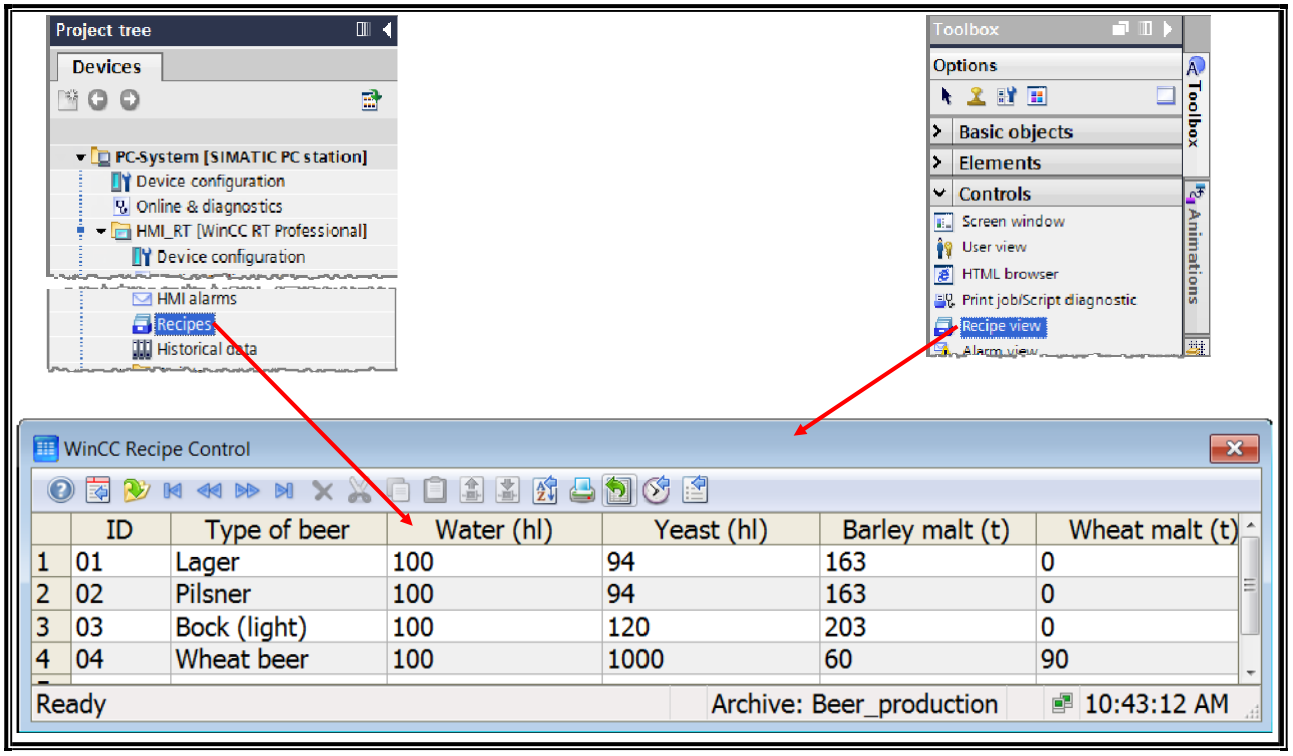
Skill AP
APSSDC

Skill AP
Learn Anytime Anywhere
Andhra Pradesh State Skill Development Corporation

Basics Of SCADA

Recipe Management

12. Recipe management



13.

In this section participants will learn about and practice working with the structure of recipes and the recipe view.

Participants will get to know the recipe editor. They will get an overview of the options for transferring data between the recipe management on the HMI device and the connected controller.

Learning objectives:
Participants will be able to configure recipes, specify recipe properties, configure recipe views, and transfer recipe data between an HMI device and controller

13.1. Overview

13.1.1. What are recipes?

Recipes are a collection of uniformly structured data.

A product often has different variants that differ, for example, in size or in the proportions of individual components. All product variants can then be collected in one recipe.

Most people will be familiar with baking recipes from the food industry or color tables from color mixing stations. But recipes are also used in many other technical sectors, e.g.:

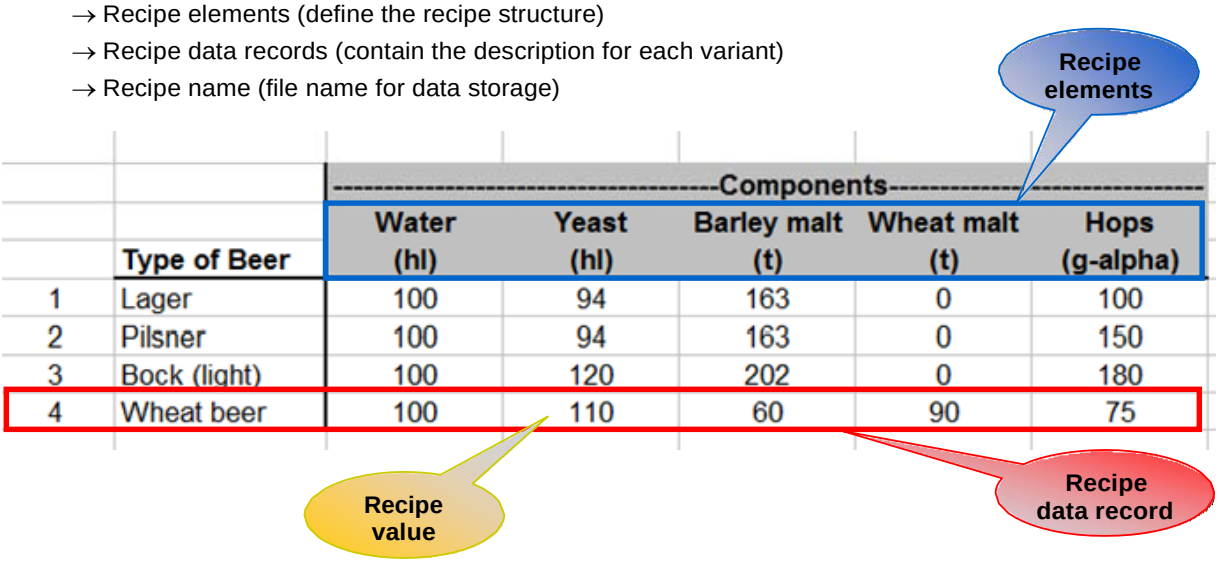
13.1.1.1. Machine parameter assignments

13.1.1.2. Tool lists

13.1.1.3. Production data such as mixing ratios, etc.

Structure and components of a recipe

- Recipes represent a grouping of data that is structured in the same way
- Recipe components
  - → Recipe elements (define the recipe structure)
  - → Recipe data records (contain the description for each variant)
  - → Recipe name (file name for data storage)



13.1.2.
13.1.3.

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

13.1.4. Entry/modification of recipe data

There are several ways to structure recipe data (initial entry). If the recipe data is already known, it can be entered during the configuring phase (in the WinCC V11 engineering software) and downloaded to the HMI device together with the configuration. If the recipe data is already available in electronic form, it can be imported on the HMI device. Alternatively, the data can be entered by the operator directly on the HMI device.

Maintenance (modification and addition) of the data is typically always carried out within the process operation directly on the HMI device.

13.1.5. Saving recipe data

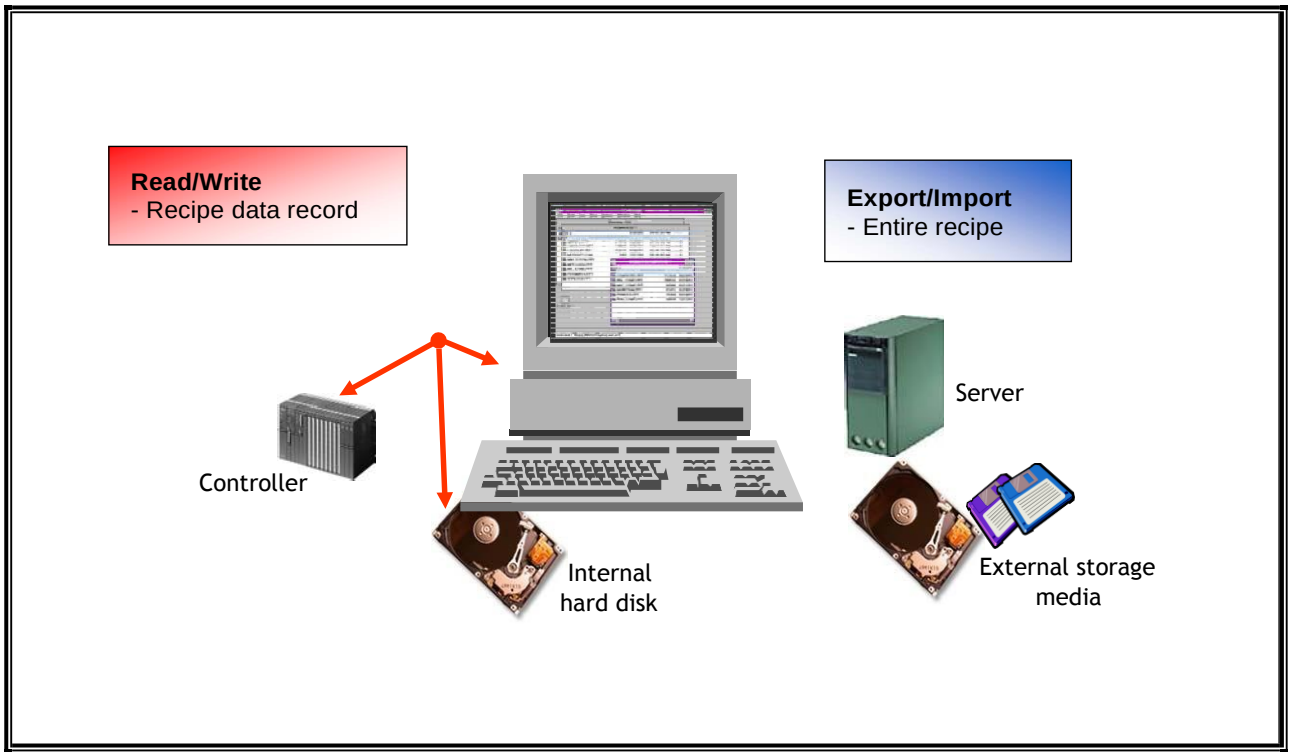
All data records of a recipe are stored in a database on the HMI device. In WinCC Professional, the memory location is always the database.

13.1.6. Backup of recipe data (export/import)

The recipe data are backed up either using a button in the toolbar of the recipe view (Export archive) or in a C script using the API function "uaArchiveExport" (API > Application Interface > calling interface for internal WinCC Professional system functions). The export is made in CSV format.
The recipe data can be imported either using a button in the toolbar of the recipe view (Import archive) or in a C script using the API function "uaArchiveImport".

13.1.7. Configurable recipe functions for data transfer

13.1.8.



For transferring the recipe data records, either a "Recipe view" with the corresponding functionalities or C scripts with corresponding API functions are used.

For exporting or importing recipe data, one API function each is available for use in C scripts.

Transfer of recipe data to the controller

Recipe data is transferred between the HMI device and the connected controller record by record.

Available functionality (detailed explanations of the functions can be found in the following sections):

13.1.8.1. Write data records to the recipe tags thereby transferring them to the controller

13.1.8.2. Read data records from the recipe tags thereby retrieving them from the controller

13.1.8.3. Delete data records

13.1.8.4. Create new data records

Exporting/importing recipe data

Recipe data can be exported or imported to the HMI device.

Available functionalities:

13.1.8.5. Export recipe data (selected recipe data records or all recipe data records) to a file in CSV format

13.1.8.6. Import recipe data (all recipe data records) from file in CSV format

13.2. Selecting a recipe data record on the HMI device and exchanging data with the controller

Transferring the recipe data between the controller and HMI device

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
- "Tags" communication type
- "Raw data tag" communication type

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

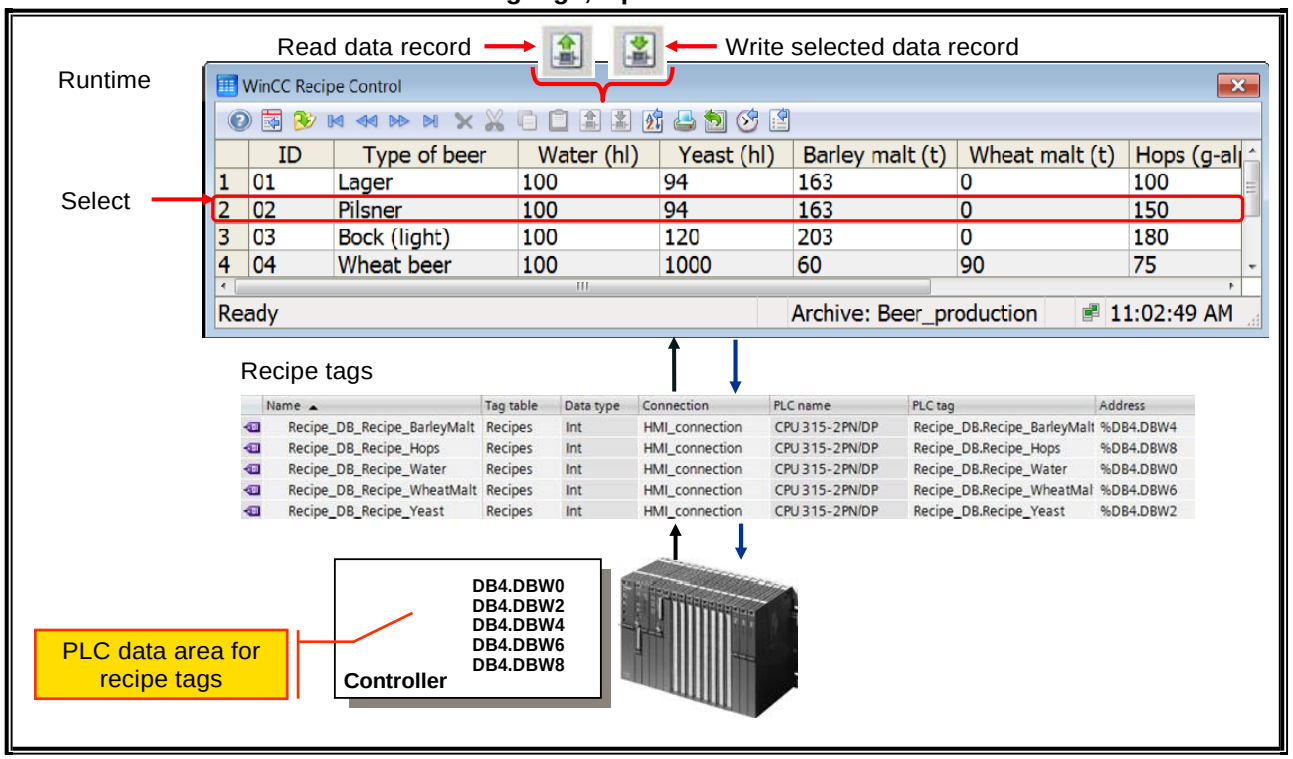
Recipe tags transfer the recipe values to the controller. Conversely, the values from the controller are written to a recipe data record via recipe tags. In so doing, the data exchange is started by the operator using the buttons (Read tag/Write tag) in the recipe view or controlled via control tags.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

Communication with the controller using tags, Operator-controlled



Recipe selection and transfer of a recipe data record can be carried out using the "Recipe view" configuration object or a freely-designed recipe screen.

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 and change the size.

