zenon manual

Keyboard

v.7.11
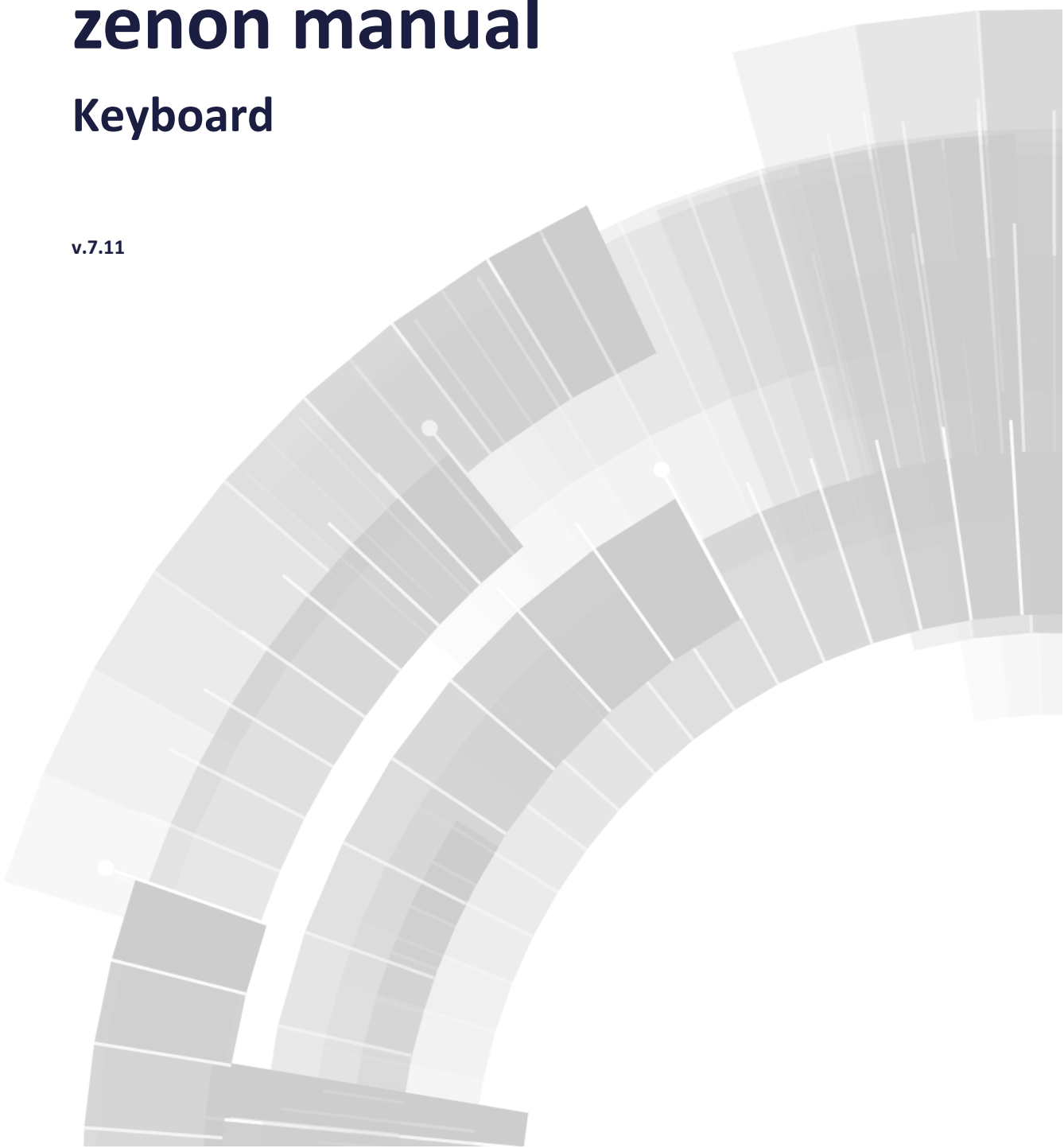
COPADATA
do it your way

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. Keyboard

Keyboards serve with zenon as virtual keyboards that allow inputs on devices without a hardware keyboard in the Runtime. Keyboards are projected with a screen type `keyboard`. zenon provides automatically generated keyboards that can be engineered by the user.

License information

Part of the standard license of the Editor and Runtime.

AUTOMATIC KEYBOARD

Automatic keyboards (on page 9) are generated by the system e.g. for setpoint inputs and are automatically loaded. They cannot be changed by the user. Self-projected keyboards can be freely defined.

Each dynamic element and each value in the Recipe group Manager has its own automatic Keyboard.

TASKBAR IN THE RUNTIME

If a keyboard is loaded in the Runtime, it stays open even if the Runtime is minimized. In the OS task bar, a drop down list for the keyboard is shown.

⚠ **Attention**

Individually adapted screens of type `Keyboard` *may under certain circumstances not work properly with the Mozilla Firefox browser.*

Reason: Mozilla Firefox loads new windows in the background without putting the focus on them. Individually adapted keyboard screens are however closed when they lose the focus.

Solution: Use another browser such as Microsoft Internet Explorer, Apple Safari or Google Chrome.

Automatic keyboards are not affected!

HARDWARE KEYBOARDS AND VIRTUAL KEYBOARDS USED TOGETHER

Virtual keyboards act like hardware keyboards in the operating system. If virtual keyboards are used together with hardware keyboards, they affect each other. Settings on the hardware keyboards - e.g. determine whether Shift is used - affect the virtual keyboard and vice versa.

EXAMPLES:

▸ If Capps lock is pressed on the hardware keyboard, it is also true for the virtual keyboard. In this case you cannot enter numbers via the automatic keyboard.

