eWON Application User Guide

AUG 008 / Rev 3.0

You Select, We Connect

viewON - Symbols

How To

Content

This guide will explain you in a few steps how to use symbols, to adapt symbols or to create your own symbols.

Table of Contents

Hardware and software requirements

Hardware requirements

In order to follow this guide you'll need:

- 1 eWON with viewON capabilities (for example : eWON Flexy, eWON 4005CD or another eWON of the 4000 family)
- 1 PC with an Internet access and a recent web browser

Software requirements

eWON configuration software:

The eWON is configured through its embedded web server. All you need is a standard Web Browser software like Internet Explorer[i] or Firefox[ii].

Additionally we suggest you download the eBuddy utility on our website : http://support.ewon.biz/softwares.htm

This utility allows you to list all the eWONs on your network and to change the default IP address of an eWON to match your LAN IP address range. With eBuddy you can also easily backup/restore your configuration or upgrade the firmware of your eWON (if required).

viewON software:

To create your viewON project you need to install the viewON Editor on your PC.

To visualize the exported viewON application on the eWON you simply need an up to date web browser (IE9 and higher, Mozilla FireFox, Google Chrome,...). This makes the viewON project visualization compatible with any kind of platform (Computer, Tablet, SmartPhone) and any kind of OS (Windows, Linux, MacOS, iOS, Android).

The setup of viewON can be downloaded on the eWON web site: http://support.ewon.biz/softwares.htm

eWON firmware version

To be able to follow this guide your eWON needs a firmware version 7.1s0 or higher. A simple way to upgrade the eWON firmware is to use eBuddy, the eWON software companion.

As of version 8.0, **C**loud **A**ccelerated **L**oading is supported. This feature makes the loading of an eWON web page faster, including viewON and configuration web pages. This is useful especially when using slow communication media.

What are Symbols

A symbol is a predefined object that you can use as you please.

Instead of using several objects inside your view and animating these objects separately, you can use symbols. A symbol regroups different objects, animations and actions inside one single entity.

Let's take a valve object as an example.

To design your valve you'll probably use:

- 2 triangle objects to design the valve
- a color animation to reflect the current state of the valve (open=green, close = red)
- a send action to change the state of the valve

Inside your different views, you'll probably want to re-use this valve several times. You want to keep the same aspect and behavior of the valve. The only thing you want to change is the linked tag for the animation.

That's where the symbol becomes useful. Transforming this object into a symbol will simplify the re-use of the valve object.

To duplicate the valve, you'll then only need to insert the symbol and to attribute an eWON tag to the valve state. No need to re-adapt animations and actions and also no risk to change the look and global behavior of the valve itself.

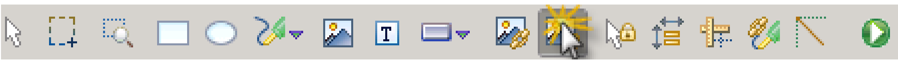
Inside this document we will first show you how to insert a symbol inside your view. ViewON already has a few predefined symbols.

Then we will show you how to create your own symbol.

