**COPADATA**
do it your way

# zenon manual

## Runtime

**v.7.10**

**COPA·DATA**
do it your way

# Contents

# 1. Welcome to COPA-DATA help

### GENERAL HELP

If you cannot find any information you require in this help chapter or can think of anything that you would like added, please send an email to documentation@copadata.com (mailto:documentation@copadata.com).

### PROJECT SUPPORT

You can receive support for any real project you may have from our Support Team, who you can contact via email at support@copadata.com (mailto:support@copadata.com).

### LICENSES AND MODULES

If you find that you need other modules or licenses, our staff will be happy to help you. Email sales@copadata.com (mailto:sales@copadata.com).

# 2. Runtime

Here you will find information on the operation possibilities in the Runtime.

zenon consists of the engineering environment - the Editor - and the Runtime: Projects are created in the Editor and operated and monitored in Runtime. The Runtime is available as a 32-bit version and a 64-bit version from zenon 7.10. Projects can run on both versions. You can read details on compatibility with versions of zenon 7.10 in the Project conversion manual, in the 64-Bit version chapter.

> **License information**
>
> *The zenon Runtime must be licensed. The license is calculated according to the used TAGs or IOs. For details about licensing see chapter Licensing.*
>
> *The Runtime is available as zenon Supervisor and as zenon Operator for Embedded operating systems. The versions differ in their functionalities. For details see Restrictions for zenon Operator.*

> ⚠ **Attention**
>
> *Screen resolution: Use the standard operating system settings for the optimum display of Runtime (100%).*

## 2.1    Starting the Runtime

You can start the Runtime as follows:

- ▶ from the Editor
    - by pressing key `F5`
    - via symbol `start Runtime` in the tool barRuntime files
- ▶ via the Startup Tool
- ▶ via the Windows start menu
- ▶ using Windows Autostart.

**DIENST ZENADMINSRV**

The `zenAdminSrv` service must be started to start the Runtime. If the service is not available, you will see the "Runtime cannot be started because the 'zenAdminSrv' service cannot be located!" error message.

When starting using Windows Autostart you receive an error message "Waiting for service 'zenAdminSrv' to start". Then zenon tries to start the Runtime up until the service is available or you close the error message by clicking `Cancel`. Then the Runtime is not started.

**FUNCTION "WINDOW TO FOREGROUND"**

**The Runtime can be moved behind other running programs with the help of function `Window to the background`. With function `Window to foreground` it is moved to the foreground.** 💡 **Info**

*At moving the Runtime to the foreground the Runtime window is defined as the topmost window. At this the alarm status line is covered. In order to get the alarm status line back to the foreground, you can:*

▸ activate the system keys (deactivate project settting `Lock system keys`) and get back the alarm status line via `Alt+Tab` to the foreground

▸ activate the Windows task bar and click un window **Status**

▸ move the Runtime back to the background

▸ restart the Runtime

## 2.2 Compatibility Runtime files

The zenon Runtime is backward compatible. The Runtime can always load projects from older version and interpret and display these projects in accordance with their version.

Projects from version 6.20 SP4 on can be started directly without being converted first. Projects with a lower version number must be converted beforehand.

A mixed operation is also possible. This means: With the multi-project administration projects from different versions can be loaded and run at the same time.

**ONLINE COMPATIBILITY**

The Runtime online compatibility makes interoperability of Runtime systems (also via web clients) in the zenon network possible even if the version of the client Runtime is higher than the version of the server Runtime.

The current Runtime cam load projects of the following versions:

- `6.20 SP4`
- `6.21 SP0`
- `6.21 SP1`
- `6.22 SP0`
- `6.22 SP1`
- `6.50 SP0`
- `6.51 SP0`
- `7.00 SP0`
- `7.10 SP0`

Due to the multi-project administration projects from different versions can be loaded. For example the Integration project can have version 7.00, a sub-project version 6.51 and another sub-project version 6.22 SP1. Mixed operation also works in the network. With this different versions can also be started with the zenon web client.

Attention: Server and standby server must have the same version.

Note for Batch Control module: Runtime files for Batch Control up to and including version 7.00 SP0 are not compatible with subsequent versions. Versions from version 7.10 and later are compatible.

