End-user manual for custom widget definition

Custom widget usage

Using already defined custom widget is easy, a user needs to go to Add Widget wizard, choose custom widget from the list of widgets, and choose the controller (widget type), DIAP, PLC and Tag (the last one depends on the mapping type, it's needed only for single tag mapping). By default, the widget is a Real Time widget, and a user can switch to Historical view or switch on the "Allow custom dates filter", which allows to choose a different time interval on the dashboard (Image 1). By clicking on "Save changes" the widget is defined and instantiated on the dashboard.

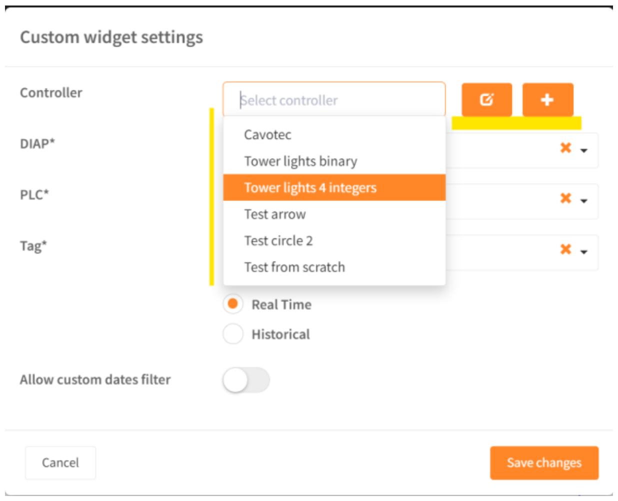


Image 1 – Custom widget settings

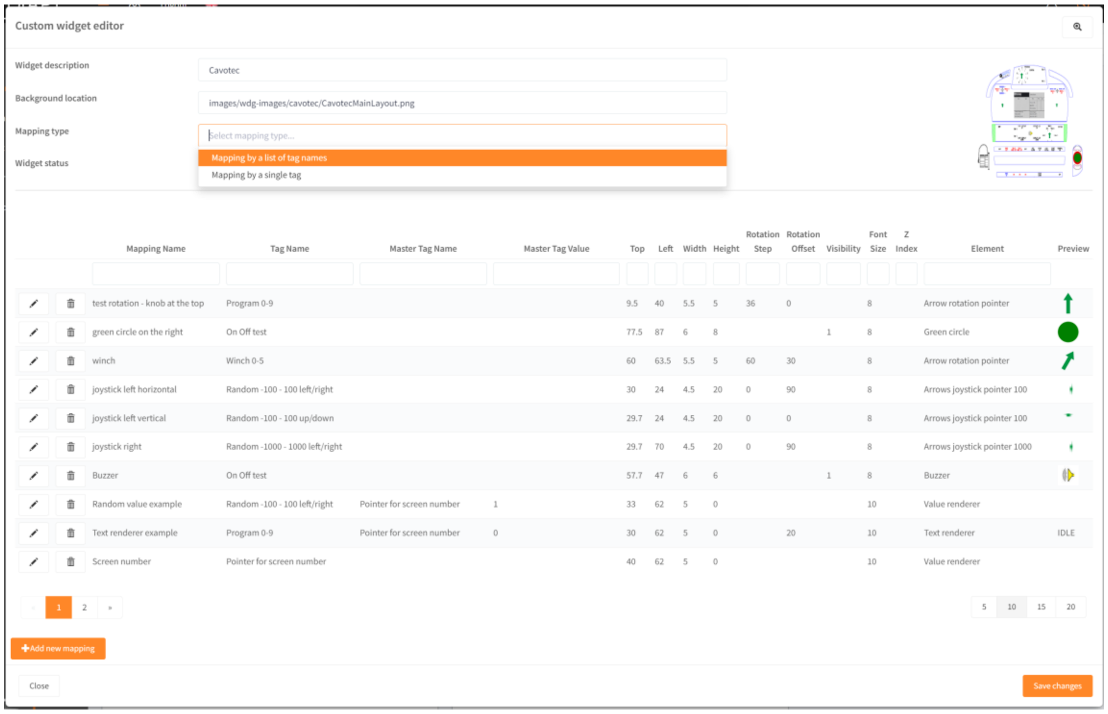
Custom widget definition

Editing or creating a custom widget is available from the widget settings by clicking on one of the two icons respectively. For editing, a user must select the widget (controller).

On the next image (see Image 2), the upper part defines the general widget information.

- Widget description – which can be a name of widget/machine/visualization, this information is presented on the custom widget settings' controller drop down list.