Using existing Symbols

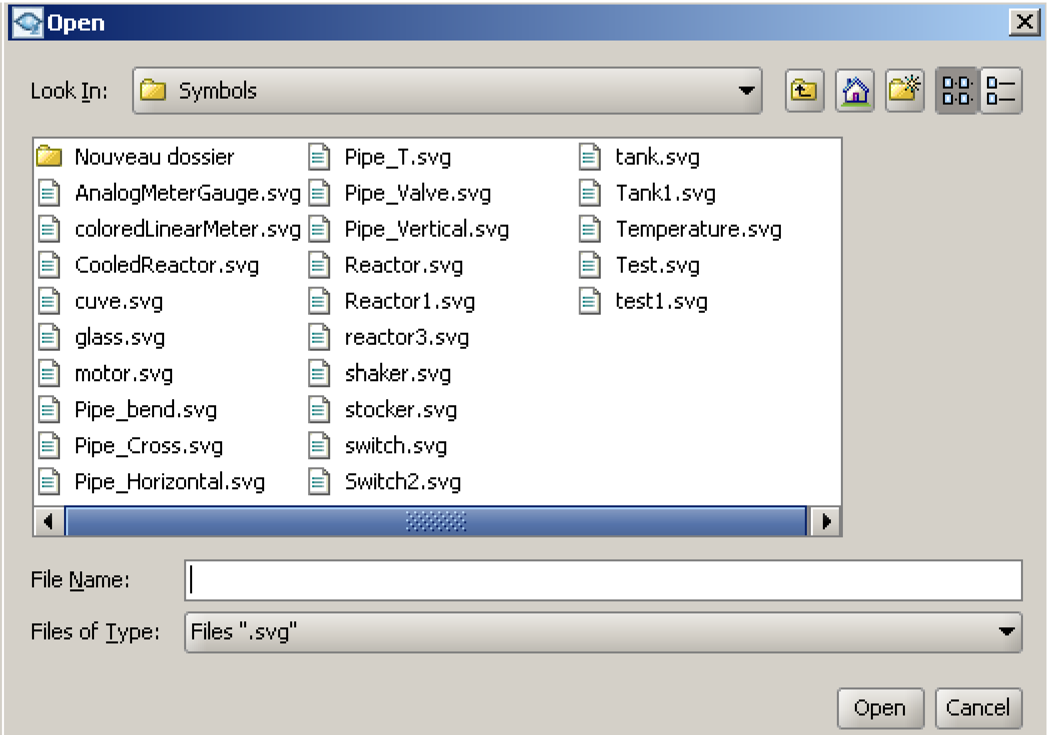
Inserting a symbol inside a view

In the toolbar click on **Insert a symbol** .



Then inside the view click on the place where you want to insert the symbol.

A window will open showing all the existing symbols. Select the symbol you want to use.



Open

Look In: Symbols

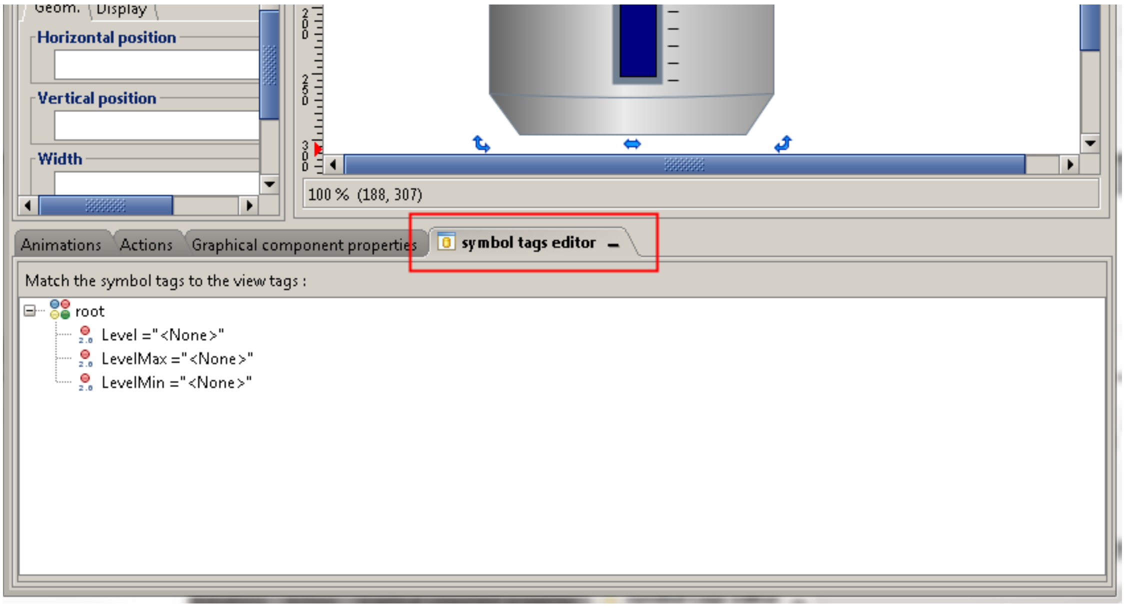
Nouveau dossier
AnalogMeterGauge.svg
coloredLinearMeter.svg
CooledReactor.svg
cuve.svg
glass.svg
motor.svg
Pipe_bend.svg
Pipe_Cross.svg
Pipe_Horizontal.svg
Pipe_T.svg
Pipe_Valve.svg
Pipe_Vertical.svg
Reactor.svg
Reactor1.svg
reactor3.svg
shaker.svg
stocker.svg
switch.svg
Switch2.svg
tank.svg
Tank1.svg
Temperature.svg
Test.svg
test1.svg

File Name:

Files of Type: Files ".svg"

Let's take the **Reactor** symbol.