**COMPATIBILITY BETWEEN THE EDITOR AND RUNTIME:**

With the zenon Editor, Runtime files can be created for different versions of Runtime. The Runtime version therefore does not need to correspond to the Editor version. This backwards-compatibility is particularly suited for use of mixed systems. For example: A project that was planned with Editor 6.50 can also be started with Runtime 6.22.

> **⚠ Attention**
>
> *If, in a project with a later version of the Editor, properties are shown that are not available in the earlier version, these are not available. This can lead to unwanted results in Runtime.*

## CREATING RUNTIME FILES

To create Runtime files in the editor for earlier versions:

1. Select the project in the project tree

2. Navigate to the `General` section in project properties

3. Open the `Create RT files for` property drop-down list

4. Select the desired version:

   - `Default`: The Runtime files are created for the current version of the editor
   - `6.20 SP4`: The Runtime files are created for version 6.20 Service Pack 4.
   - `6.21 SP0`: The Runtime files are created for version 6.21 Service Pack 0.
   - `6.21 SP1`: The Runtime files are created for version 6.21 Service Pack 1.
   - `6.22 SP0`: The Runtime files are created for version 6.22 Service Pack 0.
   - `6.22 SP1`: The Runtime files are created for version 6.22 Service Pack 1.
   - `6.50 SP0`: The Runtime files are created for version 6.50 Service Pack 0.
   - `6.51 SP0`: The Runtime files are created for version 6.51 Service Pack 0.
   - `7.00 SP0`: The Runtime files are created for version 7.00 Service Pack 0.
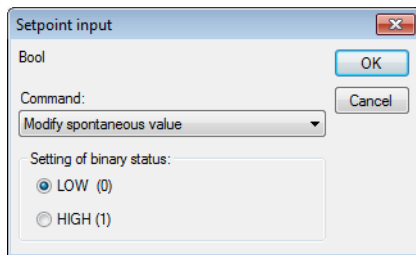
**Attention:** In order to ensure consistency of Runtime files, all Runtime files must be newly created each time this property is changed. The configurations for all drivers are converted. Settings that do not exist in the respective version are set to the default setting.

## 2.3    Operating during Runtime

Depending on the device and the engineering mouse and/or keyboard are available to you as input devices in the Runtime.
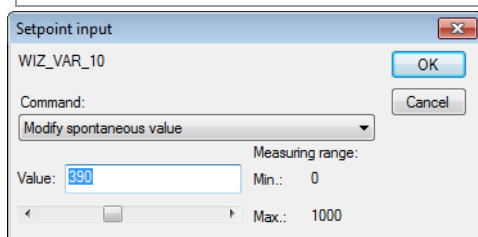
## 2.3.1    Mouse

A single click with the left mouse button an a dynamic element opens (depending on the setting of the dynamic element) the standard dialog for setpoint input. Depending on the data type of the variable the following dialogs are opened:
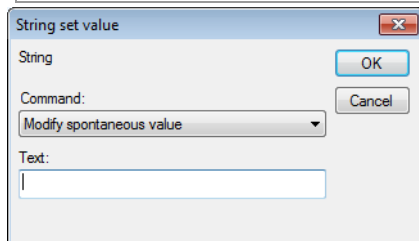


Possible operations are:

| Command | Action |
|---|---|
| Switch to spontaneous value | Display spontaneous value (value from PLC) if substitute value was displayed before or the value was turned off. |
| Modify on-change value (default) | Write new spontaneous value to the PLC |
| Switching and modes. Spontaneous value | Combination of the two commands above |
| Switch to alternate value | Display substitute value (value from variable definition), if spontaneous value was displayed before |
| Modify alternate value | Set new substitute value for variable |
| Switching and modes. Substitute value | Combination of the two commands above |
| Switch off spontaneous value | Turn off connection to the PLC for this value by setting status bit OFF (bit 20). |
| Switch on spontaneous value | Resume connection to the PLC for this value by resetting bit OFF. |
| Presetting of set status | - LOW (0)<br>- HIGH (1) |