- Background location – a relative path to the background image which will fill the widget background. It is possible to select an image from an image repo (see “Image repo functionality”), previously uploaded in the system, or it can be set as a URL for some public image resource. A type-ahead control is implemented, and it will show a list of possible images as soon as a part of the name is typed (min 1 character). The same functionality is implemented on other image resources in the editors.
- Mapping type – a list of 2 mapping types implemented:
  - Mapping by a list of tag names – usually used for complex visualization, where the widget elements are mapped by tag name. That gives an option to only select the PLC without needing to map every single element. But, on the other side, the tag names must fit these mappings to join the elements.
  - Mapping by a single tag – The tag will be selected from the widget settings window – tag selector on image 1.
- Widget status – Active or inactive – the defined widget will be or won’t be available in the dropdown list on the custom widget settings (image 1). The default is active.
- Widget instant preview – displayed on the upper right corner with possibility to zoom / expand on the screen.


Custom widget editor

Widget description	Cavotec
Background location	images/wdg-images/cavotec/CavotecMainLayout.png
Mapping type	Select mapping type...
Widget status	

Mapping Name	Tag Name	Master Tag Name	Master Tag Value	Top	Left	Width	Height	Rotation Step	Rotation Offset	Visibility	Font Size	Z Index	Element	Preview
test rotation - knob at the top	Program 0-9			9.5	40	5.5	5	36	0		8		Arrow rotation pointer	↑
green circle on the right	On Off test			77.5	87	6	8			1	8		Green circle	●
winch	Winch 0-5			60	63.5	5.5	5	60	30		8		Arrow rotation pointer	↗
joystick left horizontal	Random -100 - 100 left/right			30	24	4.5	20	0	90		8		Arrows joystick pointer 100	♦
joystick left vertical	Random -100 - 100 up/down			29.7	24	4.5	20	0	0		8		Arrows joystick pointer 100	▾
joystick right	Random -1000 - 1000 left/right			29.7	70	4.5	20	0	90		8		Arrows joystick pointer 1000	♦
Buzzer	On Off test			57.7	47	6	6			1	8		Buzzer	▶
Random value example	Random -100 - 100 left/right	Pointer for screen number	1	33	62	5	0				10		Value renderer	
Text renderer example	Program 0-9	Pointer for screen number	0	30	62	5	0		20		10		Text renderer	IDLE
Screen number	Pointer for screen number			40	62	5	0				10		Value renderer	