To animate a symbol, don't use the Animations or Actions tabs as used for the graphical objects. Use the **Symbol Tags editor** instead.

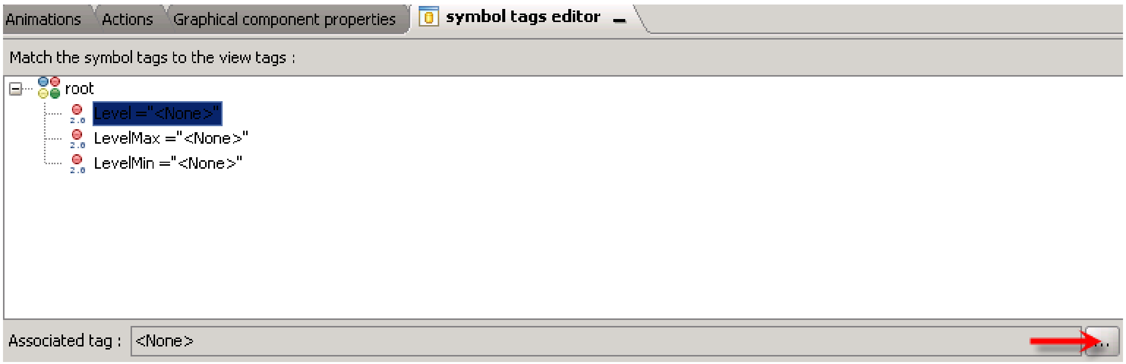


Match the symbol tags to the view tags :

- root
  - Level ="<None>"
  - LevelMax ="<None>"
  - LevelMin ="<None>"

Inside the view select the symbol and go to the **Symbol tags editor** section on the bottom of the screen. Here you'll be able to link the eWON Tags to the symbol.

For example, for the reactor symbol you'll need to add a Tag for the **Level** and a Tag for **LevelMax** and **LevelMin** values as well.

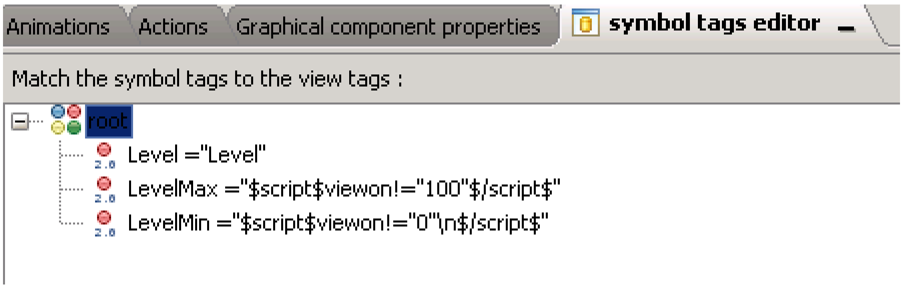


Match the symbol tags to the view tags :

- root
  - Level ="<None>"
  - LevelMax ="<None>"
  - LevelMin ="<None>"

Associated tag : <None>

Select the symbol tag you want to link to an eWON tag and click on the browse "..." button.



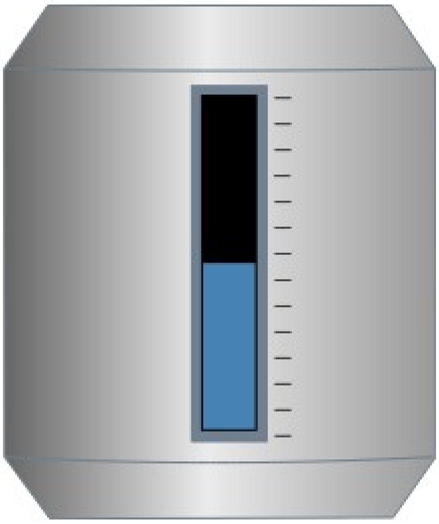
Match the symbol tags to the view tags :

```
root
  Level ="Level"
  LevelMax ="$script$viewon!="100"$/script$"
  LevelMin ="$script$viewon!="0"\n$/script$"
```

Note: Instead of linking a Tag, you could also add a Min and Max value using a Script (viewon!="0" and viewon!="100", for example).