Possible operations are:

| Command | Action |
|---|---|
| Switch to spontaneous value | Display spontaneous value (value from PLC) if substitute value was displayed before or the value was turned off. |
| Modify on-change value (default) | Write new spontaneous value to the PLC. |
| Switching and modes. Spontaneous value | Combination of the two commands above. |
| Switch to alternate value | Display substitute value (value from variable definition),if spontaneous value was displayed before. |
| Modify alternate value | Set new substitute value for variable |
| Switching and modes. Substitute value | Combination of the two commands above. |
| Switch off spontaneous value | Turn off connection to the PLC for this value by setting status bit OFF (bit 20). |
| Switch on spontaneous value | Resume connection to the PLC for this value by resetting bit OFF. |
| Value | New numerical value (within defined measuring range). |

Possible operations are:

| Command | Action |
|---|---|
| Switch to spontaneous value | Display spontaneous value (value from PLC) if substitute value was displayed before or the value was turned off. |
| Modify on-change value (default) | Write new spontaneous value to the PLC |
| Switching and modes. Spontaneous value | Combination of the two commands above |
| Switch to alternate value | Display substitute value (value from variable definition), if spontaneous value was displayed before |
| Modify alternate value | Set new substitute value for variable |
| Switching and modes. Substitute value | Combination of the two commands above |
| Switch off spontaneous value | Turn off connection to the PLC for this value by setting status bit OFF (bit 20). |
| Switch on spontaneous value | Resume connection to the PLC for this value by resetting bit OFF. |
| Text | New text |

**LASSO**

Dynamic elements which are linked with a variable or function can be pre-selected with the lasso in the Runtime and therefore by used for VBA events.

**SELECTION PER LASSO**

To select elements with the lasso in the Runtime, you must:

▶ activate property Runtime settings/Runtime lasso in the project settings

▶ activate property Runtime/selectable with lasso in the property of the dynamic element

In the Runtime several methods for selecting elements are available:

▶ Select elements: Left-click on a free area and move lasso over the screen elements while holding the mouse button pressed.

▶ Extend selecton: `Ctrl+mouse click` on an element in order to select/deselect it in addition to the other elements already selected

▶ Add elements: While spanning the lasso press and hold `Ctrl` in order to add elements to the existing selection

▶ Cancel selection: Spanning a lasso which does not contain elements.

## 2.3.2 Keyboard operation

You can use the keyboard also for the complete navigation and operation in the Runtime. At this the focus is first set to a frame. There you can navigate with the cursor keys and carry out a function with the Enter key. The control is carried out with the following functions (on page 13):

▶ Focus: set to frame

▶ Focus: Delete from frame

▶ Move focus

▶ Focus: Activate input to the element with the focus

The order in which the elements are selected is defined via the allocation of index numbers (on page 16) to the elements of a screen. The definition of the navigation order also works with **`linked symbols`**.

The keyboard operation can also be carried out with freely allocated shortcuts such as `Ctrl+M`. The shortcuts can be assigned to each operation element at the property `Key combination`.

**Functions**

For engineering the keyboard operation use the following functions:

**FOCUS: SET TO FRAME**

This function sets the focus to a defined frame when operating the keys in Runtime.

To configure the function:

1. Select, in the list of functions, in the `Screens` node, the `Set focus to frame` function
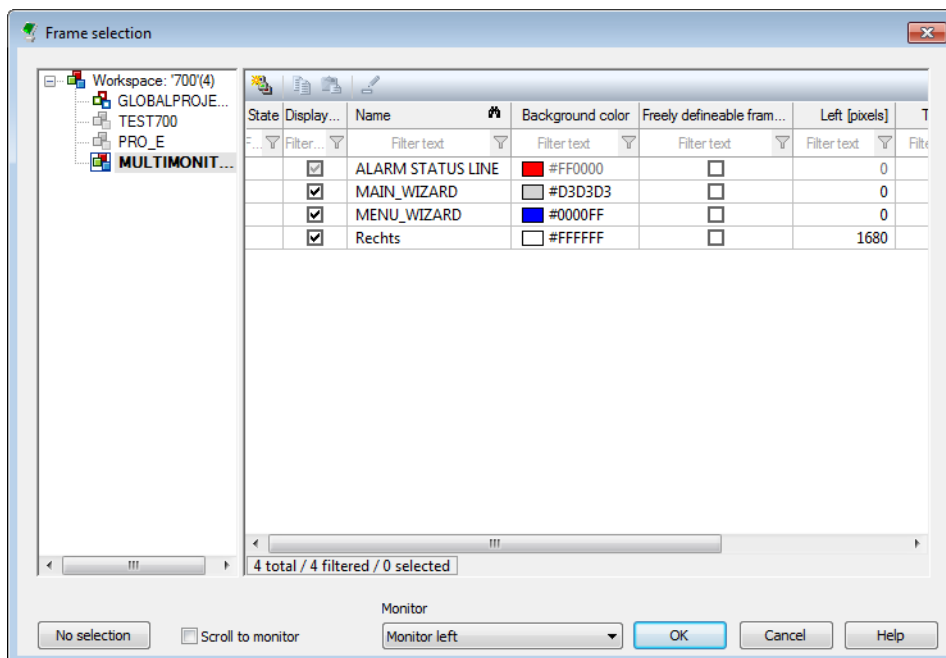
2. The dialog for selecting a screen is opened