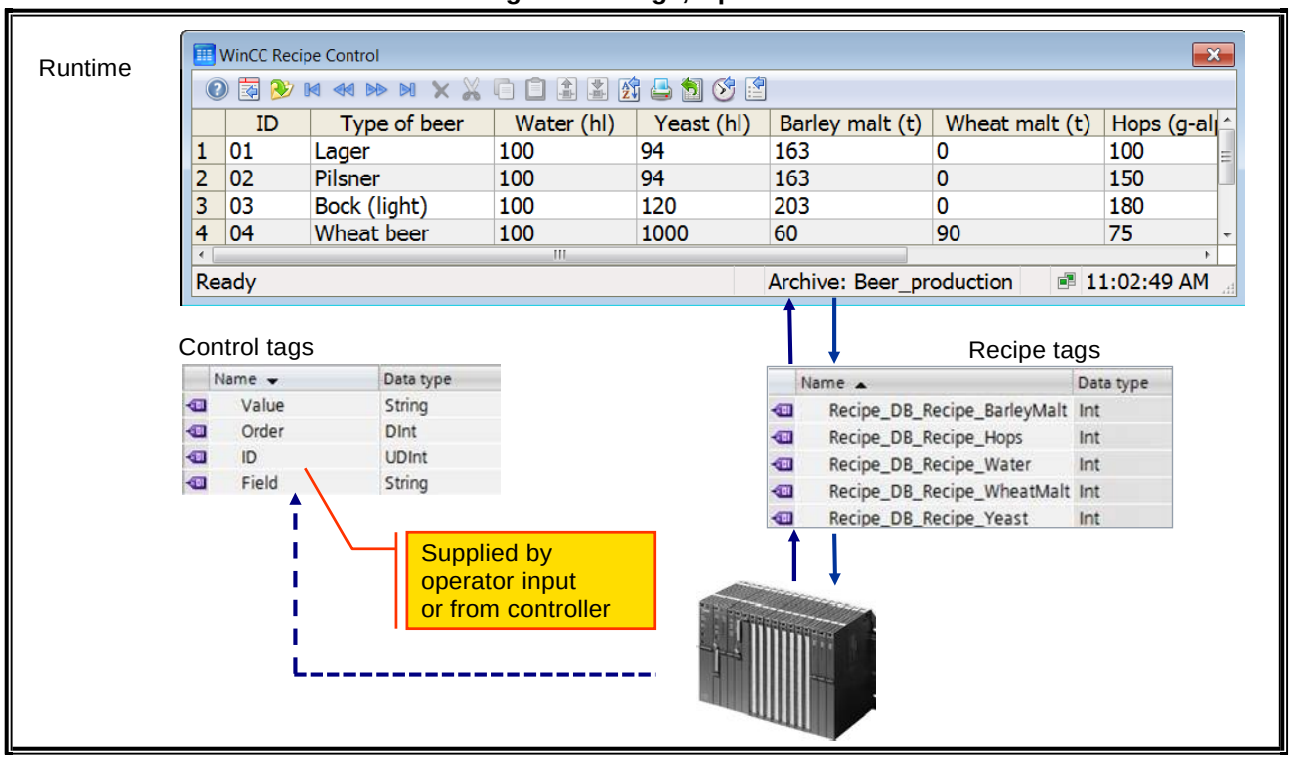
Recipe screen

In contrast to the recipe view, a recipe screen can be freely configured.
The advantages of using a recipe screen are the user-defined screen layout and the transfer functions,

e.g., configuring a specifically labeled button Write Recipes to Production instead of the standard button  .

However, the user must configure the corresponding functionality for the data record transfer, e.g., using control tags or C scripts.

Communication with the controller using control tags, Operator- or PLC-controlled



Function of the control tags

Control tags are used for controlling the data transfer between HMI device and controller (PLC). 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

Control tags

Control tag	Required data type	Meaning	
ID	DInt (long)	Number of the recipe data record within the assigned recipe	
Job	DInt (long)	Read from the recipe tags, write to the recipe memory:	6
		Read from the recipe memory, write to the recipe tags:	7
		Delete in the recipe memory:	8
		After the job is run, the "Job" control tag contains an error ID:	
		No error	0
		Error	-1
Field	String	The name of the recipe element	
Value	String	The value of the recipe element	

More value combinations of the "ID" and "Job" control tags

ID	Job = 6	Job = 7	Job = 8
-1	Data record is read from the recipe tags and appended in the recipe in the recipe memory **at the end**.	-	Data record with the **lowest** ID is deleted in the recipe in the recipe memory.
-6	Data record is read from the recipe tags and then written to the recipe data record with the **lowest** ID in the recipe.	Data record with the **lowest** ID in the recipe is written to the recipe tags.	Data record with the **lowest** ID is deleted in the recipe in the recipe memory.
-9	Data record is read from the recipe tags and then written to the recipe data record with the **highest** ID in the recipe.	Data record with the **highest** ID in the recipe is written to the recipe tags.	Data record with the **highest** ID is deleted in the recipe in the recipe memory.

"Raw data tag" communication type

As an active partner, the automation system sends the raw data tag to the HMI device in the form of a message frame. The message frame contains the following:

- Desired target recipe
- Desired target recipe data records
- Job, e.g., "Read", "Write", or "Delete"
- New recipe values, e.g., for "Write" job

The HMI device returns recipe and acknowledgement data in the form of a message frame.

The "BSEND/BRCV" function of the S7 communication is used for the data exchange. The consecutive number of the recipe (Recipe properties > General > Settings > Number) serves to identify the recipe uniquely for the controller (PLCID).

The controller requires a user program that executes the following in runtime:

- Assembles data if necessary via I/O fields in the HMI device.
- Checks data.
- Creates a raw data tag as a message frame.
- Sends message frame.

In order to start the message frame generation from the HMI device, the WinCC Professional project requires additional functions that start the user program in the controller.

Note:

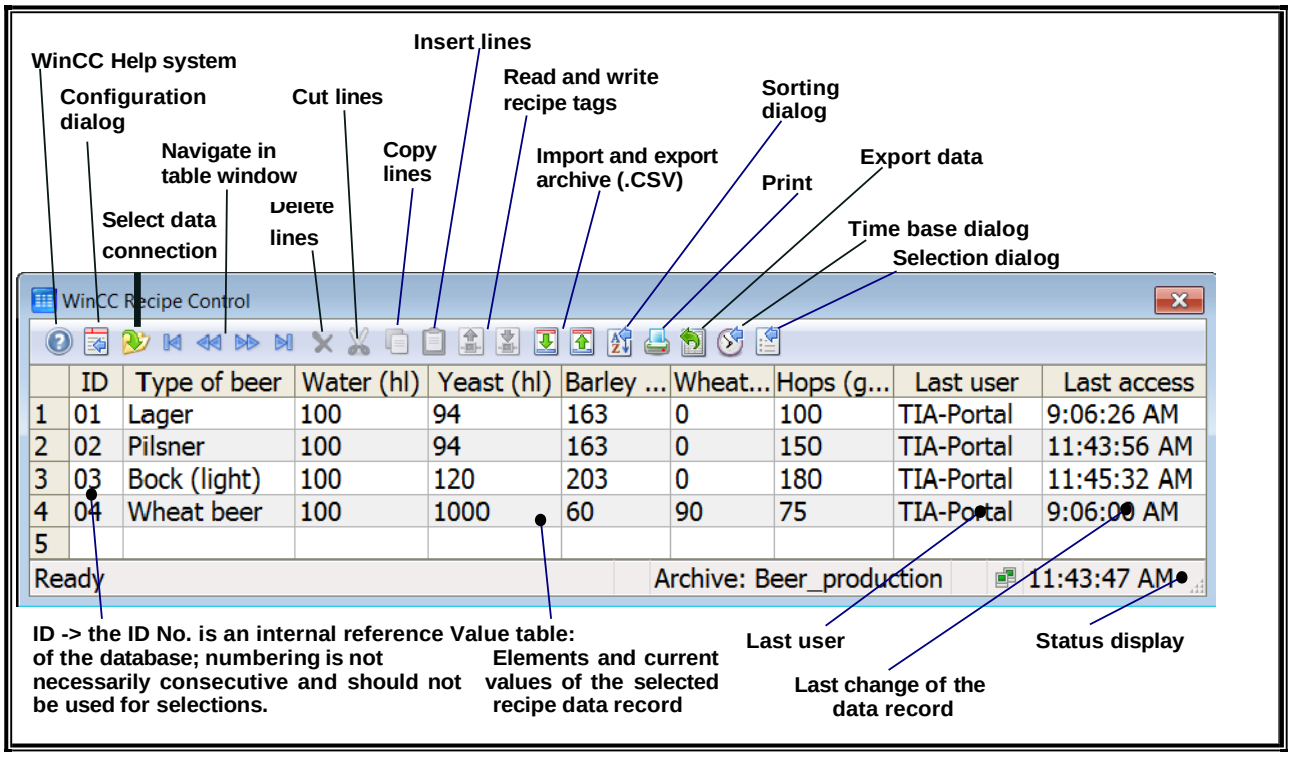
The following is possible in the "Raw data tag" communication type:

- Transfer of an individual recipe element
- Transfer of an entire recipe data record
- Transfer of multiple recipe data records

Data transfer from the recipe view or via control tags is not possible.

13.2.1. Functions of the recipe view

13.2.2.



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.

For touchscreen applications, the functions of the toolbar buttons can also be triggered using scripts for user-defined buttons (e.g., using VB script: "ScreenItems("RecipeView_BeerProduction").ToolbarButtonClick = 13", in order to write the selected data record to recipe tags; the parameter value "13" results from the number (toolbar button ID) of the button).

Toolbar buttons of the recipe view

The "Recipe view" provides various functions as toolbar buttons. Each toolbar button function can be called in a script using a toolbar button ID. An operator authorization can be configured for the toolbar buttons.

Help (ID 1)

Calls the help for the recipe view.

Configuration dialog (ID 2)

The configuration dialog can be used to change the properties of the recipe view in runtime operation.

Select data connection (ID 3)

Opens a dialog for selecting a recipe or recipe query. The recipe data are then shown in the table of the recipe view.

Navigation functions: First line, Previous line, Next line, Last line (IDs 4-7)

These buttons can be used to quickly navigate in the database in order to display the desired data records in the table of the recipe view.

Editing functions: Delete lines, Cut lines, Copy lines, Insert lines (IDs 8-11)

These buttons can be used to to perform various editing functions in the recipe view.

Read tags (ID 12)

This button is used to read the contents of the connected WinCC tags and write them to the recipe elements. The "Tags" communication type must be activated for the displayed recipe in order to use the button. The recipe elements must be connected to tags.

Write tags (ID 13)

This button is used to write the contents of the recipe elements to the connected WinCC tags. The "Tags" communication type must be activated for the displayed recipe in order to use the button. The recipe elements must be connected to tags.

Import archive (ID 14)

This button is used to import the recipes (in a CSV file) from the "ua" directory of the project folder to the table of the recipe view if no other path setting is selected.

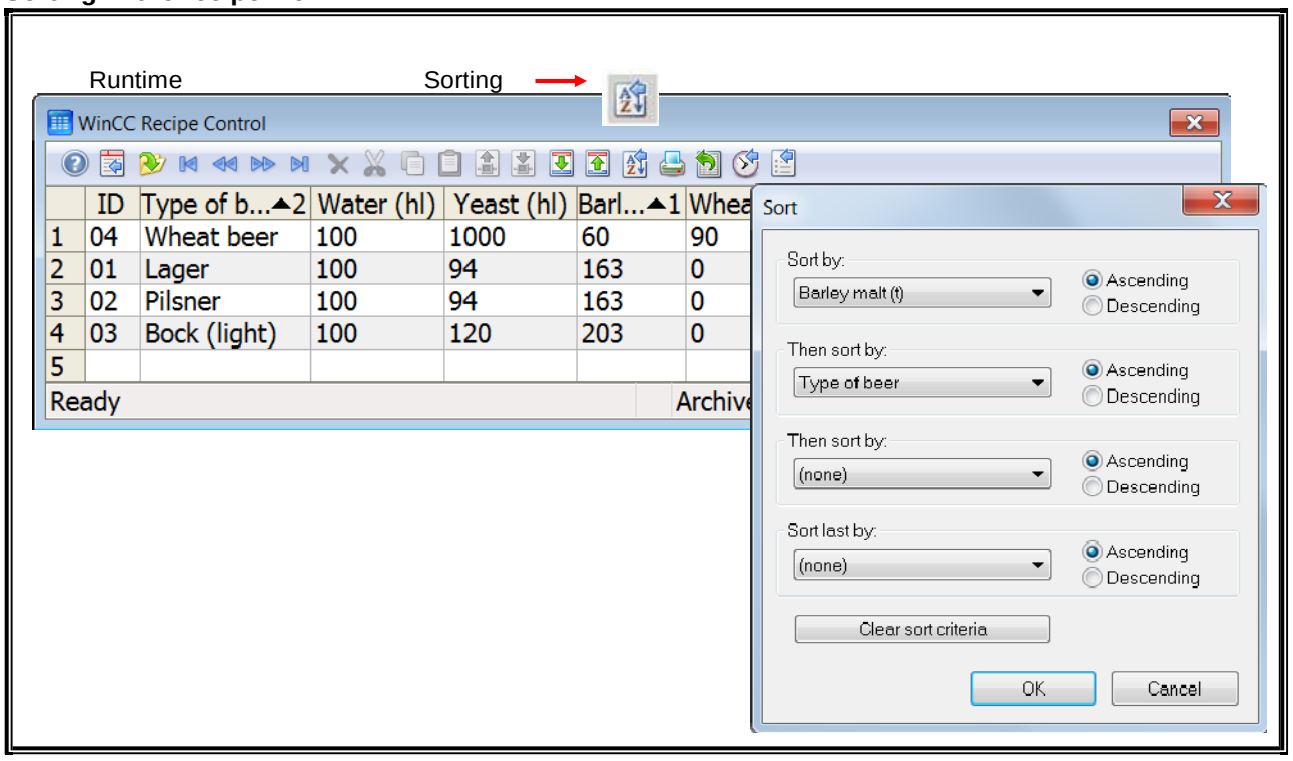
Export archive (ID 15)

This button is used to export the original loaded content of the table together with table headers. The recipe is stored as a CSV file in the "ua" directory of the project folder, if no other path setting is chosen.

Sort dialog (ID 16)

Opens a dialog box for entering a user-defined sorting of the displayed columns.

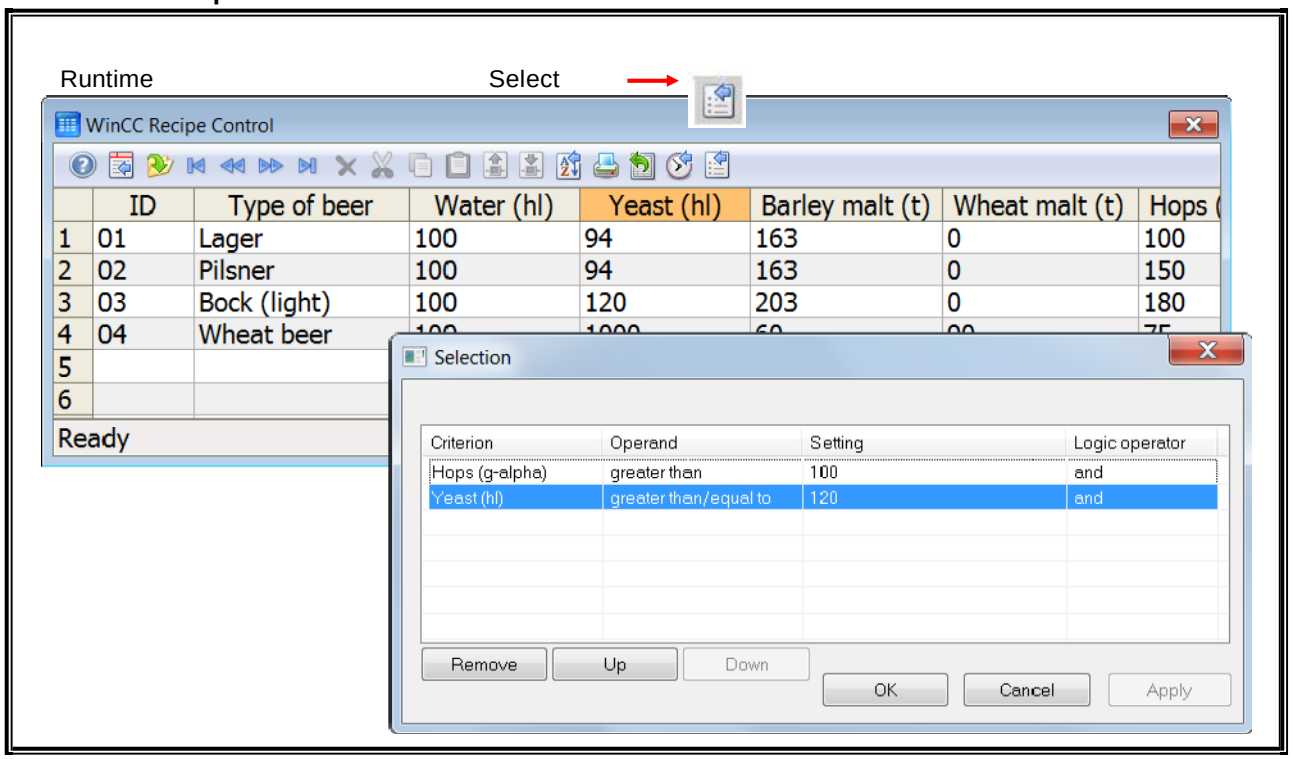
Sorting in the recipe view



Selection (ID 17)

Opens a dialog for selecting data records using various criteria

Views in the recipe view



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.

