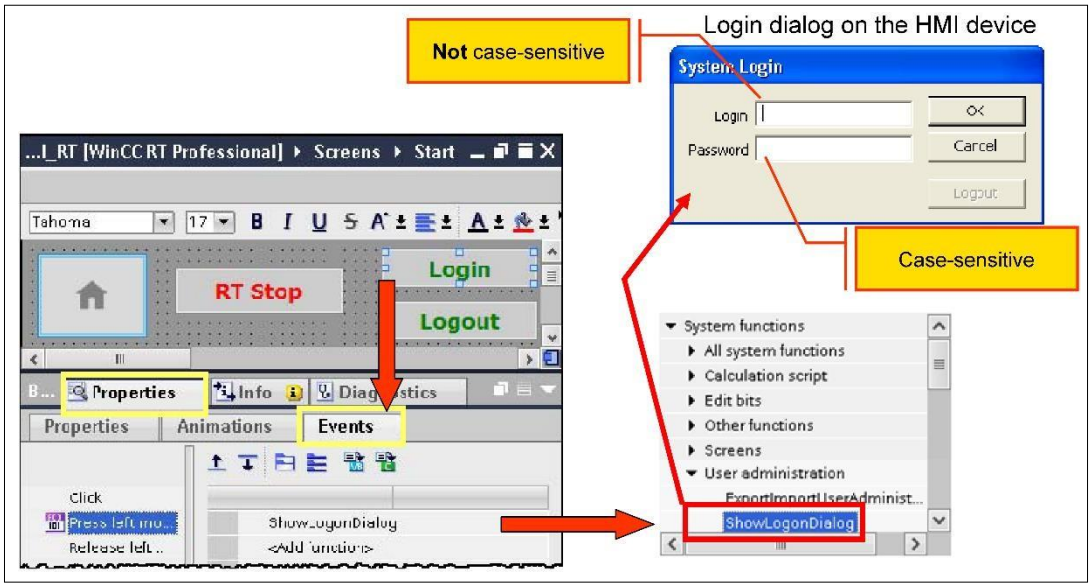
- The user is logged on and does not have "**User administration**" authorization the user view only shows the logged-on users. Users can change their own user names and passwords as well as the settings for the logoff time.
- The user is logged on and has "**User administration**" authorization (e.g. "**Administrator**" user)

The user view shows all users of the HMI device. This user can change all users displayed in the user view.

- Create user
- Delete user
- Modify user (user name, password, group assignment, logoff time)

Configuring a login dialog

The "**Logon dialog**" is displayed automatically on the HMI device in Runtime when a protected object is accessed, when no user is logged on or when the logged-on user does not have the



required authorization. So that the user is not surprised unexpectedly by this, a logon button should also be configured, permitting a specific user logon.

System function "showlogondialog"