3. select the frame you wish to assign

4. For multi-monitor projects, select the virtual monitor for opening the frame

The frame with a focus is displayed with a frame in runtime. The line width and color of the frame are defined in the `Graphical design`/`Screens` node in project properties.


## FRAME SELECTION DIALOG

In the **Frame** selection dialog, frames can be selected for the execution of functions, from:

▶ Current project

▶ Subprojects

▶ All projects in the workspace with the **Keep project in memory** option active

| Parameters | Description |
|---|---|
| `Project tree window` | Displays all projects in the workspace. Frames can be selected from the current project and from all projects with the `Keep project in memory` option active |
| `Frames window` | Selection of frame.<br><br>If several frames are selected, the first selected frame is used for the execution of the function. |
| `Open this dialog in the Runtime` | Switches to this frame selection dialog when this function is executed in Runtime. |
| `No selection` | Has no function for this selection. |
| `Scroll to monitor` | `Active`: Scroll bars are shown in Runtime, which make it possible to scroll to the monitor.<br><br>Only available with multi-monitor systems. |
| `Monitor` | Selection of the monitor from the drop-down list.<br><br>This contains:<br><br>▸ `Current monitor`<br><br>▸ All virtual monitors defined in the monitor administration Optional: `All monitors`<br><br>Only available with multi-monitor systems. |
| `OK` | Applies settings and closes the dialog. |
| `Cancel` | Discards all changes and closes the dialog. |
| `Help` | Opens the online help. |

## MOVE FOCUS

This function set the focus on a particular element in runtime with keyboard operation and can therefore be used to navigate within a frame.
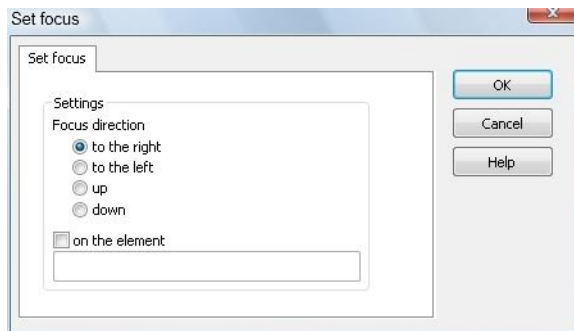
| Property | Action |
|---|---|
| Direction | Define the direction in which the focus should be moved. The following defined sequence applies in the editor in the context menu or under *Edit/change focus sequence . . .* |
| on the element | Definition of the element for the focus. Enter the object name of the element. |

**ACTIVATE INPUT TO THE ELEMENT WITH THE FOCUS**

This function activates the element that is being focused on in the frame selected.

**FOCUS: DELETE FROM FRAME**

This function takes the focus from the current frame in runtime. To continue operating the keyboard, the focus must be set to a frame again.

**Define sort order inside a frame**

You define in what sort order the elements in a screen can be selected with the keyboard when engineering the screen:

1. open the screen

2. right-click an empty area

3. select `Shift order for focus` from the context menu

4. select from the drop-down list `Order for left/right` or `Order for up/down`

5. each element in the screen is displayed with a number in the top left corner

6.  click on the element which should be selected first

7.  the number of the element changes to **1**

8.  click on all other elements in the order you want them to be selected

    - Elements which have already been defined change the background color of the number

    - Elements which cannot be selected with the help of the keyboard in the Runtime (property `Focus is set to the element inactive`) are displayed with a red number

    - if you click on an element while holding `Ctrl`, the numbering starts with the index number of this element

> 💡 **Info**
>
> *In the properties of the elements the position is displayed via the properties* `Focus position left/right` *and* `Focus position top/bottom`. *These properties are for information only. The index values of the position cannot be changed there.*

**SYMBOLS**

An index can also be assigned to **linked symbols**. During compiling the Runtime files, the symbol is disjointed and its elements are inserted. Thereby the order for selecting the elements which was defined in the symbol is taken into consideration.