Print (ID 18)

Starts the printout of the displayed values. The print job used for printing is specified in the "General" tab of the configuration dialog.

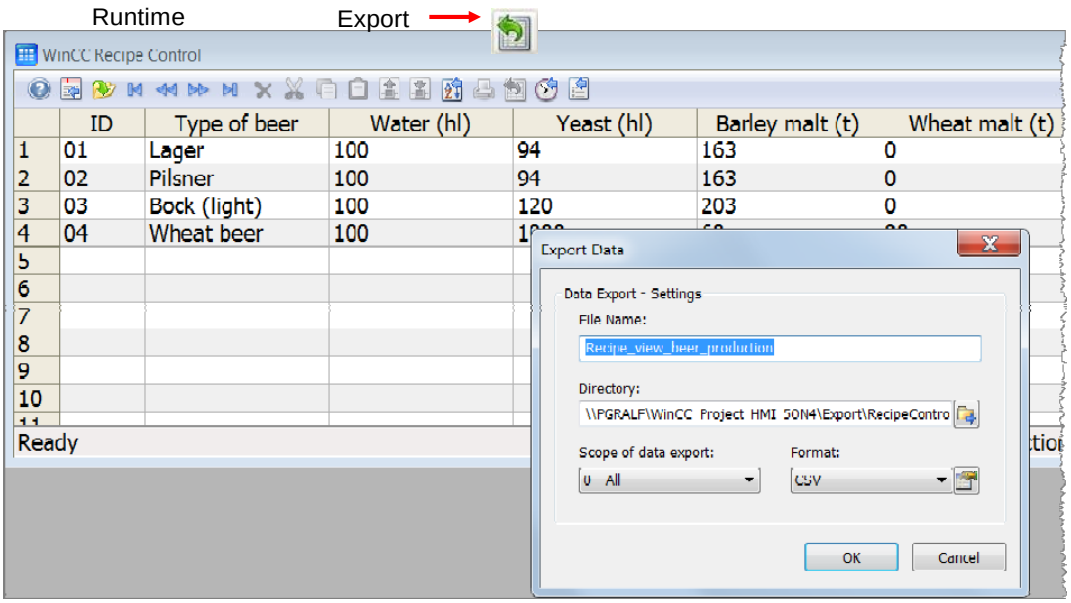
Time base dialog (ID 19)

Opens a dialog for setting the desired time base:
"Local time zone" or "Coordinated Universal Time (UTC)"

Export data (ID 20)

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.

Exporting data in the recipe view



User-defined 1 (ID 1001)

Displays the first button created by the user. The function of the button is user-defined.

C scripting functions for recipes (Runtime API)

The Runtime API provides comprehensive uaArchive functions (ua: User Archive -> Recipes) in order to process recipes and recipe queries in runtime. The TIA Portal information system (Section "Interfaces > Runtime API" > Section "Functions for recipes") contains detailed information on these functions.

13.3. Procedure for configuring a recipe

Procedure for configuring a recipe

The following steps must be performed when configuring recipes:

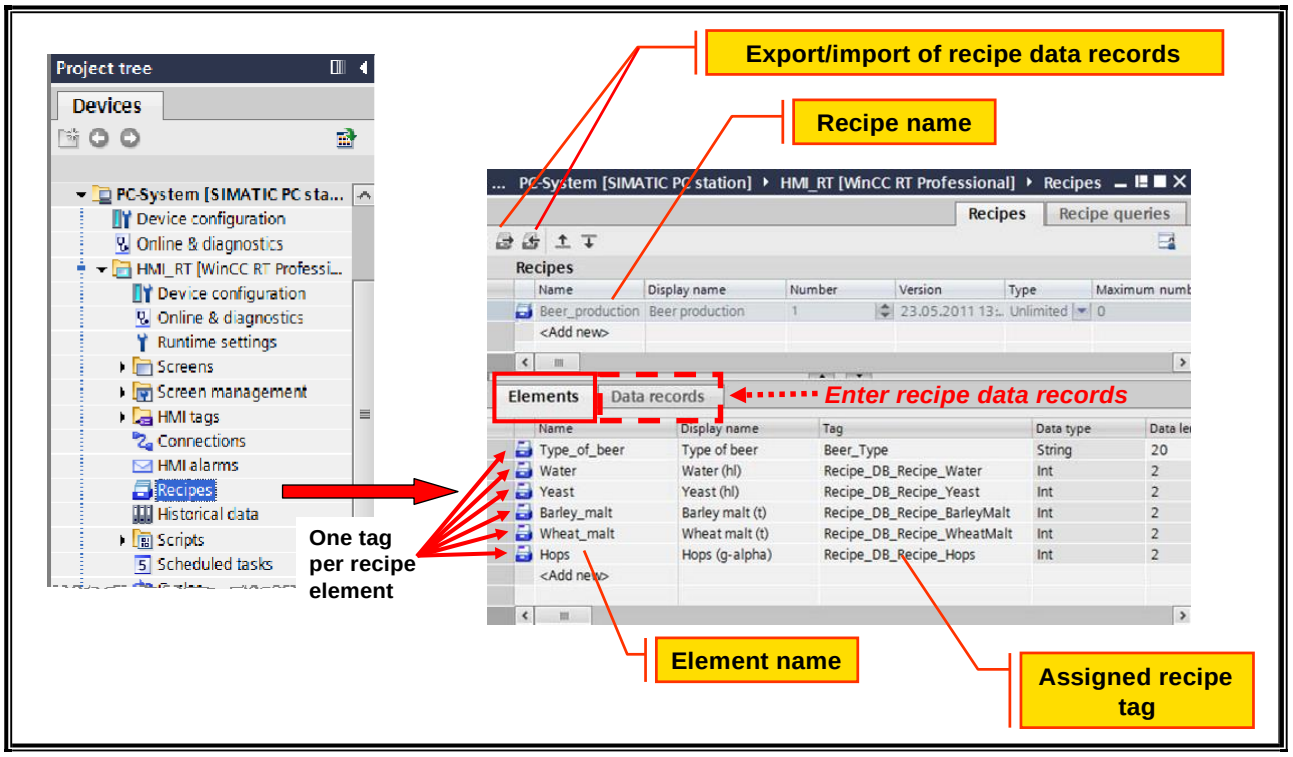
1. **Recipe definition**
   Configure the recipe structure and recipe data area in the controller

2. Set **properties** of a recipe

3. Configure **recipe view** for
   → Recipe display and selection
   → Recipe entry and modification

13.3.1. The Recipes editor

13.3.2.



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.

Name (recipe name):

13.3.2.1. Identifies the recipe uniquely within the project

13.3.2.2. 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).

Display name:

13.3.2.3. Appears on the HMI device, e.g., in the recipe view

13.3.2.4. Multilingual configuration is possible

Recipe number:

13.3.2.5. Identifies the recipe uniquely within the project

Version:

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

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".

Display name (recipe element display name):

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

Tag:

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

Default value:

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

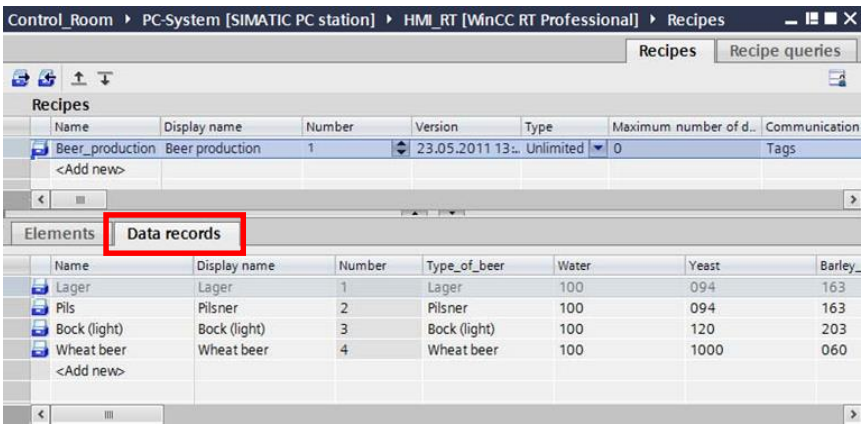
Tooltip:

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

Entry of recipe data

Initial entry

- **At time of configuring**
  - When recipes are configured (in the "Recipes editor")



Recipe entry

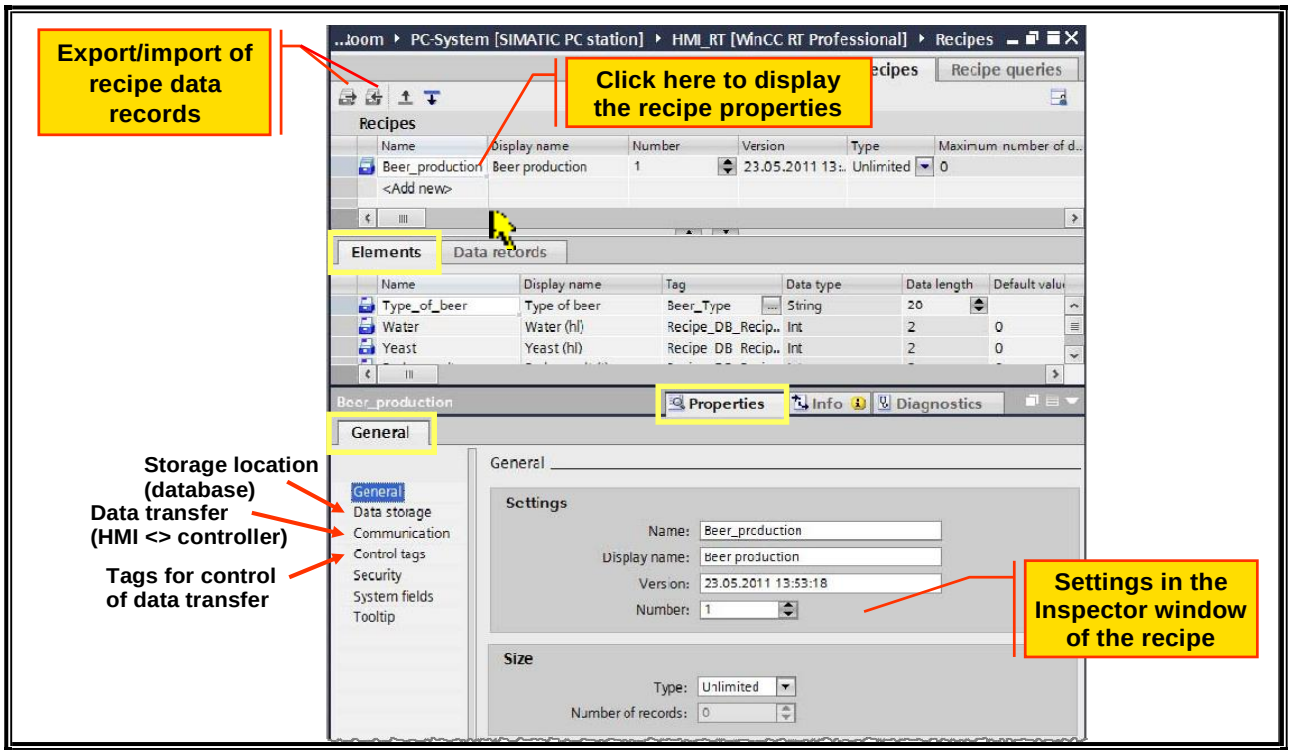
Recipe data records can be entered at the time the recipes are configured (in the "Recipes" editor). These recipes are then transferred to the HMI device along with the configuration.

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.

13.3.3. Setting recipe properties

13.3.4.



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

10.3.2.1. 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:

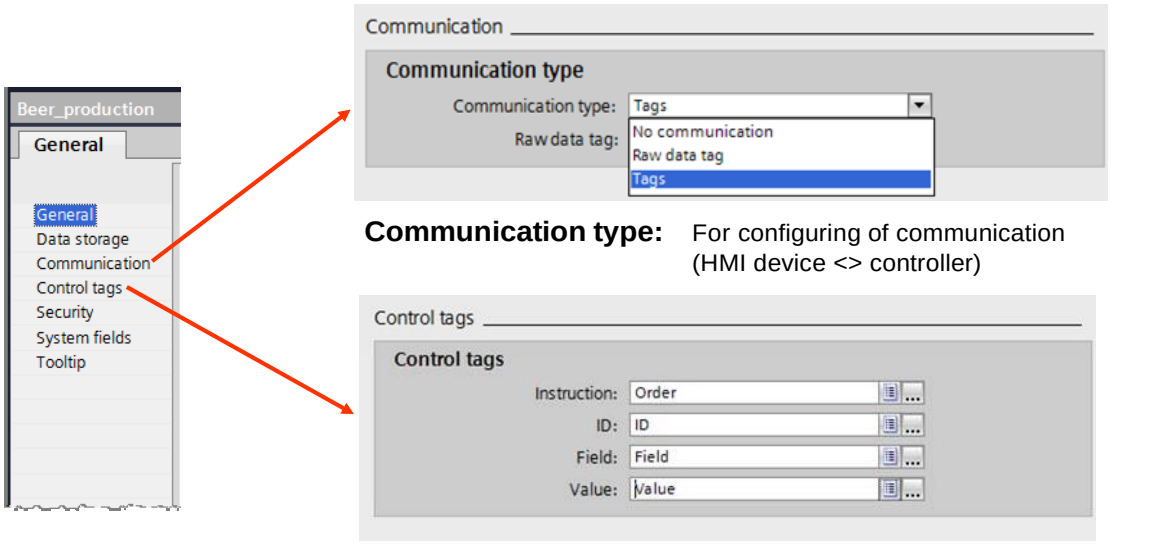
- By means of recipe tags via user-actuated buttons in the recipe view or with the aid of control tags

- 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.

Recipe properties: Communication using tags/control tags



Communication type: For configuring of communication (HMI device <> controller)

Control tags: for “Tags” communication type between HMI device and controller

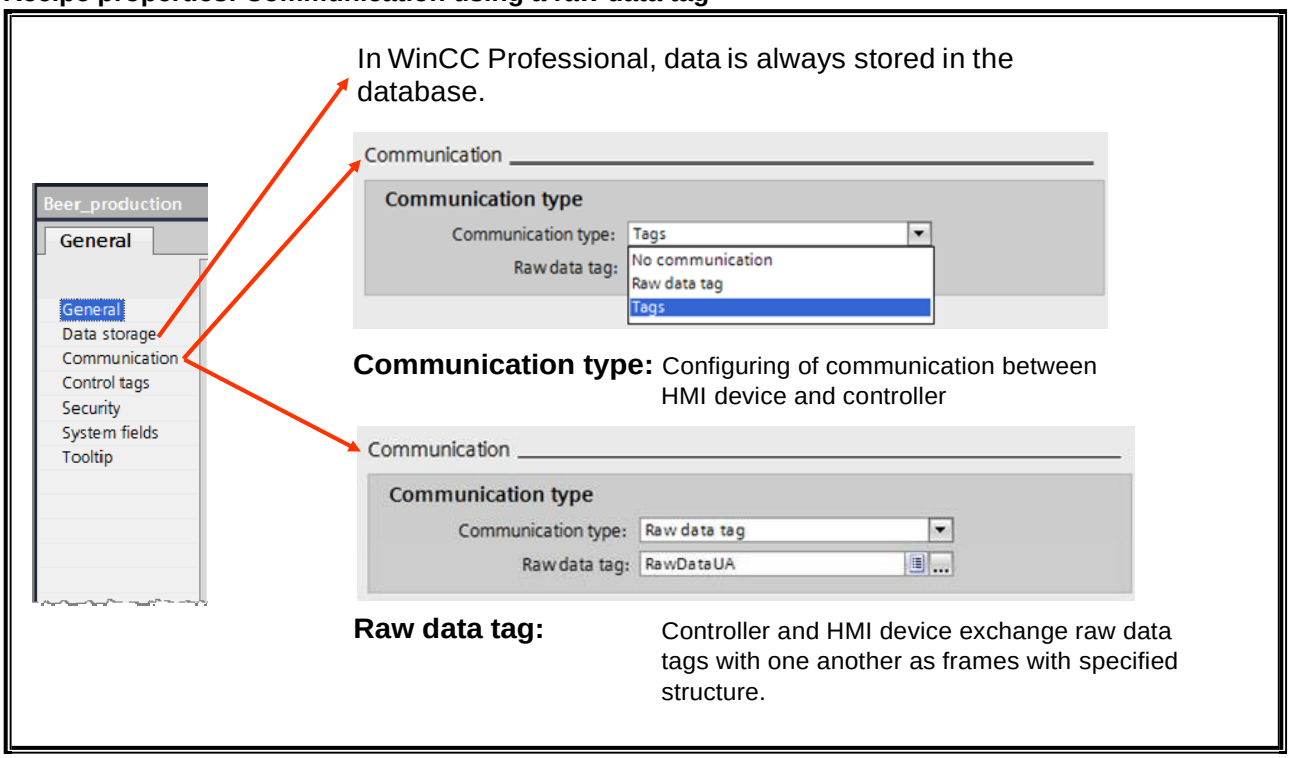
Communication type **"Raw data tag"**

The controller and the HMI device exchange raw data tags as message frames with a fixed structure.