To test the animation, click on **Test animations and actions button** as usual.

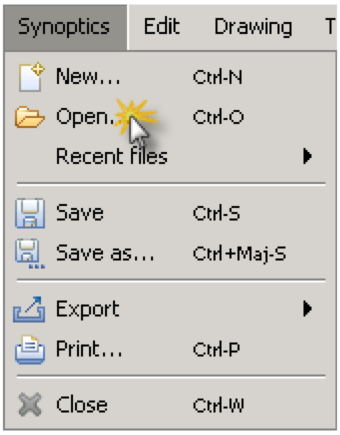
The reactor will be filled according to the Tag value. You do not need to add an animation as the symbol has an already configured one linked to a Tag.



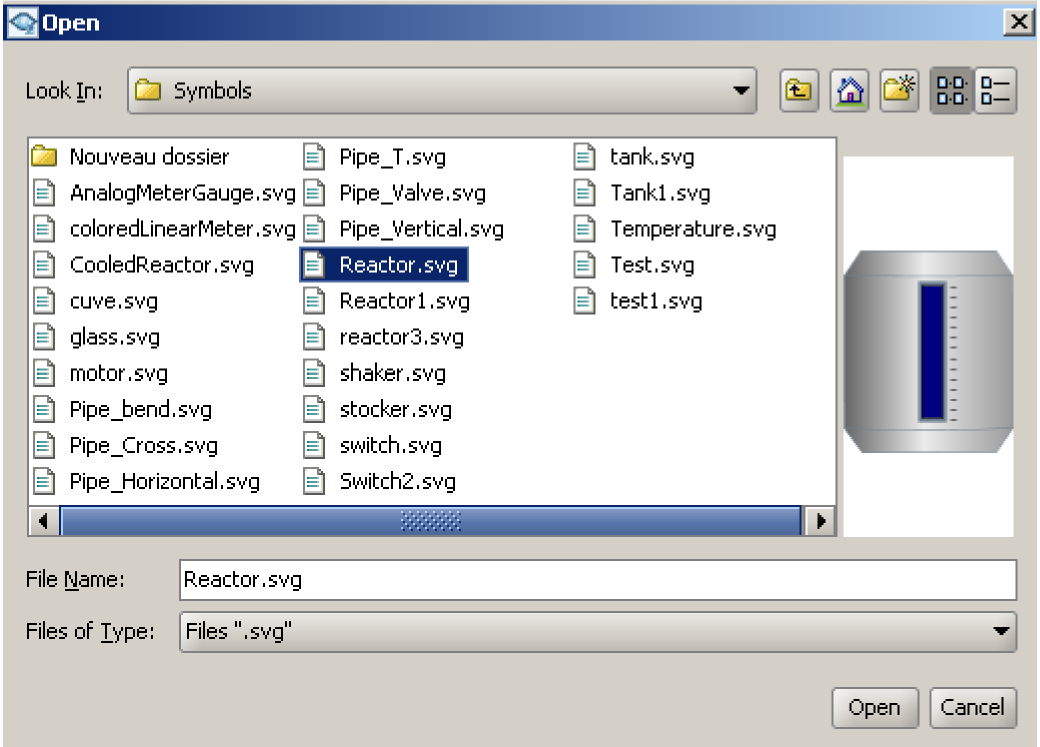
Modifying an existing symbol

It may be sometimes required to adapt the existing symbol before using it inside your view(s).
Let assume you want to personalize the symbol called "Reactor".
For the Reactor symbol the fill color is dark blue. We will show you how to change the fill color of this symbol. Also we will hard code the minimum and the maximum of the "level" range for the animation.

Click on **Synoptics** (on the menu bar) and then on **Open**.



A window that lists all the symbols pops up (open the directory **C:\Users\<*CurrentUser*>\Documents\ViewOn Projects\Symbols**).