> ▣ **Example**
>
> There are three elements in a screen: the numerical value **N**, the button **B** and the links symbol **LS**. The linked symbol from the project library receives the three numerical values: **N1**, **N2** and **N3**.
>
> If you define the order in the screen as: **B** - **LS** - **Z**,
> the order in the Runtime is: **B** - **N1** - **N2** - **N3** - **N**.

**Lock Windows shortcuts**

You can lock Windows shortcuts such as `Alt+Tab`, `Alt+Esc` or `Alt+F4` in the Runtime:

In order to lock these shortcuts, deactivate property `Lock system keys` in area `Runtime settings` of the project properties.

| Locked are: | Not locked are: |
|---|---|
| ▸ `Alt+Esc` | ▸ `Ctrl+Alt+Del` |
| ▸ `Ctrl+Esc` | |
| ▸ `Alt+Tab` | |
| ▸ `Alt+F4` | |
| ▸ Key `Windows` | |
| all corresponding mouse actions | |

To lock all systen keys in the Runtime:

- ▸ Activate the `Lock system keys` property

- ▸ start the Runtime via Keyblock Runtime Start

- ▸ consider the hints in chapter **protect Runtime files**

## 2.4 Runtime profiles

A Runtime profile is a reproducible optical snapshot of the screen in the Runtime including information about:

- ▸ Screen pattern (succession and position)

- ▸ Monitor allocation

- ▸ Filter

Runtime profiles are project- and user-related.

The following screen types support Runtime profiles:

- ▸ Alarm Message List

- ▸ Archive revision

- ▶ Chronologic event list

- ▶ Extended trend

- ▶ Industrial Maintenance Manager

- ▶ Industrial Performance Analyzer

- ▶ Report Viewer

- ▶ Message Control

- ▶ Variable diagnosis

Every zenon user has an own folder in which he can administrate his profiles. Administrators can see and edit all profiles of all users and copy them to user `System`. From user account `System` all users can load Runtime profiles. It serves as an exchange account for Runtime profiles.

**Note:** Only administrators an copy to and manage profiles in user account `System`. All other users can only manage their own profiles and load the profiles of user `System`.

**PREDEFINED PROFILES**

Besides the individual profiles there are two pre-defined Runtime profiles:

- ▶ `DEFAULT`: created by the user and selected as standard profile

- ▶ `LAST`: is automatically saved to the folder of the user when he logs out
  **Exception:** Profile `LAST` is not saved:
  - • for user `System`
  - • a temporary login without writting permission

To each zenon user a start profile (on page 23) can be allocated during log in.

> ⚠ **Attention**
>
> *Profiles can only be saved correctly at the Server if all projects are available at the Server:*
>
> ▸ from which screens are called up at the Client
>
> ▸ which call up screens with a screen switch function
>
> Saving Runtime profiles which do not match these conditions can lead to errors in the Runtime.

## 2.4.1 Load and create profile in the Runtime

In order to create a profile in the Runtime you must engineer a respective function (on page 20):

- ▶ Save profile: saves the current profile as it was defined in the function

- ▶ Profile administration (on page 23): opens the administration of the profiles for saving, loading, allocating and administrating profiles

In addition already existing profiles can be activated in the Runtime:

- ▶ Load profile

When loading a profile, the same state as when the profile was saved is restored. The screen structure including all filters as it was when the profile was saved is restored.

**Attention:** If there are screen switch functions defined at the properties of a screen for functions `Start function` or `End function` - such as (**Screen switch, Index screen** or **Screen back** ), they are not carried out! All other function types are carried out again.

If in those two functions a script is called, the script is carried out completely. Including screen switch functions if there are any. Therefore it can happen that the screen structure after loading the profile does not match the screen structure when saving the profile.