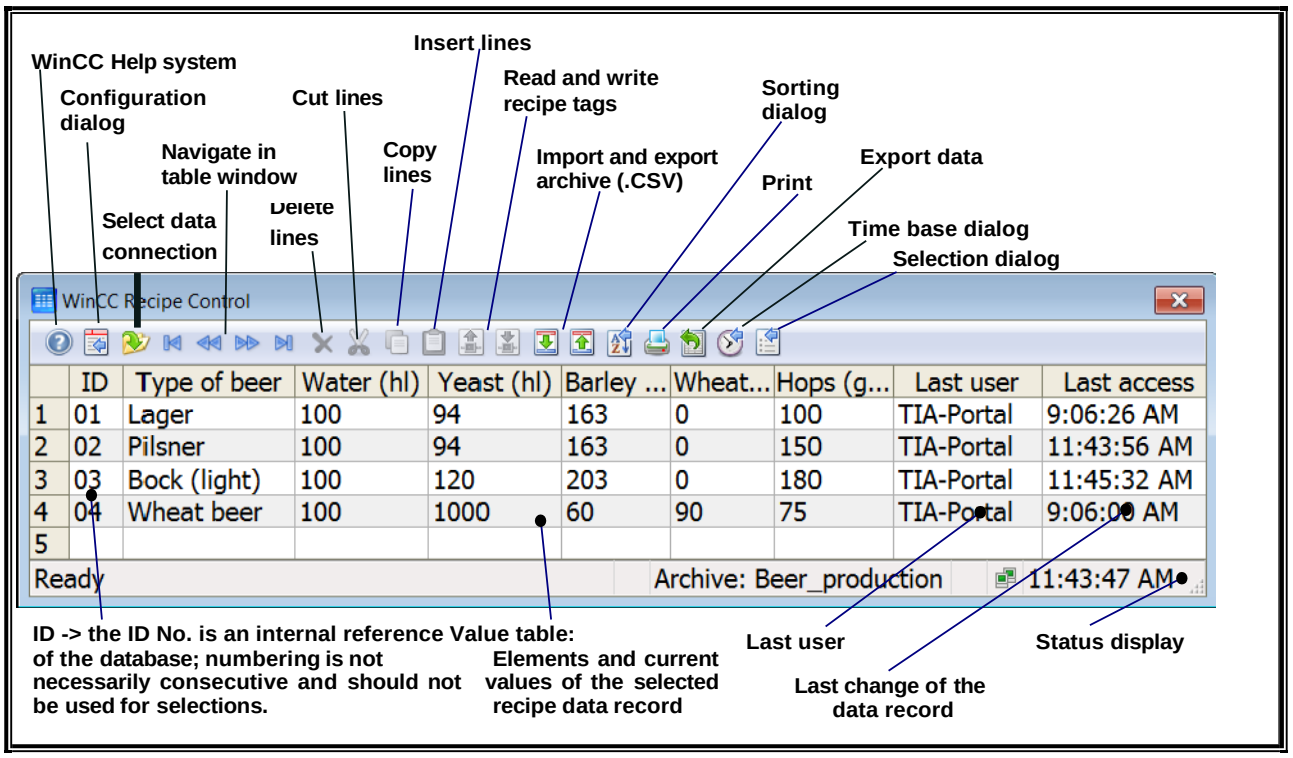
Raw data tags offer the following advantages:

- 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

Recipe properties: Communication using a raw data tag



10.3.2.2. Recipe view

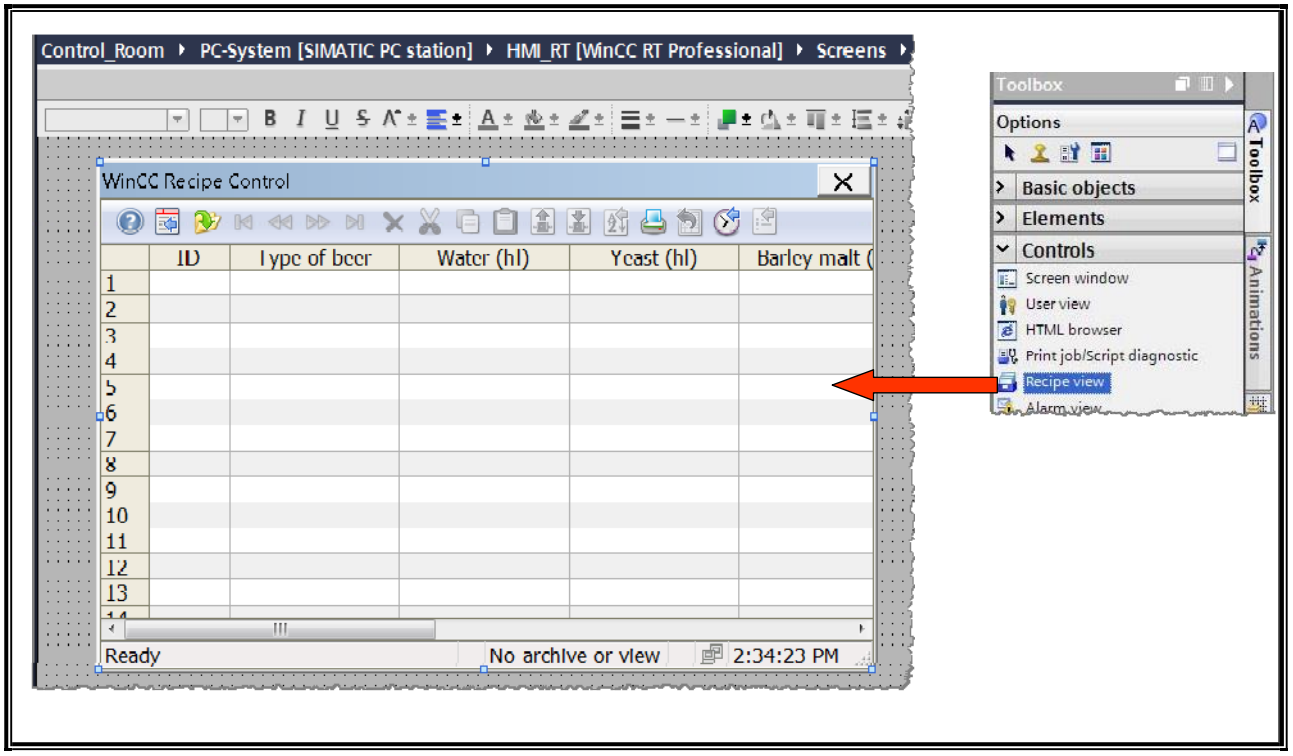


The recipe view has a fixed structure, which can be configured during the configuring step. The advantage is that configuration is simple and fast. However, the basic structure of the recipe view cannot be changed. It is only possible to show/hide display elements (properties such as table columns, toolbar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

10.3.2.3. Configuring the recipe view
10.3.2.4.



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.

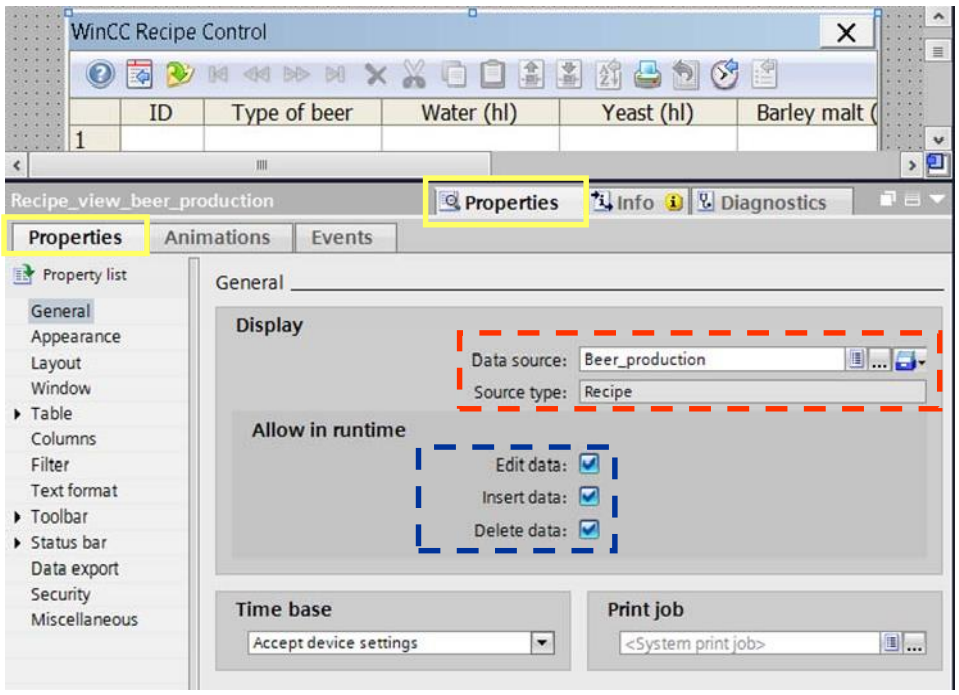
The most important properties of a recipe view can be found in the Inspector window in the sections:

- General
- Appearance
- Columns
- Symbol bar
- Label

These properties should always be checked and configured.

10.3.2.5. The most important properties of the recipe view

General property

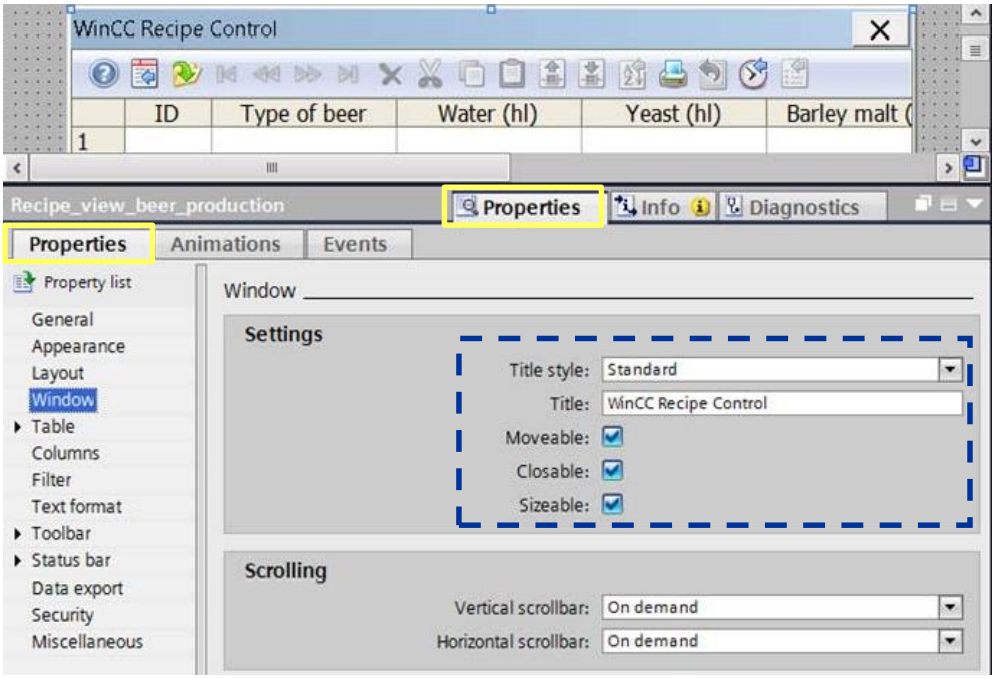


Selection of data source (recipe or recipe query)

>> Inspector window > Properties + Properties tab > General Section: "General"

In the "General" section, the read/write permissions for the recipe view in runtime are also specified in the "Allow in runtime" area.

Window property

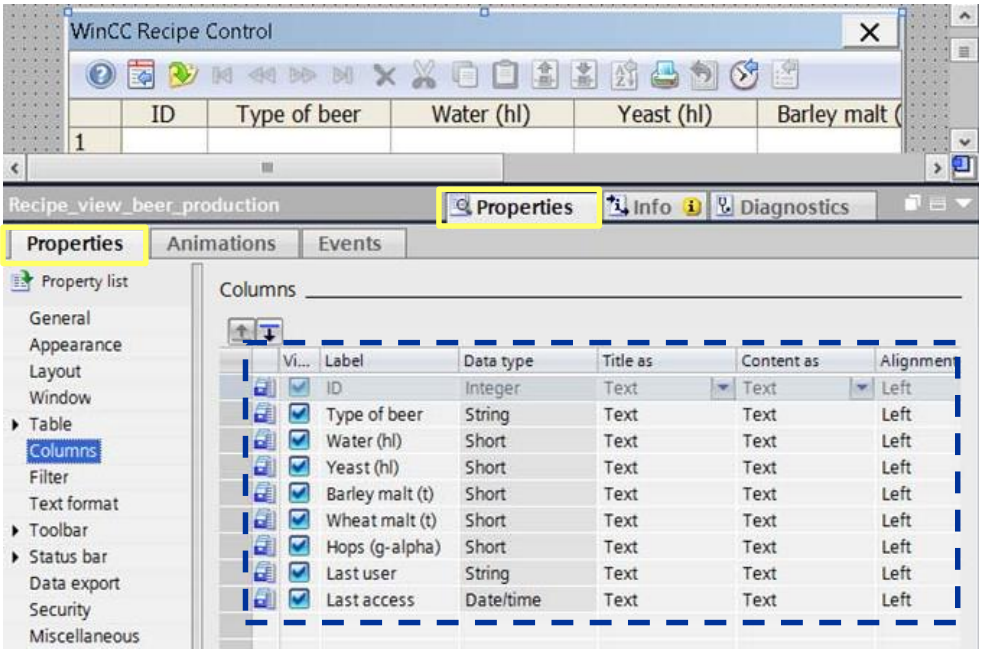


Window settings

>> Inspector window > Properties + Properties tab > Window
Section: "Settings"

In the "Window" section, the runtime behavior of the scroll bars is also specified in the "Scrolling" area.

Columns property

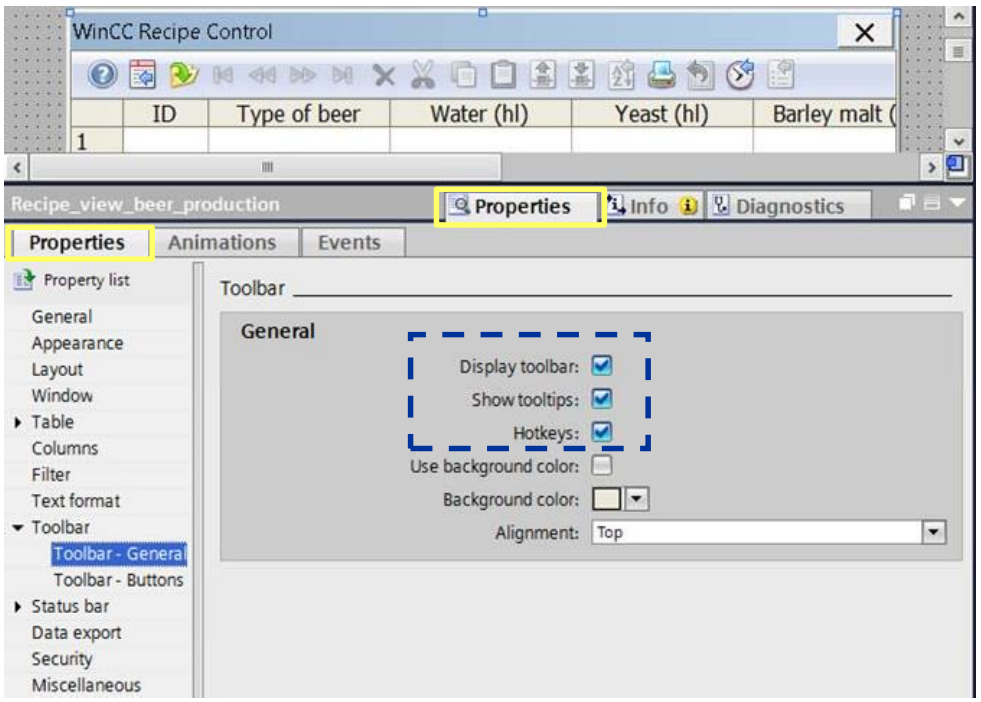


Columns

>> Inspector window > Properties + Properties tab > Columns

Here, the visible columns are selected and the column header is configured.