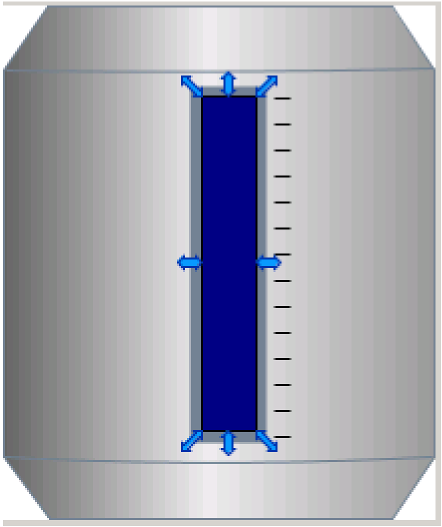
File Name: Reactor.svg

Files of Type: Files ".svg"

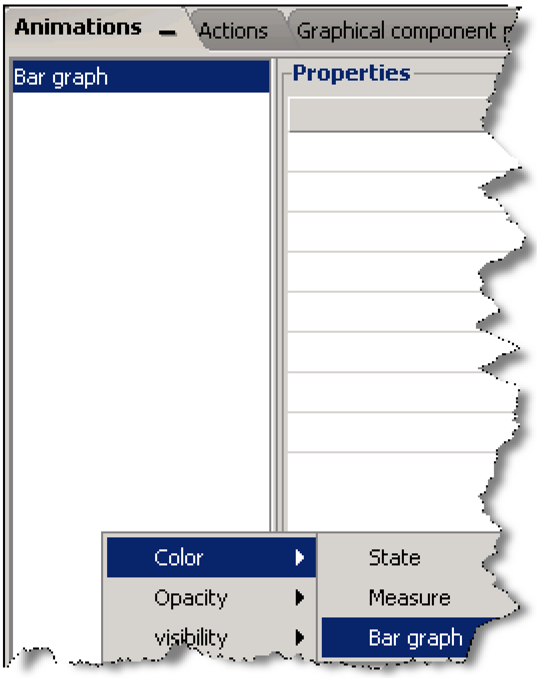
Choose the **Symbol** you want to modify (the Reactor, in our example).

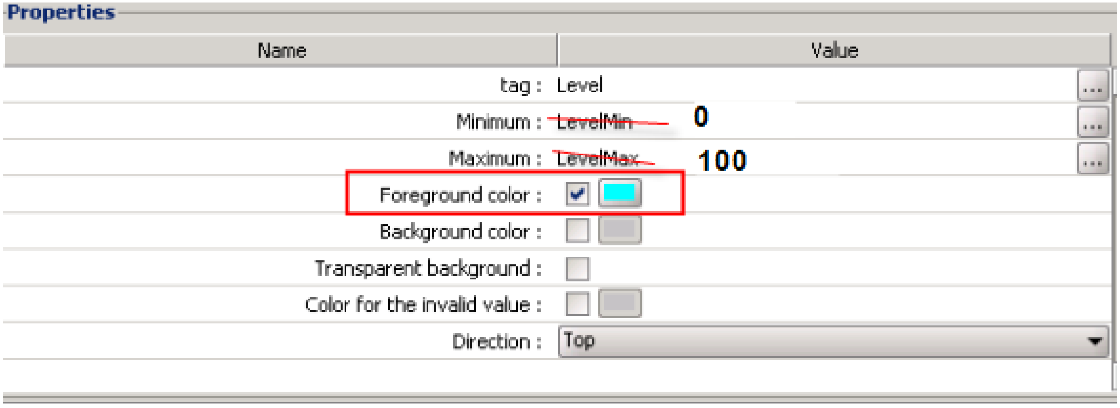
Once a **Symbol** has been opened in **Synoptics,** it behaves like a simple object, so you will be able to customize it according to your needs.

Click with the mouse on the the vertical bar of the reactor. You can now change the properties or the animation of this object.



Click on the fill and go to the **Animations** section (in the second part of the screen). There, you will see the **Bar graph** properties that you will be able to modify.