A similar behavior displays variable `Screen active variable`: If a limit which contains a screen switch function is carried out in the variable, the function is carried out and may affect the display in an undesired way.
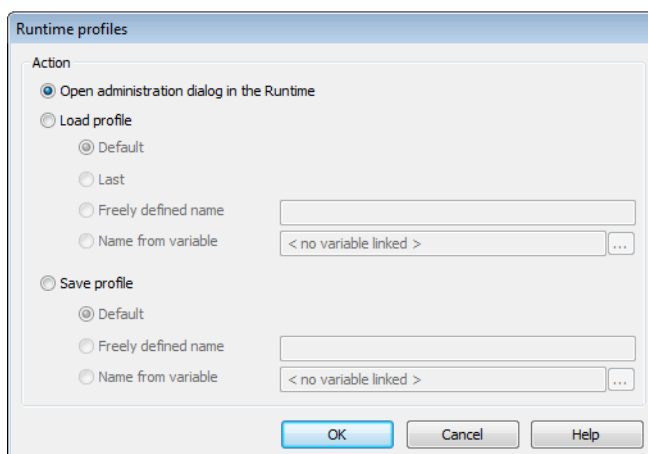
**Configuring functions**

In order to load, save and administrate profiles in the Runtime, you engineer a function in the Editor and assigned it to a button:

- ▶ select New Function

- ▶ navigate to node **Screens**

▶ select `Runtime profiles`
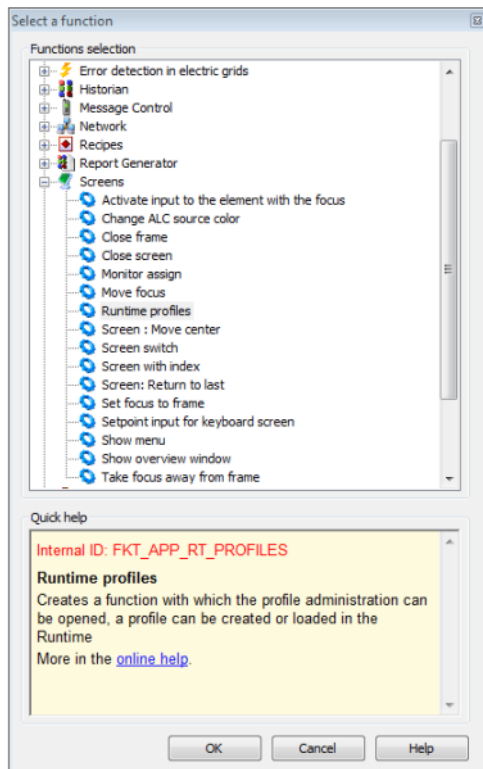


▶ the dialog for configuring functions is opened

| Property | Action |
|---|---|
| `Open administration dialog in the Runtime` | Opens the dialog for administrating (on page 23), creating, saving and loading profiles in the Runtime. |
| `Load profile` | Loads the profile defined in the dialog in the Runtime: |
| `Default` | Loads the profile DEFAULT.<br>It was selected as the standard profile by the user from all existing profiles. |
| `Last` | Loads the profile LAST.<br>It was automatically written to the folder of the user at log out. |
| `Freely defined name` | Loads the profile with the name which was defined in the dialog. If the profile does not exist, the current profile is kept. |
| `Name from variable` | Loads the profile whose name was generated from the defined variable. |
| `Save profile` | Saves the profile which was defined in the dialog. |
| `Default` | Saves the current profile as DEFAULT. |
| `Freely defined name` | Saves the profile with the name which was defined in the dialog. |
| `Name from variable` | Saves the profile under the name which was created from the variable defined in the dialog. |

## 2.4.2    Export and import profiles

With the control elements for import and export you can transfer Runtime profiles to other projects and other computers.