Toolbar - General property

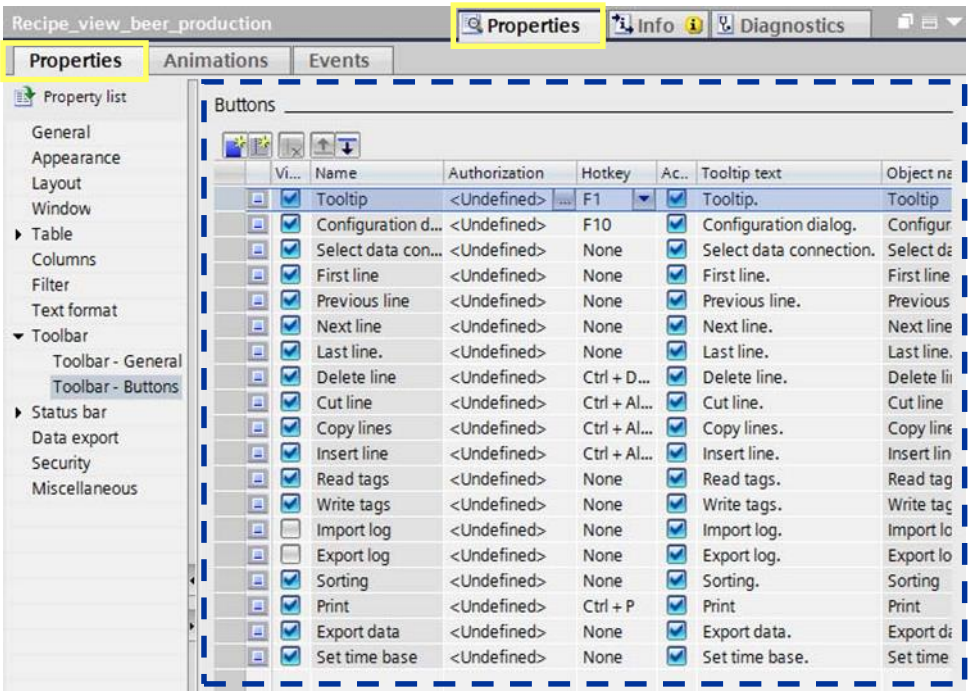


Toolbar - General

>> Inspector window > Properties + Properties tab > Toolbar_General Section: "General"

Here the runtime behavior of the toolbar is configured.

Toolbar buttons property

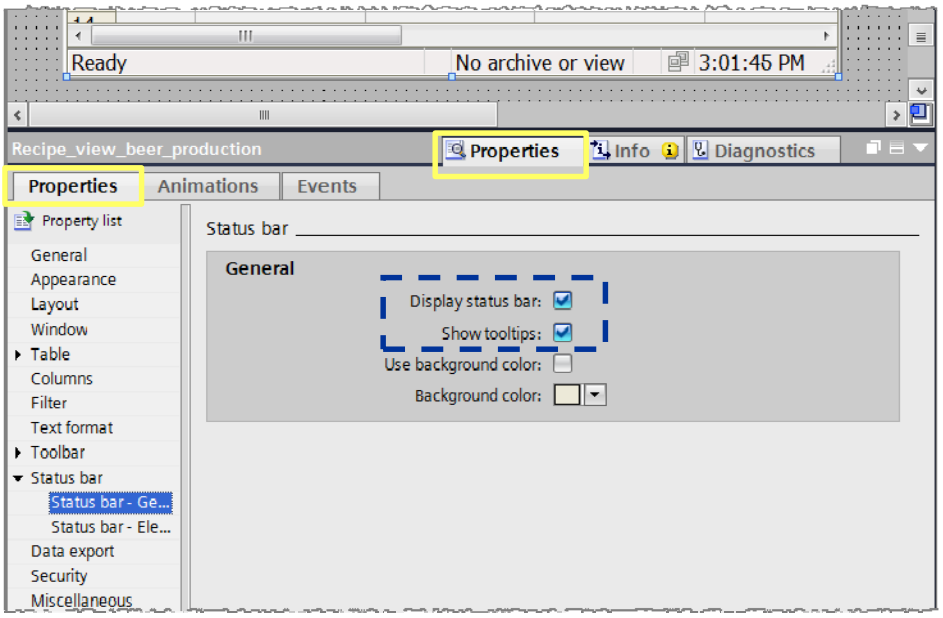


Toolbar buttons

>> Inspector window > Properties + Properties tab > Toolbar buttons

Here, the toolbar buttons required in runtime operation are selected, e.g., buttons for data record transfer, for import/export functions, or for sorting functions.

Status bar - General property

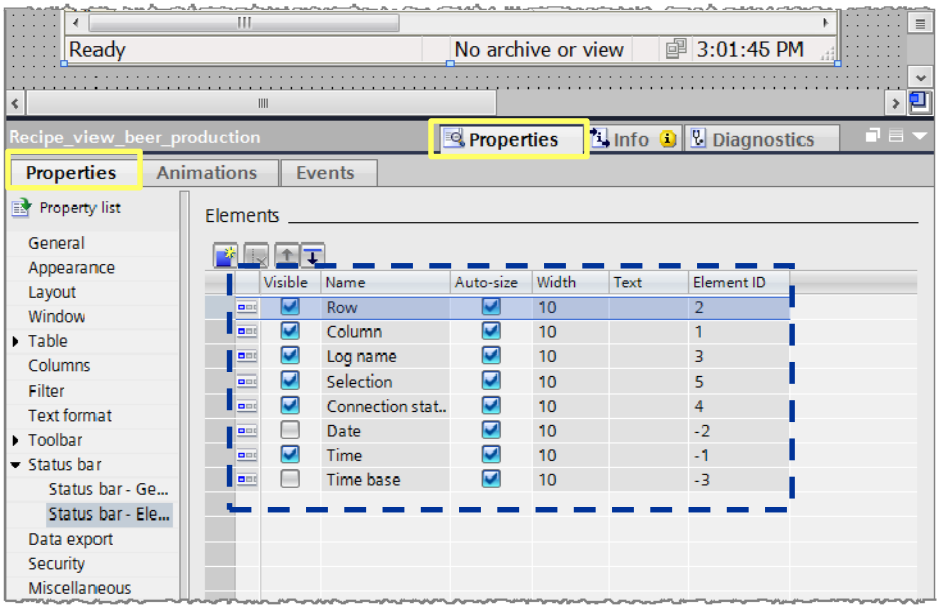


Status bar - General

>> Inspector window > Properties + Properties tab > Status bar - General

The status bar parameters are assigned here.

Status bar elements property



Status bar elements

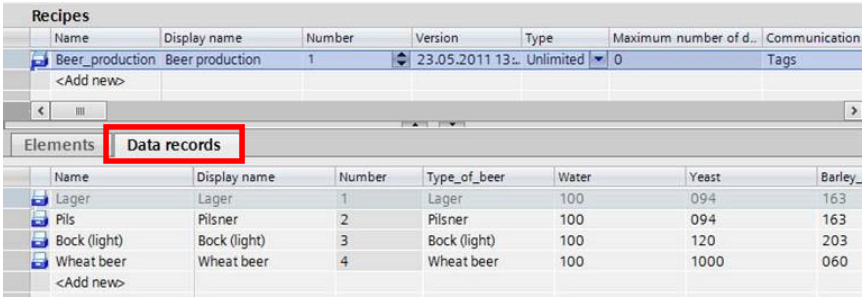
>> Inspector window > Properties + Properties tab > Status bar elements

Here, the parameters of the status bar elements visible in runtime operation are assigned.

10.3.2.6. Recipe data record entry

Initial entry

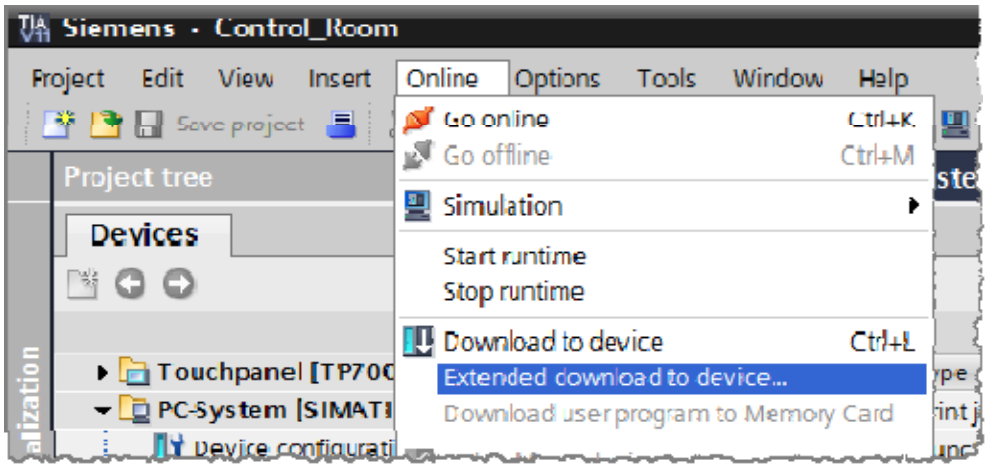
- **At the time of configuring**
  - When recipes are configured (in the "Recipes editor")
  - Transfer with configuration to the HMI device
    Notice!!! Existing recipe data on the HMI device can be overwritten



- **By data import on the HMI device using the recipe view**
- **By operator on the HMI device using the recipe view or recipe screen**

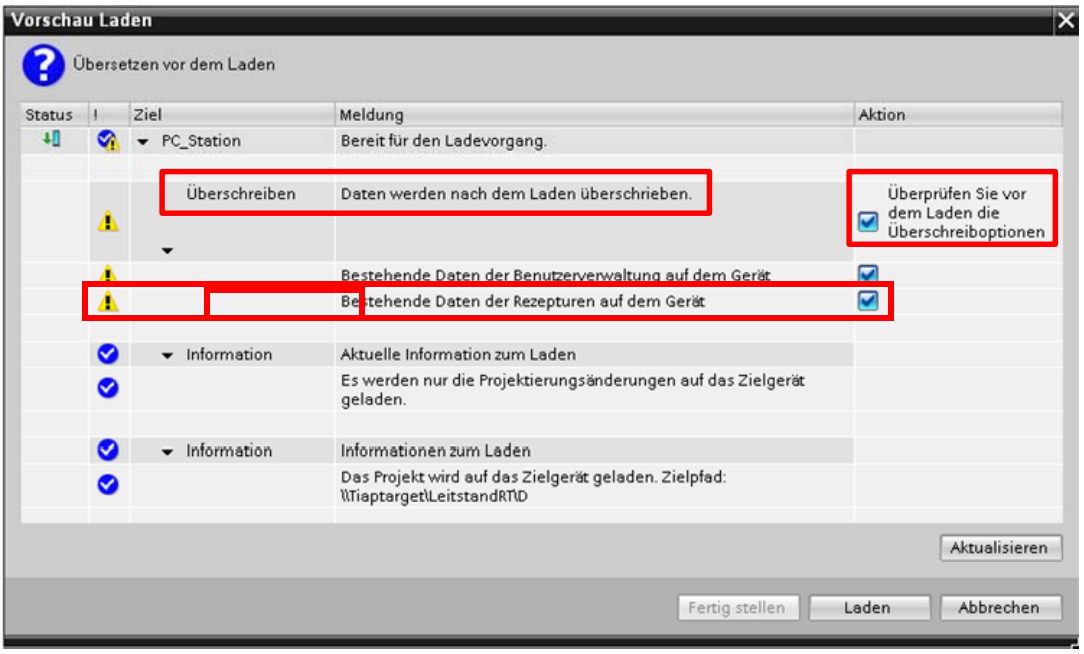
Recipe data records can be entered at the time the recipes are configured (in the "Recipes" editor).

If the WinCC project is not to be operated on the configuration computer but rather on a different computer (target computer), the overwrite options must be checked when loading the compiled project to the target computer to prevent loss of data from existing recipes on the target computer. The loading operation is called using menu item "Online > Advanced download to device".



Loading HMI device, Overwrite options

Load HMI device: Notice!! Existing recipe data on the HMI device can be overwritten.



The overwrite options must be checked before every loading operation to prevent loss of data from the recipes on the target computer.

Entry of recipe data records in runtime operation by data import on HMI device

Data can be imported to the HMI device using the import function in the recipe view.

Initial entry of recipe data records from production by the operator in runtime operation

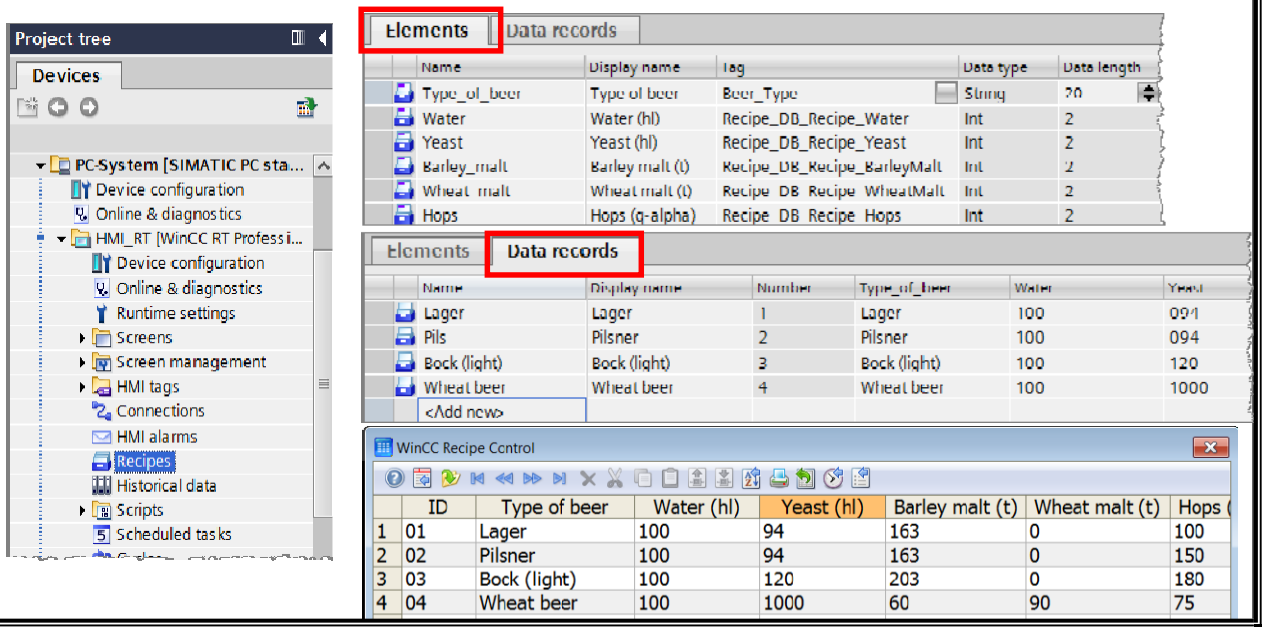
If a "recipe view" or a corresponding recipe screen is configured, recipe data records can be entered by an operator on the HMI device.

Change of recipe data records in runtime operation by operator on HMI device

If a "recipe view" is configured, additional recipe data records can be entered by an operator on the HMI device in runtime, or existing ones can be modified.

13.4. Exercise 1: Configuring a recipe

- Beer production, different types of beer → Data storage in a recipe
- Entry of recipe in the engineering step
- Data records will be editable on the HMI device



Task

The control room project will be expanded to include a recipe for beer production and a recipe screen for displaying recipes. Various data records will be displayed in the recipe view. In subsequent steps, a recipe for beer production will be created with data records for various beer types, and a recipe screen will be inserted in the control room project from a global library.
A WinCC recipe view will be added to the "Recipes" recipe screen.