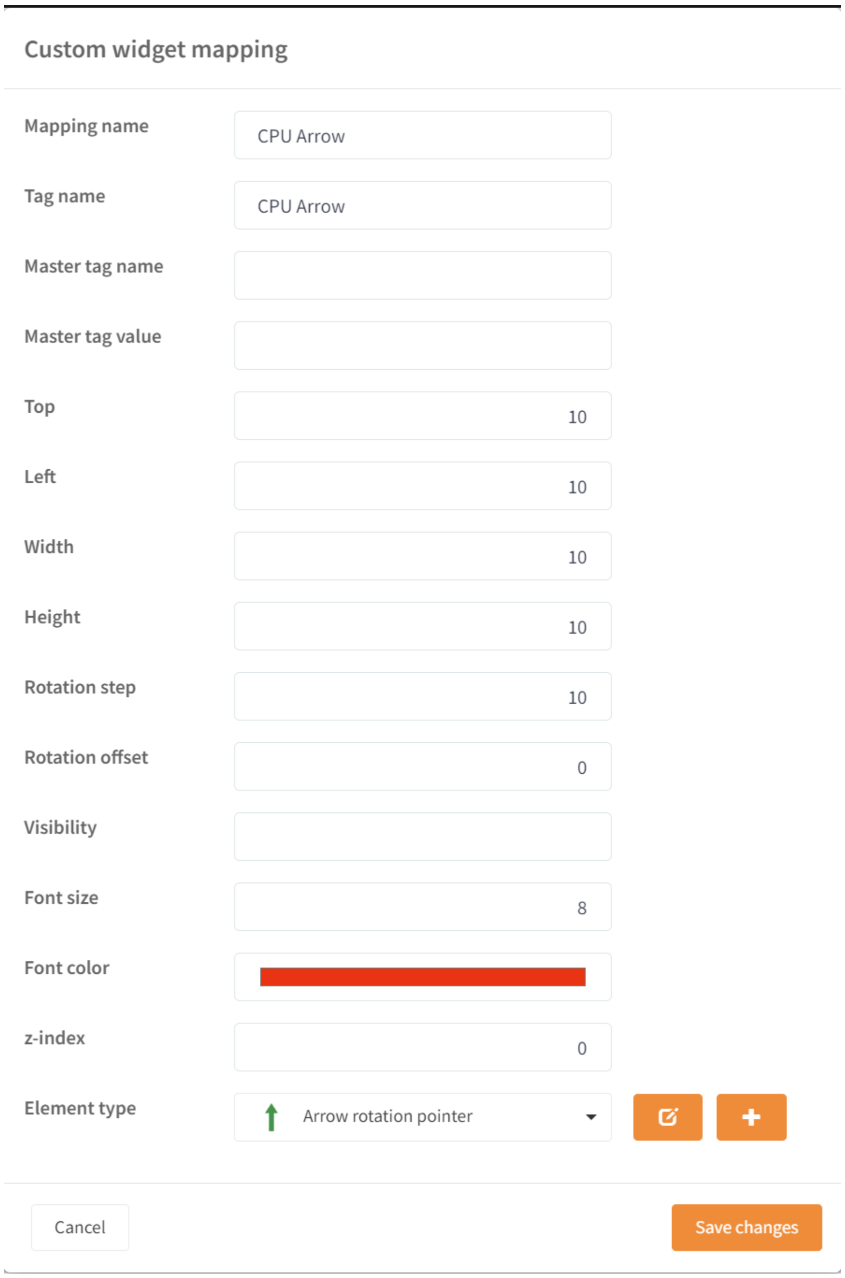
Image 2 – Custom widget mappings

- List of mappings defined on the widget. The list is displayed in a table format that contains the major properties of the item, with an image/text preview of the selected element.
- If mapping by a single tag is chosen, only one mapping is necessary and allowed to be defined.

Custom widget mapping

The user can create a new widget mapping or edit an existing mapping by clicking on the correspondent button. The application will show a popup window with the editable properties and an option to choose a visualization element, which can be a multiple type of image or text or value renderer. The UI is shown in image 3. Here is the list of the parameters:

- Mapping name – custom field that describes the mapping, not important in the mapping run-time engine.
- Tag name – If the widget is defined as "Mapping by a single tag", this field is also custom field, as the settings UI will have a function to connect the tag (image 1). But, for "Mapping by a list of tag names", the engine will match the tag names from the selected PLC with this definition, so, it must be matched/paired by name.
- Master tag name/master tag value properties – these properties can define a conditionally shown/hidden element based on other tag's values. For instance, if some "control tag" has value 1, the element will be shown, if it has value 0, the element will be hidden. The parameters are optional.
- Top, left, width and height of the element – they define the position and size of the element on the screen. The unit is percentage of the widget width/height. The example on image 3 defines an element shown on position 57.7% from the left, 47% from the height, and width and height of 6% of the widget.
- Rotation step and rotation offset – the parameters are meaningful for rotation objects – the elements that can be rotated depending on the tag value (knobs, pointers, arrows etc.). The rotation will be applied with formula Tag value * Step + Offset.
- Visibility - The parameter is optional, and the element will be shown if the tag's value matches the defined value or hidden if there's mismatch. If not set, the element is always shown.
- Font size – meaningful only for text/value renderers
- Font color – a color picker which defines font color of the rendered text.
- Z-index – html z-index of the element, null/0 is the default value, setting negative value will push the element in the background of other elements, positive value will pull the element to the front.
- Element type – list of possible elements for the mapping – different elements defined by the system administrator. The elements can be edited, which will have a global impact on already defined widgets, and there's an option for definition of new global elements (next chapter).


Custom widget mapping

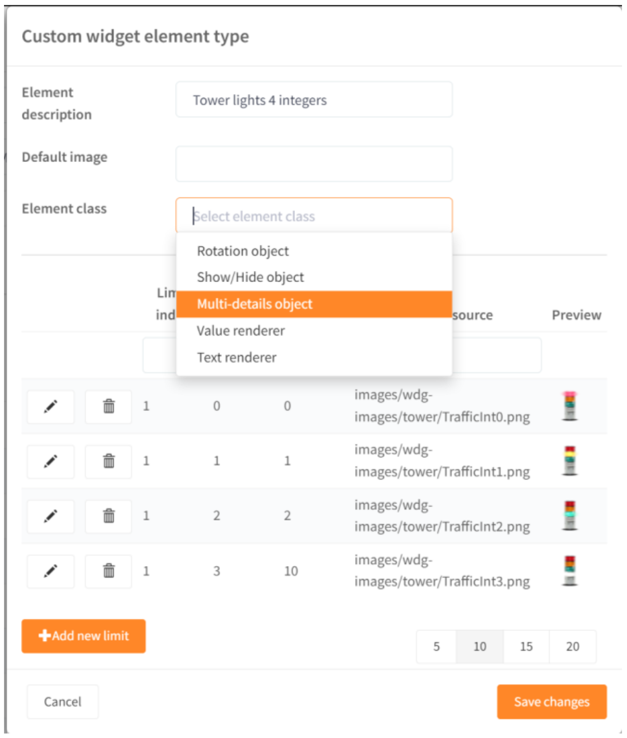
Mapping name	CPU Arrow
Tag name	CPU Arrow
Master tag name	
Master tag value	
Top	10
Left	10
Width	10
Height	10
Rotation step	10
Rotation offset	0
Visibility	
Font size	8
Font color	🟥
z-index	0
Element type	Arrow rotation pointer