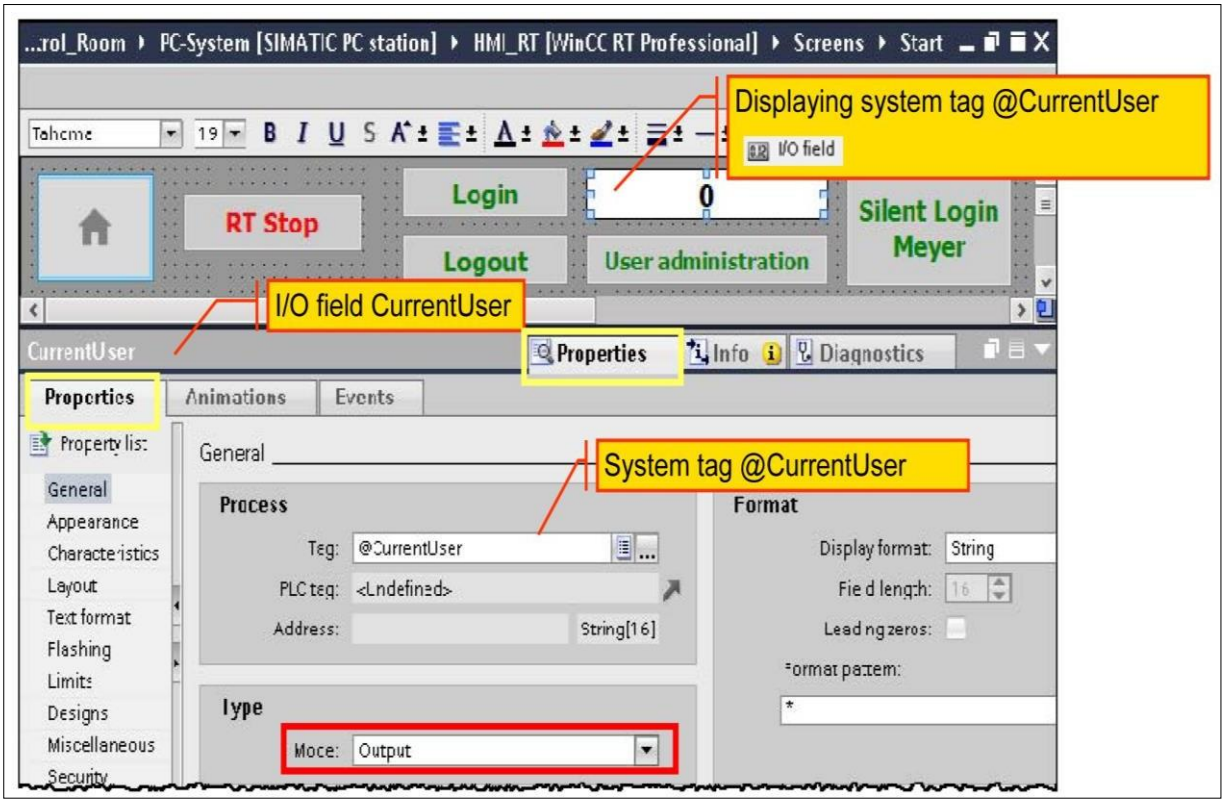
> Inspector window > Properties+ Events tab > "showlogondialog" system function

This function displays the logon dialog, and enters the user as a logged-on user following a successful logon at the HMI device. If logging on is unsuccessful (e.g. Incorrect password or similar), a system alarm is output, and no user is then logged on to the HMI device.

This system function is located in the function group "User administration".

Displaying name of logged-on user

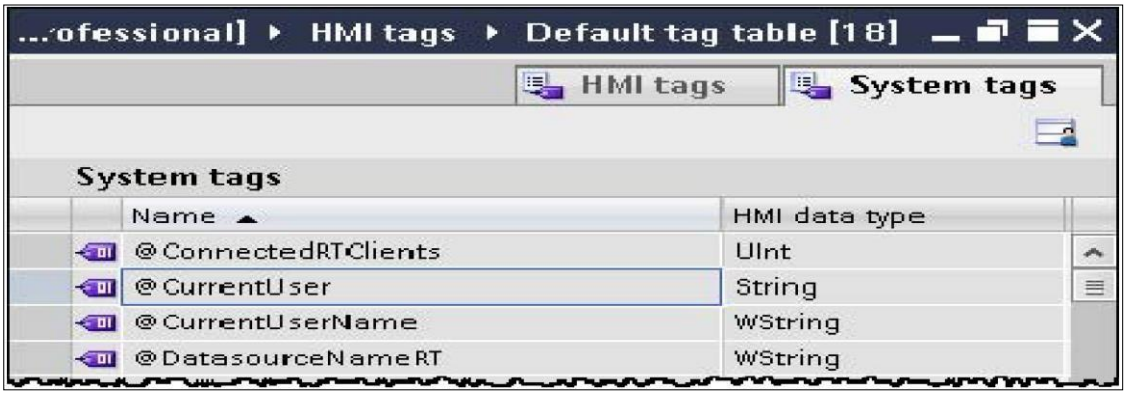
To display the currently logged-on user, the system tag **@Current User** is available. This tag is generated as an internal tag (string tag) when a project is created.



System tag @Current User

WinCC Professional provides various system tags. The system tag @Current User can be found in the System tags tab of the standard tags table:

> Project tree > HMI device > HMI tags > Default tag table > System tags tab



Displaying system tag @ current user

The system tag "**current user**" of the type "**String**" is displayed via an I/O field in output mode in a screen (it makes sense to display this in the permanently visible area of the start screen if a screen window is used in the start screen).

Task card >> Toolbox > Elements > I/O field

> Inspector window > Properties tab+ Property