The recipe or recipe view will have the following properties:

- Beer production - different types of beer will be produced
  → Data storage in a recipe
- Recipe is predefined, i.e., can be entered during the engineering step
- Data records will be editable on the HMI device
- The data record is transferred to the controller by the operator using buttons in the recipe view

Procedure

1. **Create recipe**
   → Name "BeerProduction", display name "Types of beer"
   → Communication type: Tags
2. **Create internal tag for displaying beer type**
3. **Create recipe elements**
   → Elements beer types, water, yeast, barley_malt, wheat_malt, and hops
4. **Insert "Recipes" screen from library**
5. **Configure recipe view in the "Recipes" screen**
6. **Enter data records** → see table
7. **Test the configuration**

1. Create recipe

Open the "Recipes" editor and click on the "<Add>" entry.

Recipes

	Name	Display name	Number	Vers..	Type	Maxi...	Communication type
	Beer_production	Beer types	1	23.0..	Unlimited	0	Tags

Use the following properties:

- Name: "Beer_production"
- Display name: "Beer types",
- Type: "Unlimited"
- Communication type: "Tags"

2. Create internal tag "Beer type"

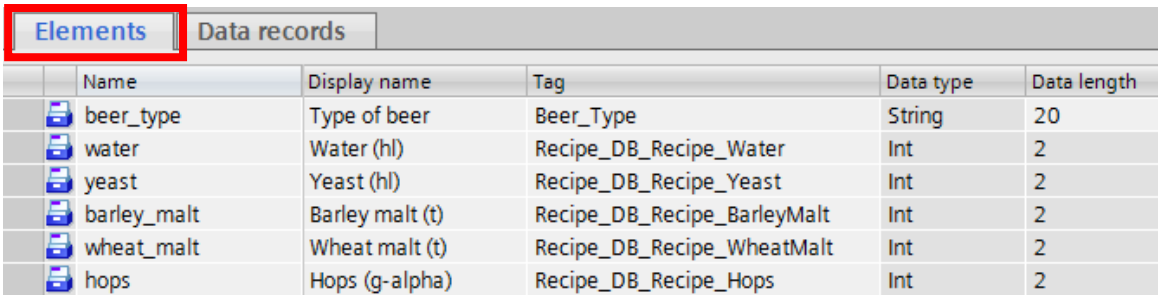
Create the tag table: I_Recipe_BeerProduction in the HMI tags editor

Configure the internal tag "Beer type" with the following settings:

- Name: "Beer type"
- Data type: "String"
- Length: "14"

This tag is used to display the beer type in the recipe view.

3. Create recipe elements



Elements | Data records

	Name	Display name	Tag	Data type	Data length
	beer_type	Type of beer	Beer_Type	String	20
	water	Water (hl)	Recipe_DB_Recipe_Water	Int	2
	yeast	Yeast (hl)	Recipe_DB_Recipe_Yeast	Int	2
	barley_malt	Barley malt (t)	Recipe_DB_Recipe_BarleyMalt	Int	2
	wheat_malt	Wheat malt (t)	Recipe_DB_Recipe_WheatMalt	Int	2
	hops	Hops (g-alpha)	Recipe_DB_Recipe_Hops	Int	2

To create the elements, double-click the next free line in the "Elements" tab, and change the default name.
Then create the elements beer_type, water, yeast, barley_malt, wheat_malt, and hops with corresponding display names.

Select the assignment of the recipe tags indicated above:

4. Insert "Recipes" screen

The "Recipes" screen is contained in the Copy templates\Screens directory of the "Library_Screens" global library. To import the screen, open this global library.

Task card >> Libraries > Global libraries > Open global library

The "Library_Screens" library is in the folder: D:\Courses\WCCSUP\Libraries. Library files have the file extension "al11".

Use a drag-and-drop operation to move the "Recipes" screen from the global library to the list of screens in your project. To do this, drag the "Recipes" screen from the global library to the Screens area of your project while holding down the left mouse button.

Add a button in the "Start" screen for selecting the recipes screen.

Use a function list for the **Press left mouse button** event for this button. Use the "ActivateScreenInScreenWindow" system function in this function list to open the "Recipes" screen ("new screen name" parameter).

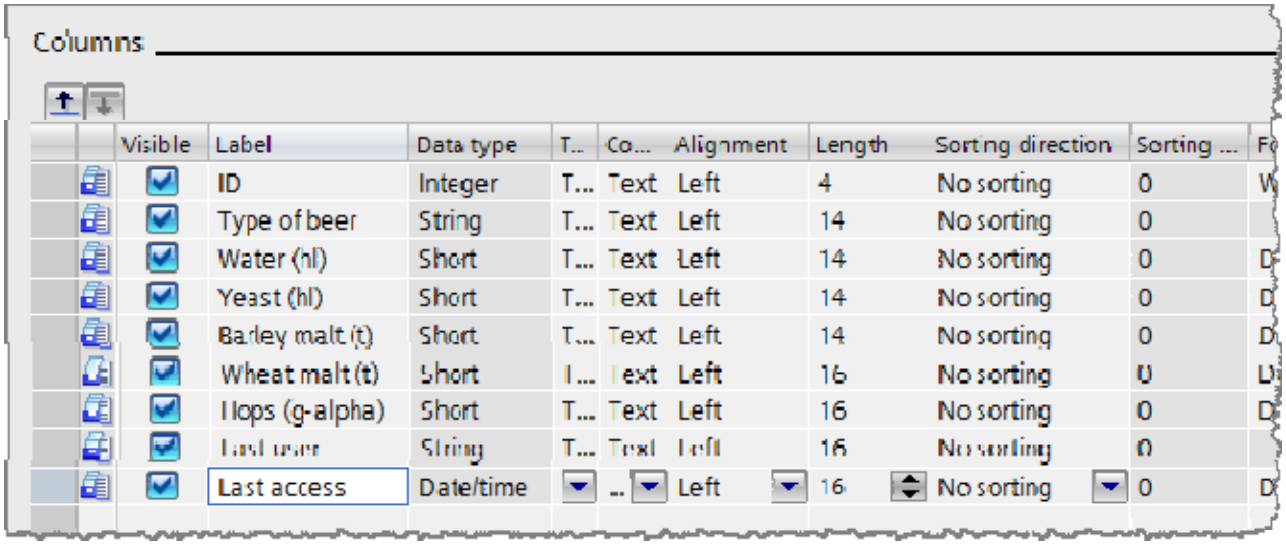
5. Configure recipe view in the "Recipes" screen

Open the "Recipes" screen, and insert the following object

Task card >> Toolbox > Controls > Recipe view

Make the following settings in the Inspector window:

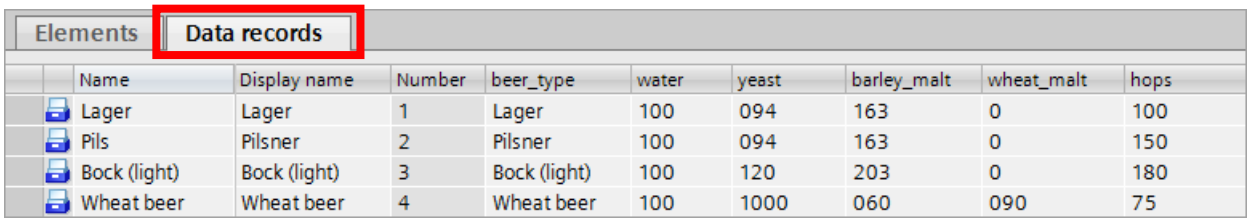
- Section General/Display:
  Data source: "Beer Production"
- Section General/Allow in runtime
  Options: Select Edit data, Insert data, and Delete data, so that recipes can be edited in runtime operation
- Layout/Position & size:
  X: 45, Y: 115, width: 1190, height: 372
- Section Columns
  Select all columns (from "ID" to "Last access")



- Section Toolbar/Toolbar - General
  Options: Select Show toolbar, Show tooltips, and Hotkeys
- Section Toolbar/Toolbar buttons
  Select all buttons using "Visible" option
- Section Status bar/Status bar - General
  Options: Select Show status bar and Show tooltips
- Section Status bar/Status bar elements
  Select all elements using "Visible" option

6. Enter data records

Open the recipe or change to the recipe editor and select the "Data records" tab.



To create the data records, double-click the next empty line and change the default name.

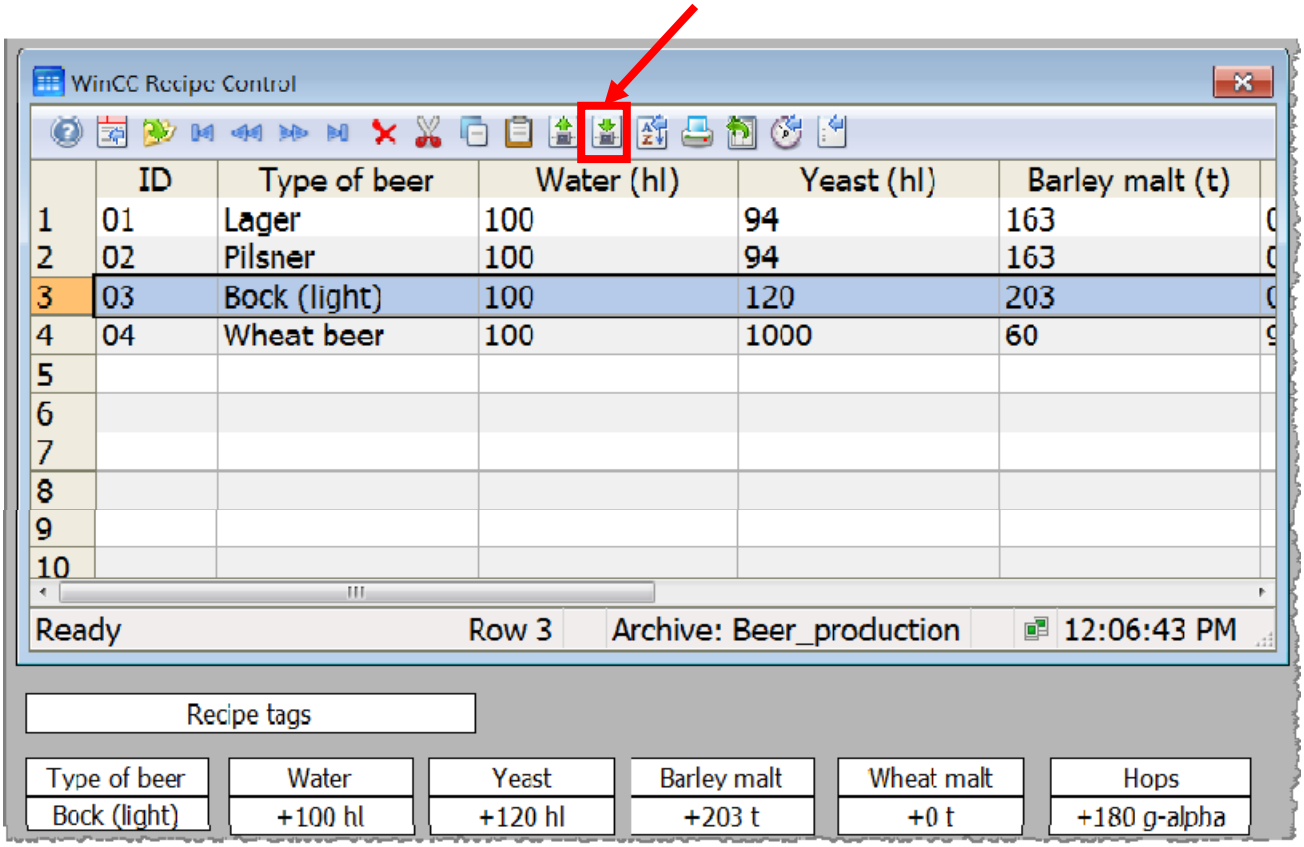
Create the data records according to the table indicated above.

Don't forget to save your project!

7. Test the configuration

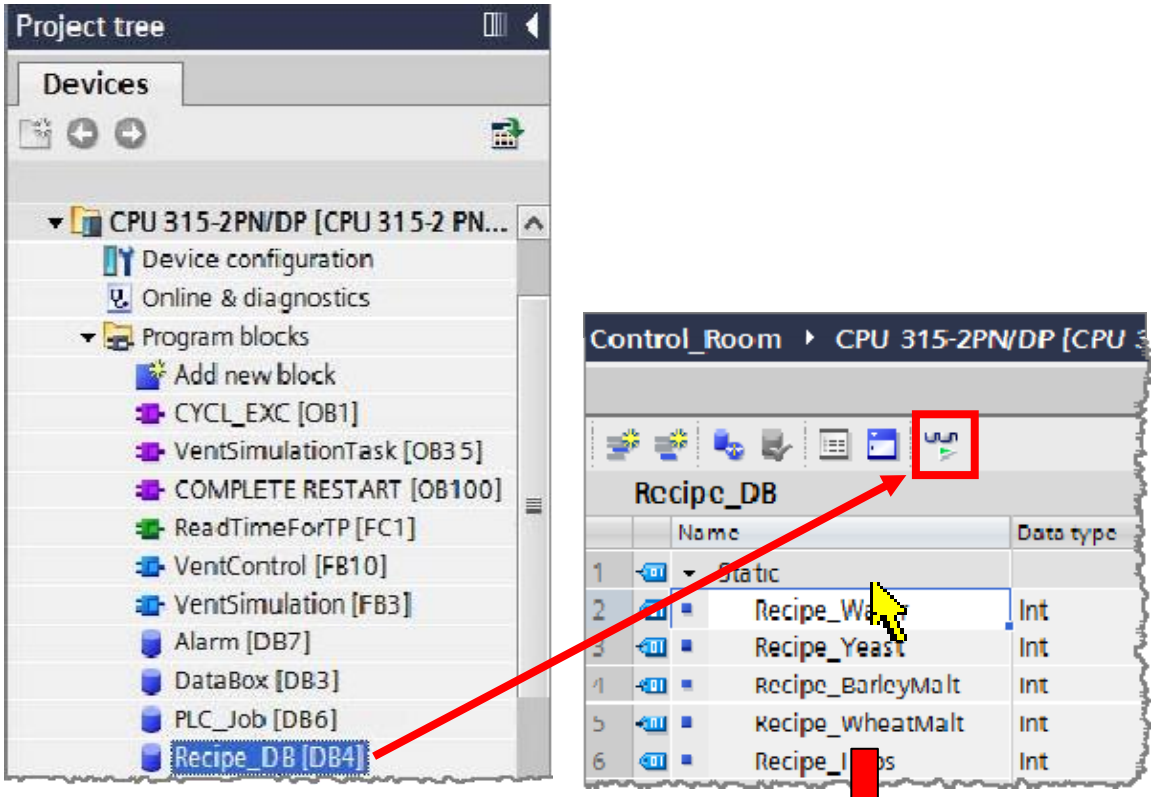
8. Test the function of the recipe view in runtime operation.

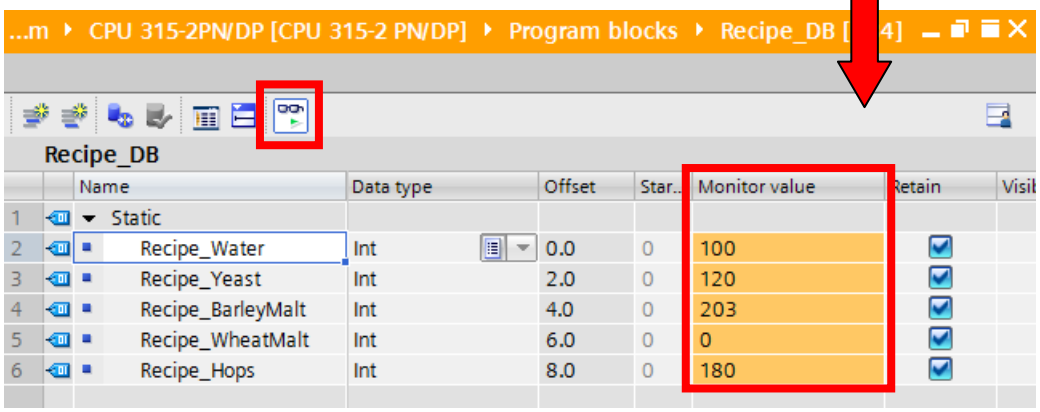
To do so, select a data record in the recipe view in the recipes screen and transfer it using the "Write tags" button. Check the transferred values in the I/O fields of the recipe tags.



Alternatively, you can check the data transfer in the DB4 data block.