Image 3 – Editing a custom widget mapping

Custom widget element

There's a possibility to define a new or edit an existing visualization element. Every element has the following properties (shown on the image 4):

- Description – This is a custom field that will represent the element in the element type picker on the mapping UI.
- Default image – it is a default path to the image that will be displayed on the widget, or a text if the element is textual. Type-ahead is possible (see Image repo functionality)
- Element class – It is a list of possible element classes that are implemented. Here's the list:
  - Rotation object – The main purpose of the class is that an element can be rotated by a tag value.
  - Show/Hide object – The main purpose of the class is that the element can be conditionally shown or hidden on the widget.
  - Multi-details object – The main purpose is that the element will be displayed with different images depending on the tag's value.
  - Value renderer – The element will render a tag value.
  - Text renderer – The renderer will display a different text based on a tag value.

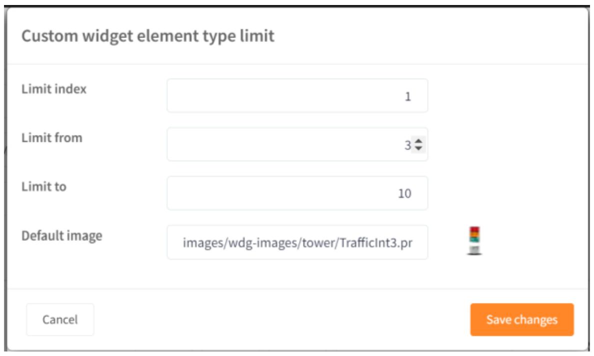

Custom widget element type

Element description	Tower lights 4 integers
Default image	
Element class	Select element class

		Lim ind			source	Preview
		1	0	0	images/wdg-images/tower/TrafficInt0.png	
		1	1	1	images/wdg-images/tower/TrafficInt1.png	
		1	2	2	images/wdg-images/tower/TrafficInt2.png	
		1	3	10	images/wdg-images/tower/TrafficInt3.png	

Image 4 – Definition of element

- List of limits – There's a possibility to define a different image that will be shown depending on the tag's value. The first entry that matches the interval defined with [limits from, limit to] will be displayed on the widget. Limit index defines a priority of the rules. In case of same indices, the order of definition stands. It's good practice to set distinct values for indices.

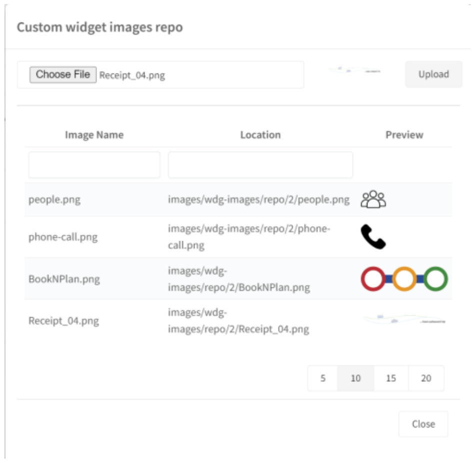

Custom widget element type limit

Limit index	1
Limit from	3
Limit to	10
Default image	images/wdg-images/tower/TrafficInt3.pr

Image 5 – Element types limits

Image repo functionality

Image repo functionality is accessible from the custom widget definition window. Here, a user can upload all images needed for widgets and widget elements. Upload is a simple option, click on Choose File button, or use a built-in drag and drop functionality to drag an image to the input field and press Upload. All uploaded images are shown as a preview. It is worth mentioning that in order to have a simple interface, uploading an image with the same name will overwrite the existing image with new content. Of course, if different clients have files with the same name, they will have a separate copy of the file. A type-ahead control is implemented in all fields that references images (widget definitions, elements, limits per element), and it will show a list of possible images as soon as a part of the name is typed (min 1 character).


Custom widget images repo

Choose File Receipt_04.png

Image Name	Location	Preview
people.png	images/wdg-images/repo/2/people.png	
phone-call.png	images/wdg-images/repo/2/phone-call.png	
BookNPlan.png	images/wdg-images/repo/2/BookNPlan.png	
Receipt_04.png	images/wdg-images/repo/2/Receipt_04.png	

Image 6 – Image repo functionality