To export profiles:

1. create the control elements for import and export in the screen:
   *Control elements -> Filter profiles -> Import or Export*

2. start the Runtime

3. open the screen

4. create the desired profiles

5. save the profiles

6. export the profiles:

   these are saved in an XML file and can be imported at the same screen type in another project

To import profiles:

1. save (on page 27) the XML file with the desired profiles at the Runtime computer if it is another computer as the export computer

2. start the Runtime

3. open the screen

4. import the profiles

   ### 💡 Info

   *At the import all existing profiles are deleted. The profile active at the time is not changed.*

   *XML files can only be imported in the screen type in which they were created. If you try to import profiles of other screen types, the import is canceled and an error message is displayed.*
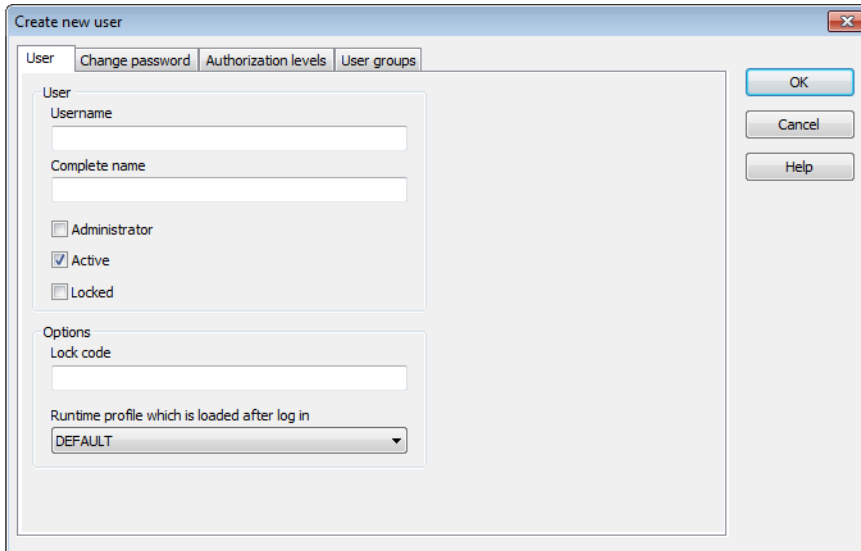
## 2.4.3 Allocate and administrate profile

Profiles can be allocated in the Editor and in the Runtime. In the Runtime you can create and save them.

**IN THE EDITOR**

In the Editor you can allocate Runtime profiles to users:

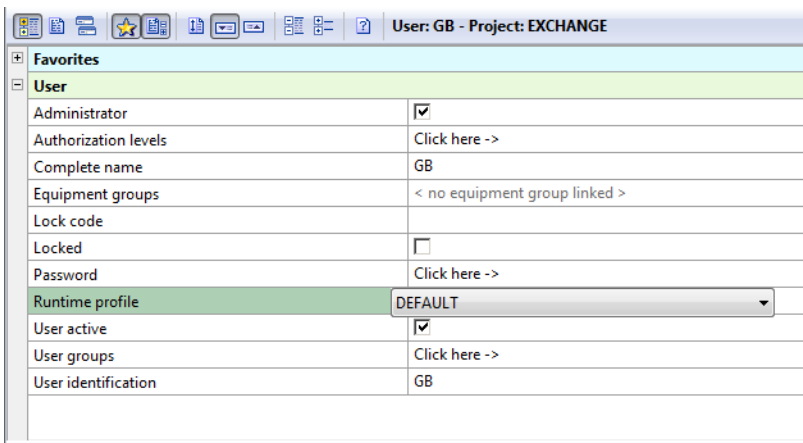▶ in the dialog for creating a new user with the property `Runtime profile which is loaded after log in:`



▶ Via the property `Runtime profile` of a user:



▶ In the drop-down list the following settings are available:

- `None:` No profile is allocated (default setting)

- `DEFAULT:` Profile `DEFAULT` is allocated

- `LAST:` Profile `LAST` is allocated

## IN THE RUNTIME

In the Runtime you can save, load, allocate and administrate profiles with the help of the profile administration. For this you must engineer function (on page 20) `Runtime profiles` with property `Open administration dialog in the Runtime.`