To do so, double-click DB4 from the program blocks of the CPU to open it, and select monitoring mode:


Project tree
Devices
CPU 315-2PN/DP [CPU 315-2 PN...
Device configuration
Online & diagnostics
Program blocks
Add new block
CYCL_EXC [OB1]
VentSimulationTask [OB35]
COMPLETE RESTART [OB100]
ReadTimeForTP [FC1]
VentControl [FB10]
VentSimulation [FB3]
Alarm [DB7]
DataBox [DB3]
PLC_Job [DB6]
Recipe_DB [DB4]
Control_Room ▸ CPU 315-2PN/DP [CPU
Recipe_DB
Name
Data type
Static
Recipe_Water
Recipe_Yeast
Recipe_BarleyMalt
Recipe_WheatMalt
Int


...m ▸ CPU 315-2PN/DP [CPU 315-2 PN/DP] ▸ Program blocks ▸ Recipe_DB [DB4]
Recipe_DB
Name
Data type
Offset
Monitor value
Retain
Static
Recipe_Water Int 0.0 100
Recipe_Yeast Int 2.0 120
Recipe_BarleyMalt Int 4.0 203
Recipe_WheatMalt Int 6.0 0
Recipe_Hops Int 8.0 180

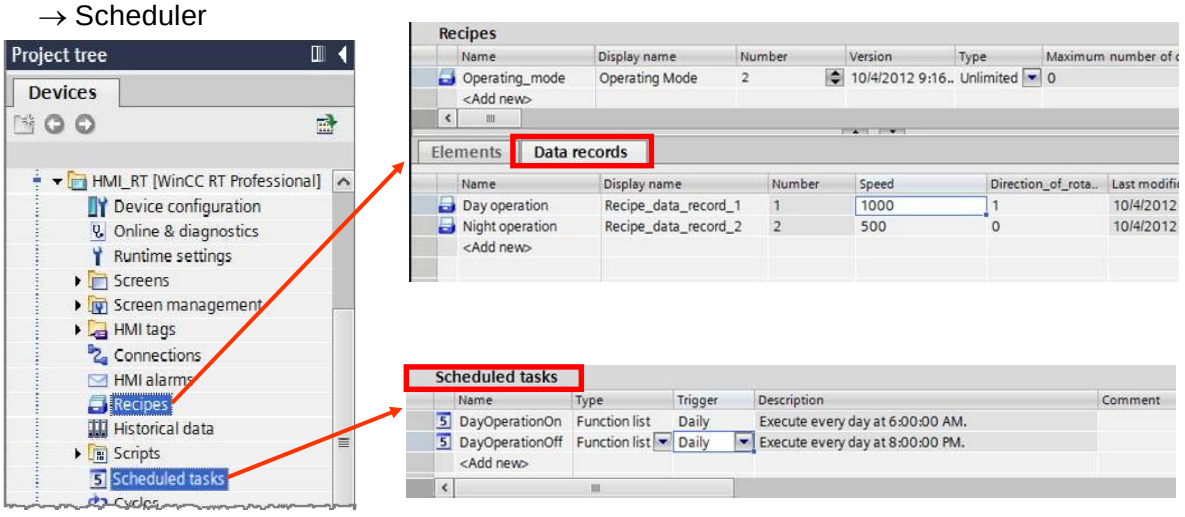

STOP

13.4.1. Exercise 2 (extra): Configuring automatic operating mode changeover

13.4.2.

Motor controller: Day operation and night operation
→ Data storage in a recipe

- Entry of recipe in the engineering step
- Automatic time-controlled changeover
  6:00 a.m. - 8:00 p.m. day operation, 8:00 p.m. - 6:00 a.m. night operation
  → Scheduler



Task

- The motor will run with two different parameter sets, depending on the time of day
  Operating modes: Day operation and night operation → Data storage in a recipe

	Day operation	Night operation
Speed	1000 rpm	500 rpm
Direction of rotation	Clockwise	Counterclockwise

- Operating mode changeover via the scheduler
- Day operation will be activated at 6:00 a.m. and night operation at 08:00 p.m.
- Recipe will already be entered during the engineering step.

Procedure

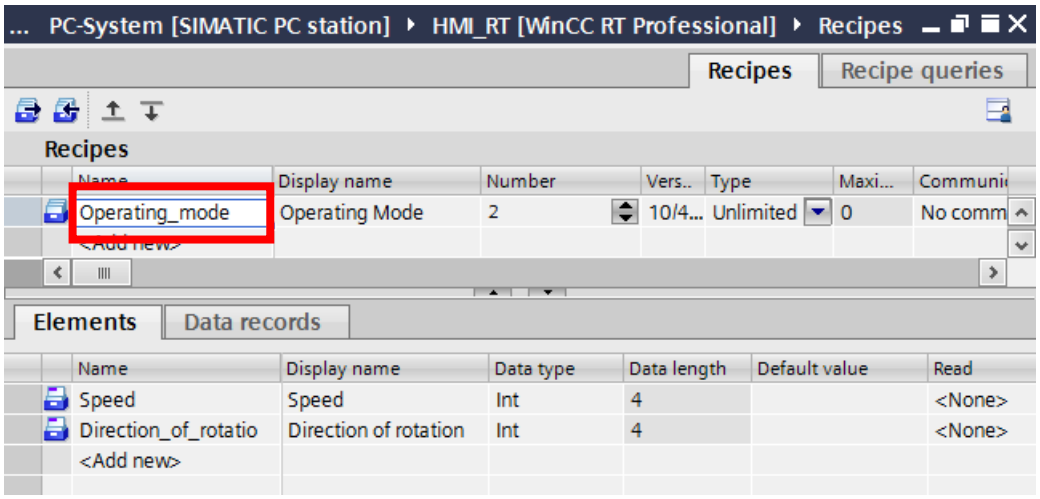
1. **Create recipe and set recipe properties**
   → Name: Operating mode
   → Communication type: Tags
2. **Insert control tags/interconnection with recipe**
3. **Create recipe elements**
4. **Enter data records**
   → Day operation, speed: 1000, right direction
   → Night operation, speed: 500, left direction
5. **Set up the data record changeover - Scheduler**
   → Day operation daily 06:00 a.m.
   → Night operation daily 08:00 p.m.
6. **Insert screen for operating mode selection**
   → **Add "Recipe_DayNight_Operation" screen for the operating mode changeover in the motor screen (in the screen window)**
7. **Test the configuration**

1. Create recipe and create recipe elements

Open the "Recipes" editor in the project tree and add a new recipe by double-clicking the "<Add>" entry.

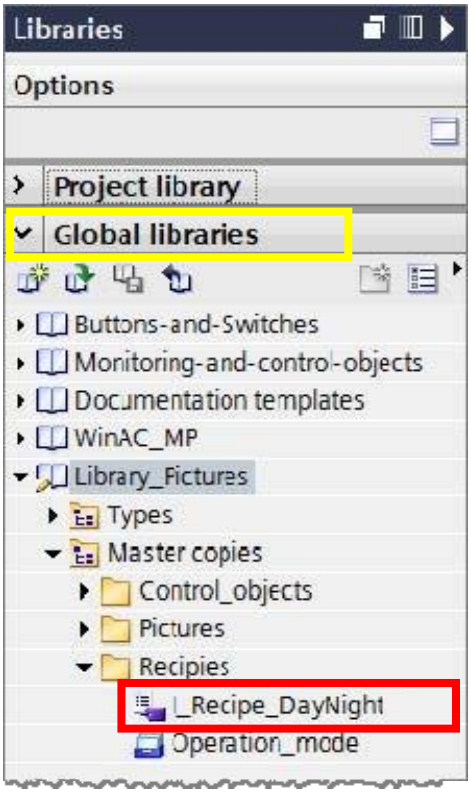
- Name "Operating_mode", display name: DayNightSwitch,
  Type: Unlimited, Communication type: Tags
- Elements: Speed and Direction_of_rotatio

To create the elements, double-click the next empty line in the "Elements" tab, and change the default name.

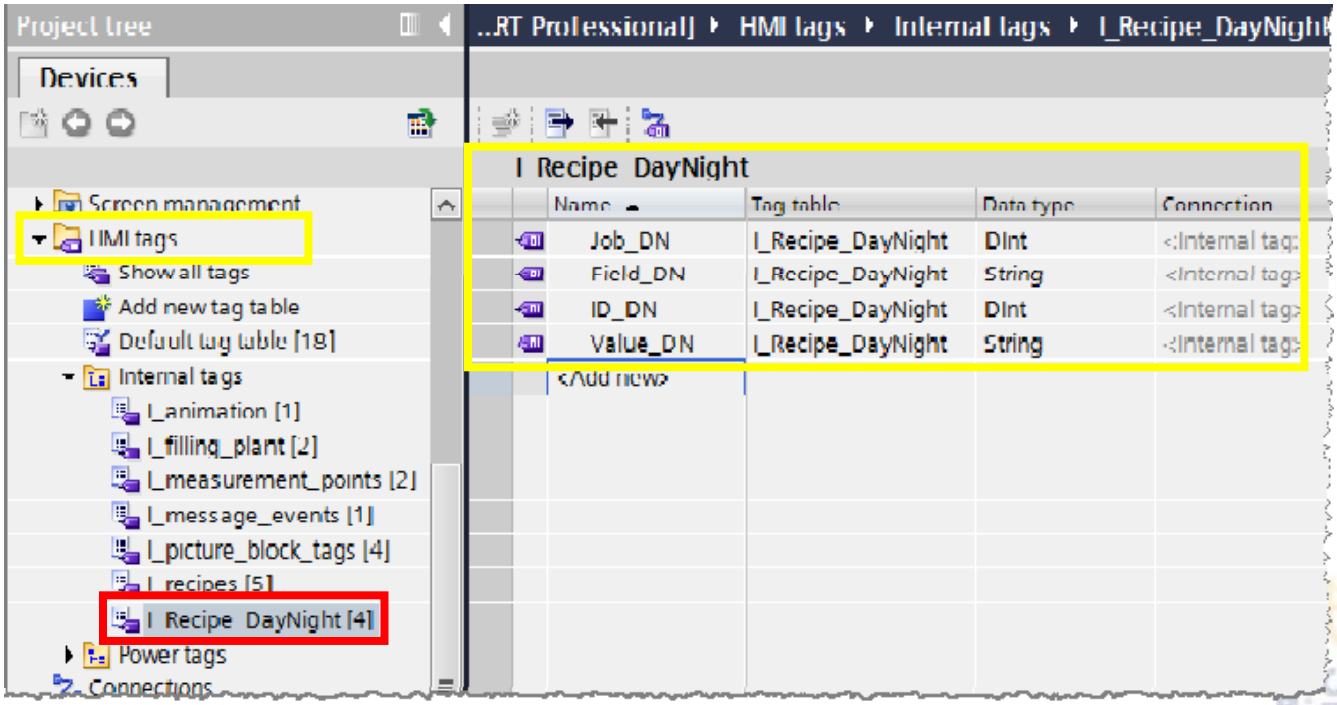


Insert two control tags from global library and interconnect with recipe

Insert the "Recipe_DayNightSwitch" tag table in the "Internal tags" tag group from the "Library_Pictures" global library. The tag table is located in the Master copies\Recipes folder.

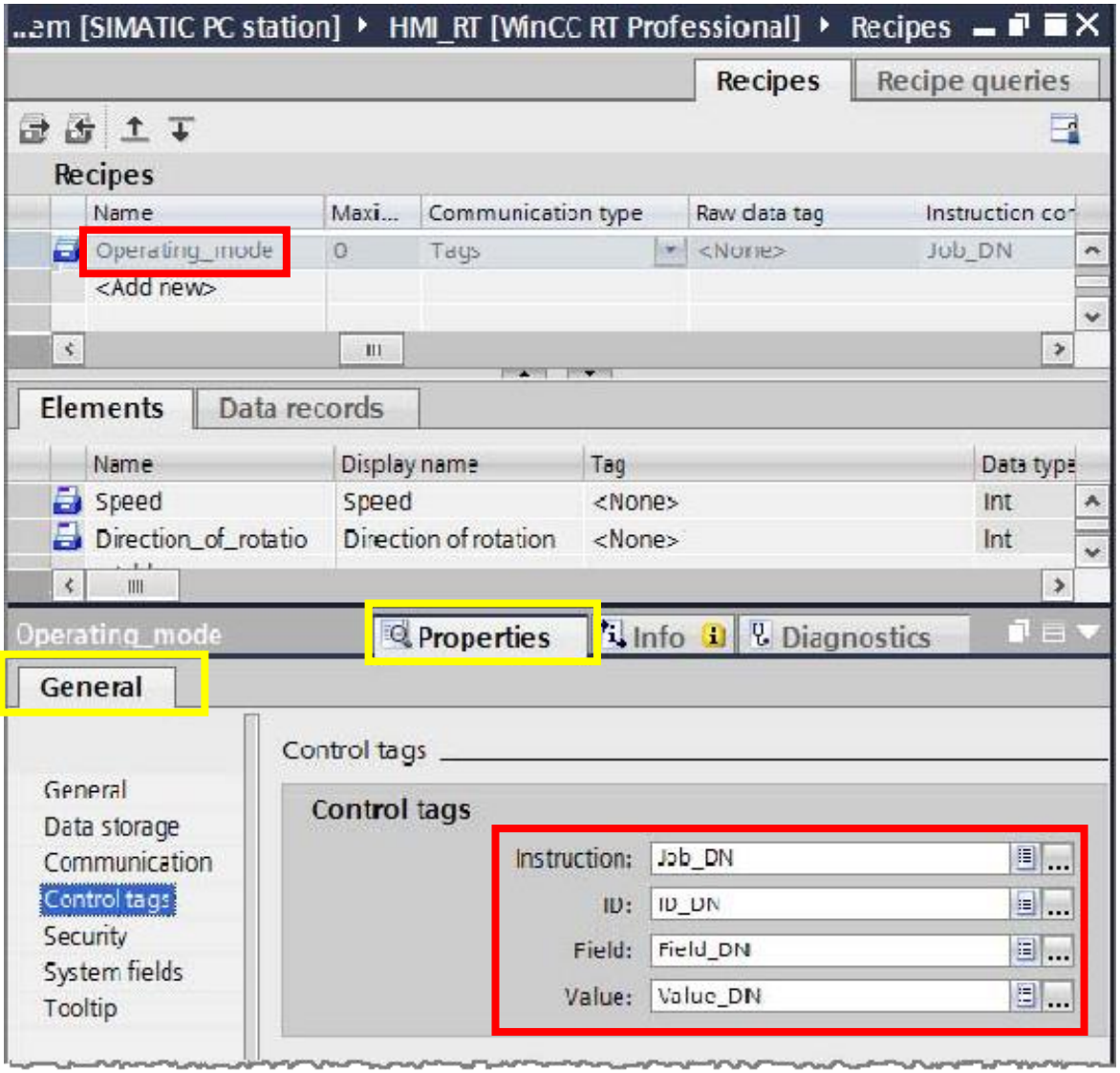


The "I_Recipe_DayNight" tag table contains the control tags that will be used subsequently for automatic changeover of the operating mode.



Interconnect the control tags with the "Operating mode" recipe as shown in the figure below.

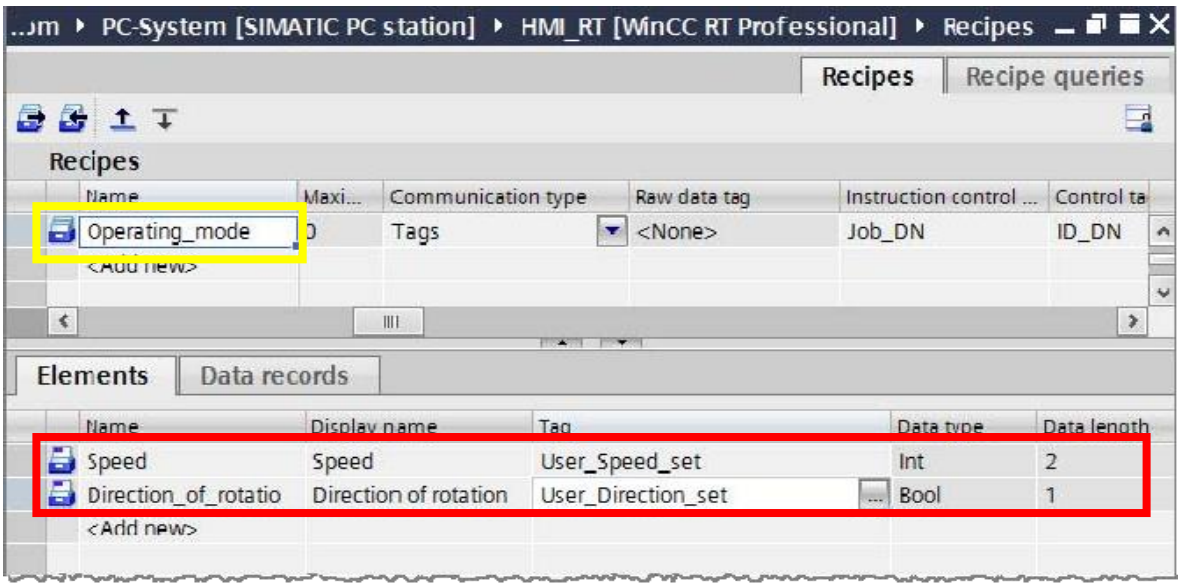
The "Instruction" control tag is used to specify whether a data record is to be written to tags or read from tags. The "ID" control tag contains the data record number.