▸ If the `Num` key on the hardware keyboard is pressed in order to switch of the number pad, the number pad on the virtual keyboard also does not work.

3. Creating a screen of the type keyboard

To create a screen of the type keyboard:

1. choose the command screen new in the Project Manager´s context menu in the node screens

2. choose screen type keyboard in the drop down list of the new screen´s detail view

3. choose in the group's properties `General` as `Name` a predefined Keyboard or define an unallocated name:

   Attention: Keyboard screens with reserved names have priority over self-defined names

4. add the desired keyboard elements from the menu control elements

5. **Tip:** If a a screen type keyboard is called up on the whole screen instead of `Modal dialog`, some menus may be obstructed. In that case, project a function `Close frame` which allows you to close the open keyboard-frame.

> ⚠ **Attention**
>
> *Use the control element* `Setpoint input` *only for keyboards which are created for the input of setpoints (SETVALUEKBD, SETSTRINGKBD). It is automatically connected with the variable of the dynamic element that shall get the set value. When using a normal Keyboard, unwanted results may occur.*
>
> *It is imperative for Keyboards that are used for setpoint input to have the control element* `Setpoint input` *projected in the screen. If not, it is impossible to enter values or to send set values*

KEY LABELING

The labeling of the Keyboard can be either freely defined or defined by the operating system. To do so, activate or deactivate in the screen properties `Automatic labeling` in node `General`. For this, the following applies:

1. Active:

   - In the Editor, characters, numbers and f-keys can no longer be changed. It remains possible to label function keys such as shift or tab individually.
   - The labeling of the keys is taken over by the settings of the operating system in the Runtime.

2. Inactive:

   - The labeling of the keys can be adapted in the Editor.
   - In the Runtime the alternative label is displayed.

3. The keypad of the keyboard is not automatically labeled and the label can be adapted in the Runtime at any time independent of the setting of property `Automatic labeling`.

RESERVED NAMES

For screen type keyboard, the following names have been reserved:

Name	Description
DIALOGKBD	if a picture with this name exists, it will be opened in the Runtime every time a dialog is loaded, e.g. for entering a new recipe name in the RGM. Attention: The DIALOGKBD is not designated to be used with control element Set value element and not suitable for it. The use of this combination can cause errors in the Runtime.
SETBOOLKBD	if a picture with this name exists, it will be opened in the Runtime every time an input for binary variables is required.
SETSTRINGKBD	If a picture with this name exists, it will be used in Runtime instead of the standard dialog box for string variables.
SETVALUEKBD	If a picture with this name exists, it will be used in Runtime instead of the standard dialog box for numeric variables. The control element set value element must be inserted in the keyboard. Attention: When Caps lock is activated, you cannot enter anything in a keyboard of type SETVALUEKBD as instead of numbers the respective special characters would be entered. Exception: French keyboards need the Shift key or Caps lock for entering numbers.

Keyboard screens with reserved names have priority at Runtime over other keyboard screens, including the screen Automatic keyboard (on page 9) for touch operation.

⚠ **Attention**

Is in the project

▸   a screen of type Keyboard with the reserved name DIALOGKBD available

▸   and at the same time of type Login

▸   and is property Automatic keyboard in group Keyboard is active,

the screens must not be based on the same frame.

In this case, the DIALOGKBD would be used automatically.

4. Keys for Recipegroup Manager

Under   *Control elements -> RGM specific* you can find the following elements which are available for the Keyboard in the recipe group manager:

Element	Description
`Send value`	The set value is sent to the variable and the keyboard is closed
`Save recipe`	The recipe is saved and the keyboard is closed.
`Send value and save recipe`	The set value is sent to the variable, the recipe is saved and the keyboard is closed
`Value displays as Text 1` to `Value displays as Text 32`	Keys can be linked with limits from a reaction matrix. For this the corresponding variable ▸ must be linked with **`Numeric reaction matrix`** or **`String reaction matrix`** and ▸ `equals` states must be included In the Runtime, used keys are shown. Clicking the button writes the linked value as recommendation for the recipe value in the input window.

5. Automatic keyboard

Automatic keyboards are created by the system and are automatically displayed in the Runtime during input. They cannot be changed by the user. If a keyboard with the reserved name DIALOGKBD exists, it is displayed instead of the automatic keyboard.

> ⚠ **Attention**
>
> *There are no automatic keyboards available for the profile control elements.*
> *Solution: Create your own keyboard and activate it using a button. It is therefore also*
> *possible to enter a profile name using your own, manually-activated keyboard.*

DEFINE POSITION

For the automatic keyboard you can define a `Favored position` and supplementing a `Alternative position`. You make the settings for this in the properties of the project in area `Keyboard`.

If there is not enough space for the keyboard at the preferred position, it will be displayed at the alternative position. If there is also not enough space there, the position will be defined by the system. You define the position with the help of four values.

Property	Description
`Horizontal`	`Left` (default) positions the left corner of the keyboard in relation to the top or bottom left corner of the element. `Right` positions the right corner of the keyboard in relation to the top or bottom right corner of the element.
`Horizontal movement [pixel]`	Horizontal space (in pixels) to the selected element for the preferred position of the automatic keyboard. Positive values move the keyboard to the right, negative values move it to the left. Default: `2`.
`Vertical`	With `top` (default) or `bottom` you position the keyboard above or below the element.
`Vertical movement [pixel]`	Vertical space (in pixels) to the selected element for the preferred position of the automatic keyboard. Positive values move the keyboard upwards, negative values move it downwards. Default: `2`.