**Note:** Administrators can administrate the profiles of other users.
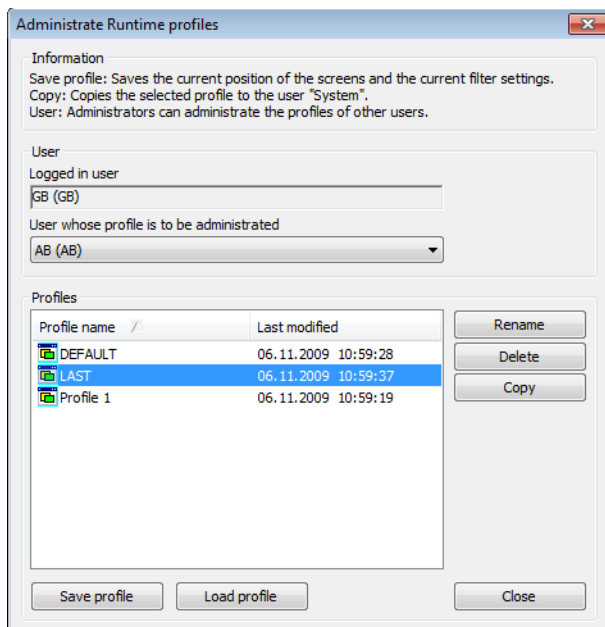
| Property | Action |
|---|---|
| `Logged in user` | User who is currently logged in to the system. |
| `User whose profiles are administrated` | User whose profiles are displayed and can be administrated. |
| `Profiles` | List of the available profile. |
| `Profile name` | Name of the profile. |
| `Last modified` | Date and time of the last modification. |
| `Rename` | Opens the selected profile name in order to rename it. |
| `Delete` | Deletes the selected profile after a confirmation message. |
| `Copy` | Copies the selected profile to the user `System`. At this the name can be changed. **Hint:** Profiles at user `System` are available for all users which are not logged in. |
| `Save profile` | Opens the dialog for assigning a profile name and saving the profile under this name. The following characters are forbidden: `Space` and the special characters `/ \| \ : * ? ! " < > $` |
| `Load profile` | Loads the selected profile. |
| `Close` | Closes the dialog. |

**KEYBOARD SHORTCUTS**

| Key | Action |
|---|---|
| `F5` | Updates the profiles' list. At unexpected events in the zenon network such as the loss of server or standby the contents of the list are automatically updated. |
| `Esc` | Closes the dialog. |
| `Input` | Loads the selected profile. |
| `Ins` | Saves the current state of the new profile. |
| `F2` | Makes it possible to change the name of the selected profile. |
| `Del` | Deletes the selected profile after a confirmation message. |

## 2.4.4 Storage directories of the profiles

The profile data are stored differently depending on their use in the Runtime:

| Type of Runtime | Storage directory |
|---|---|
| `Server or standalone:` | Data are stored locally. |
| `Client` | Data are stored on the server and are requested and changed interactively. |
| `Standby` | Data are stored on the server and are requested and changed interactively and are synchronized locally. |

# 2.5 Handling of date and time

Date and time in zenon are either local time, UTC time or time periods. The basic principle is that:

▸ all times "`from - to`" in filter such as screen switch to AML or CEL are saved in UTC

▸ Times in modules such as PFS and EMS are made in local time

▸ Time periods are in seconds

Exceptions:

▸ the IPA writes historic data in local time as "`datetime`" in the database

▸ **`Read time from variable`** or **`write to variable`** does not save times but takes over the local time formatted as string from the control or writes it to the control

**UTC**

UTC stands for Coordinated Universal Time. The time unit is SI-second. UTC is the uniform basis for international time and is provided publically via time broadcasters and other time services. Depending on the time zone, a certain time gap must be added to or subtracted from the UTC. This time gap can vary by an hour if there is a change to summer time by an hour. Examples:

| Country | Local time |
|---|---|
| Alaska | `UTC − 9` |
| Australia, Queensland | `UTC +10` |
| Bulgaria | `UTC +2` |
| United Kingdom | `UTC` |
| Korea | `UTC +9` |
| Central Europe (CET) | `UTC +1` |
| Central European Summer Time (CEST) | `UTC +2` |
| Saudi Arabia | `UTC +3` |
| USA, East Coast | `UTC − 5` |
| United Arab Emirates | `UTC +4` |

**EFFECTS ON CONFIGURATIONS**

Projected times have different times results, depending on the actual execution position of the runtime. For example:

In the Editor, the value `14.00` for the execution of a function is configured locally in the time zone `UTC +1` for the PFS. After the transfer of the files to a Runtime in the time zone `UTC +10`, this function is executed at `23:00`.