3. Create recipe elements

Create the following recipe elements for the "Operating_mode" recipe:
- Speed
- Direction_of_rotatio

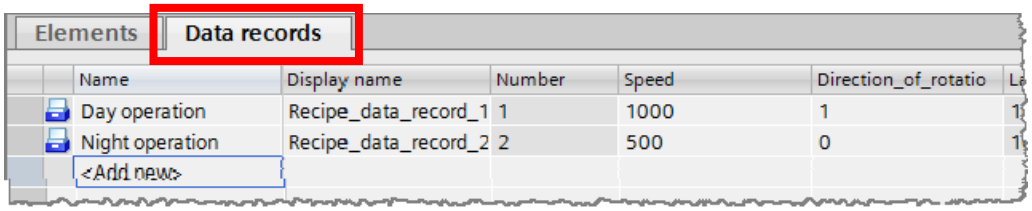
Adjust the columns “Tag” and “Data type”.



4. Enter data records

Select the "Data records" tab.

To create the data records, double-click the next empty line and change the default name.



5. Set up data record changeover in the scheduler

Automatic transfer of the data records at the defined times is implemented using the scheduler function.

>> Project tree > HMI device > Scheduler

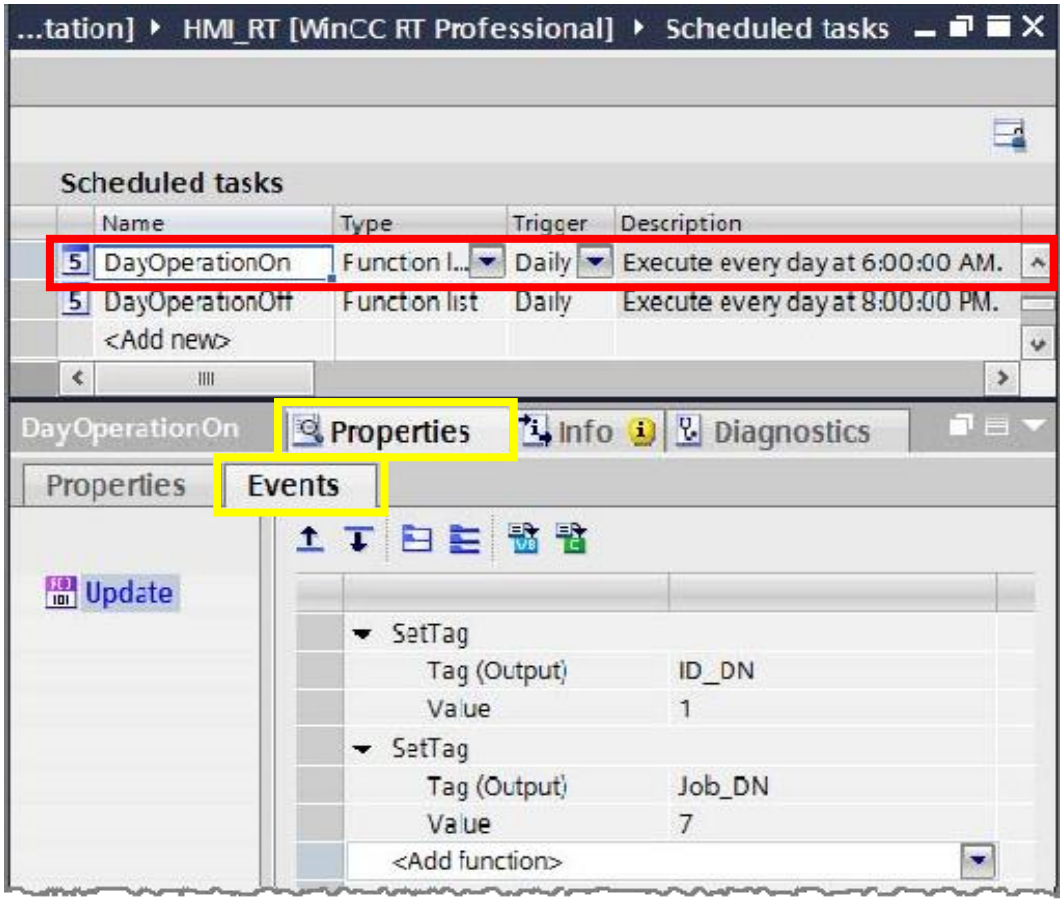
Open the "Scheduler" editor in the project tree and add a new job by double-clicking the "<Add>" entry.

Use the name "DayOperationOn" for this task.

Select the following settings for the start time:

- Trigger: "Daily"
- Run every day at: "06:00:00 AM."

Enter a function list as shown in the graphic below in order to set the control tags. The selected control tag values will be used subsequently to trigger the transfer of data record "1" (day operation).

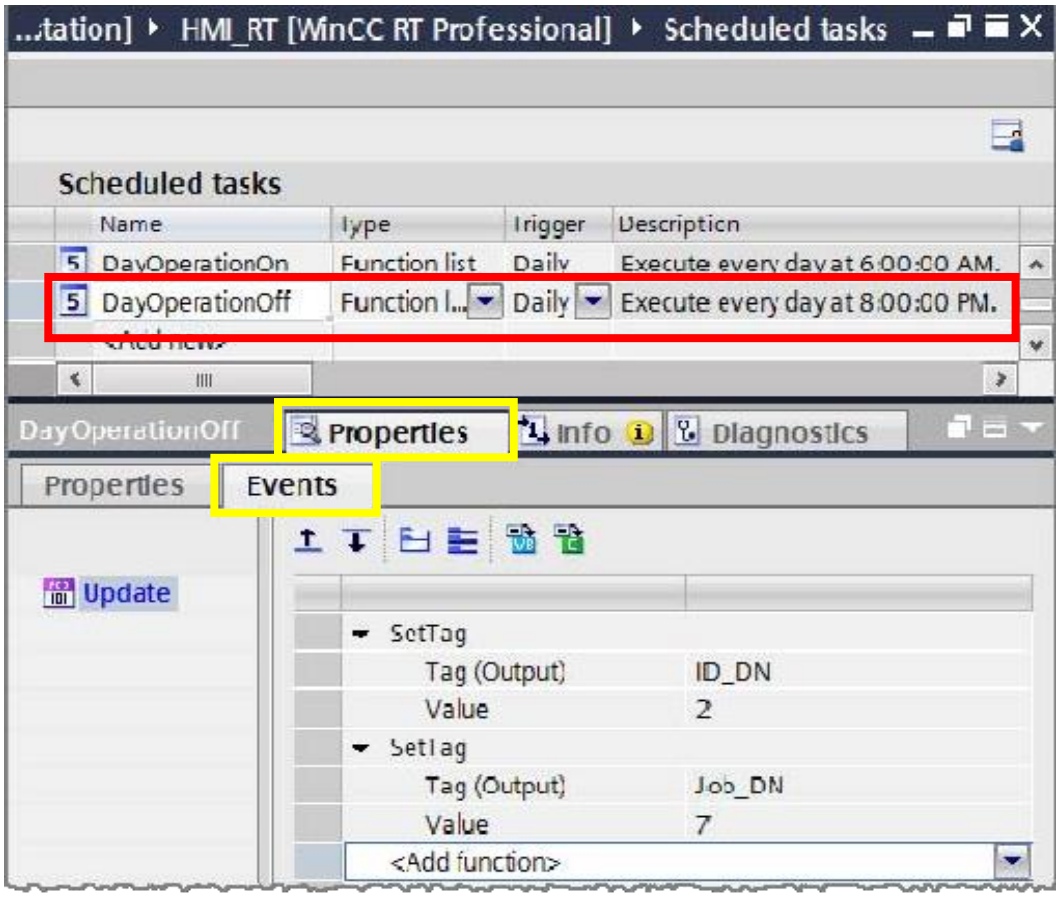


Double-click <Add> in the job table, and set up an additional job with the name "DayOperationOff".

Select the following settings for the start time:

- Trigger: "Daily"
- Run every day at: "08:00:00 PM.".

Enter a function list as shown in the graphic below in order to set the control tags. The selected control tag values will be used subsequently to trigger the transfer of data record "2" (night operation).

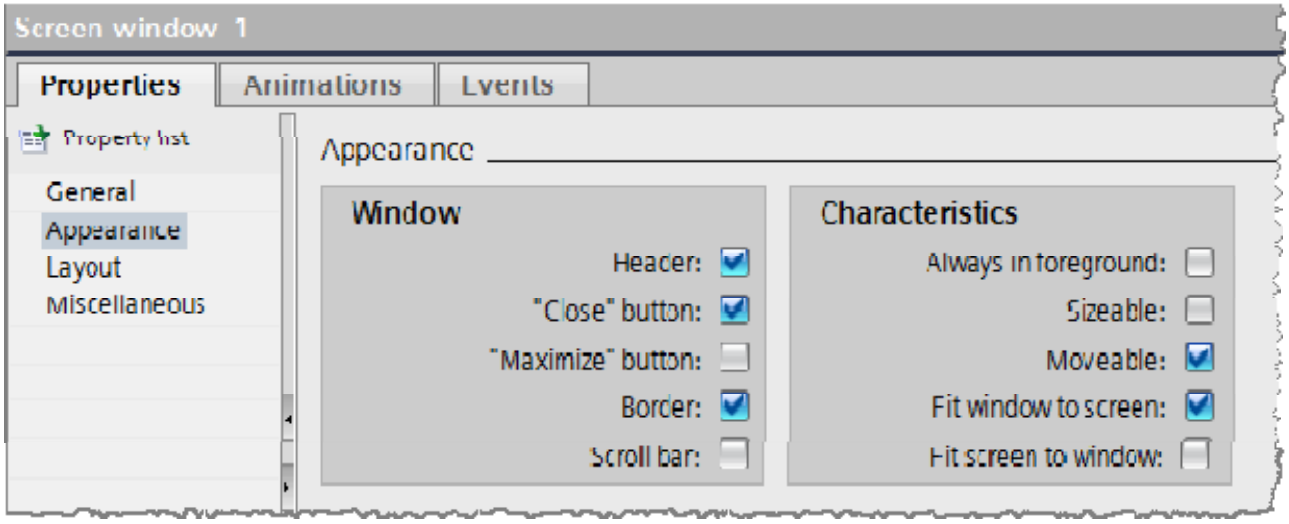


6. Insert screen for operating mode selection

To test the operating mode changeover, the "Recipe_DayNight_Operation" screen from the Copy templates\Screens folder of the "Library_Screens" global library is used.

- Insert the "Recipe_DayNight_Operation" screen in your project from the "Library_Screens" global library.
- Insert a screen window from the "Toolbox" task card in the "Motor" screen.
  >> *Toolbox > Controls > Screen window*
- Use the following settings for the screen window:
  >> Inspector window > Properties + Properties tab > General
  Section: "General":
  - Content/title: Day-night changeover
  - Displayed screen "Recipe_DayNight_Operation"

  For the "Appearance" of the screen window, select the settings as shown in the graphic below:



In the "Layout" area, choose the following values for the position of the screen window:

- X: 940
- Y: 190

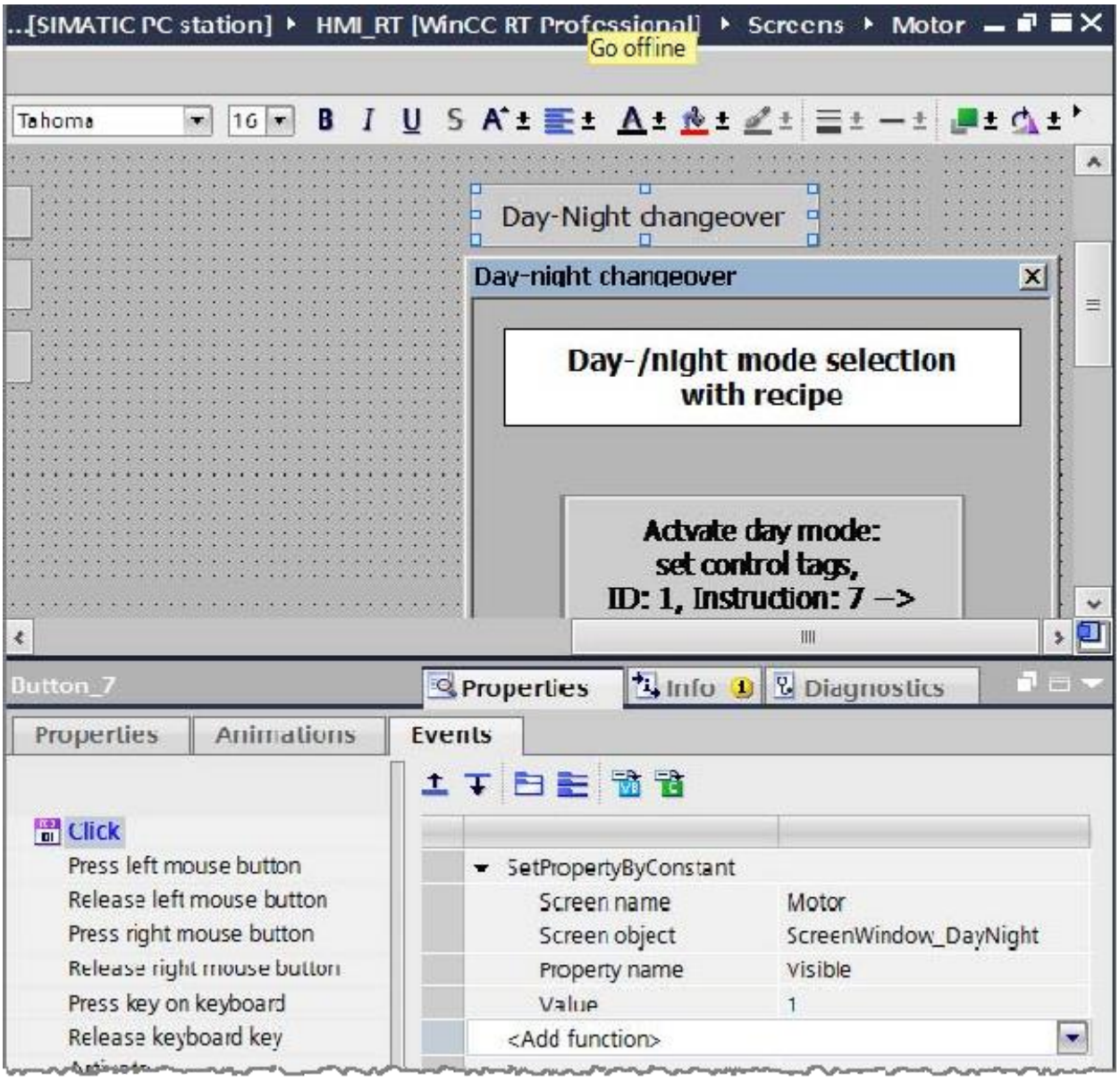
Choose the following settings under "Miscellaneous" for the screen window name:
- Object > Name: "ScreenWindow_DayNight"
- Object > Visibility: disabled

- Insert a button from the "Toolbox" task card in the "Motor" screen above the "ScreenWindow_DayNight" screen window.

  >> Toolbox > Elements > Button

  This button is used to open the "Recipe_DayNight_Operation" screen in the "ScreenWindow_DayNight" screen window.

  Configure a function list for the "Click" event for the button, as shown in the graphic below:



Don't forget to save your project!

7. Test the configuration

Test your project in runtime operation:

Switch on the motor and test the operating mode changeover using the buttons in the "Recipe_DayNight_Operation" screen or by adjusting the PC time.