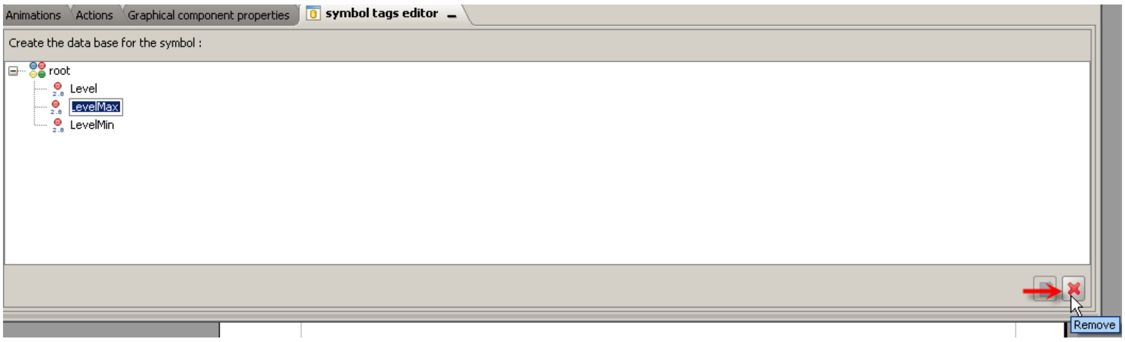
Properties

Name	Value
tag :	Level
Minimum :	~~LevelMin~~ 0
Maximum :	~~LevelMax~~ 100
Foreground color :	☑ 🟦
Background color :	□
Transparent background :	□
Color for the invalid value :	□
Direction :	Top

In the **Properties** section, set a **Minimum** and a **Maximum** (ex: 0 and 100) by deleting **LevelMin** and **LevelMax** in the **Value** section. This action will allow you to hardcode the Min and Max value.
For the foreground color,select another color.

For this Symbol we no longer need the Symbol Tags “LevelMin” and “LevelMax”. We will remove these tags from the Symbol.

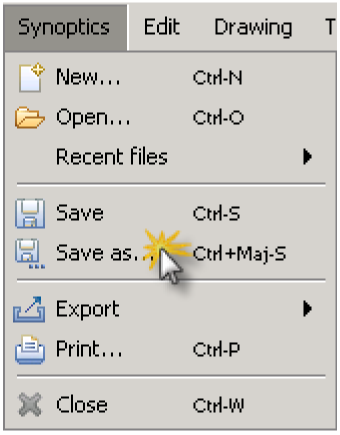
Open the Symbol tags editor section on the bottom of the page,



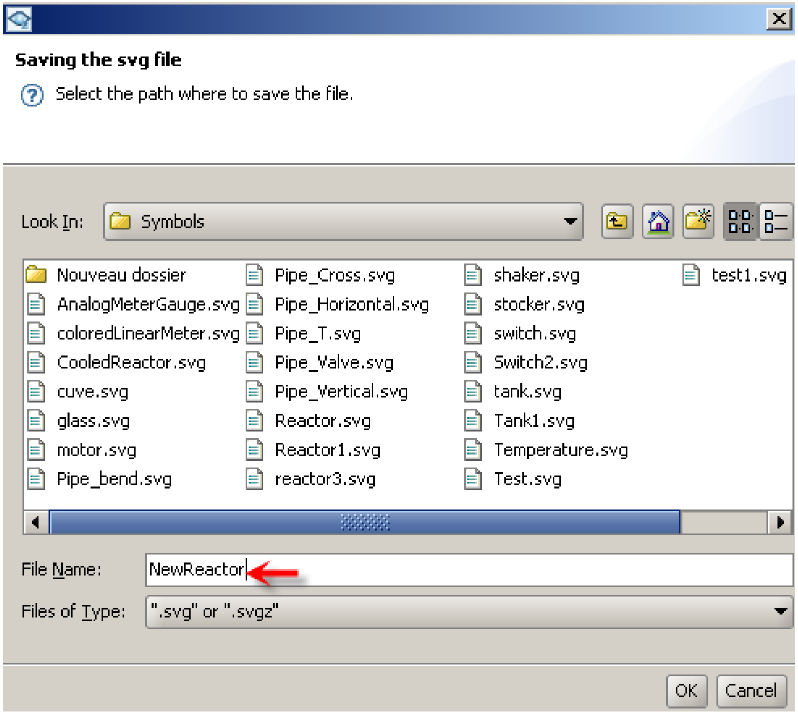
Create the data base for the symbol :

and remove the *LevelMin* and *LevelMax* symbol Tags.

To validate the modification of the symbol, click on **Synoptics** and save your object as a new symbol.



Do not forget to rename your new symbol before saving it.



Saving the svg file

Select the path where to save the file.

Look In: Symbols

Nouveau dossier, AnalogMeterGauge.svg, coloredLinearMeter.svg, CooledReactor.svg, cuve.svg, glass.svg, motor.svg, Pipe_bend.svg, Pipe_Cross.svg, Pipe_Horizontal.svg, Pipe_T.svg, Pipe_Valve.svg, Pipe_Vertical.svg, Reactor.svg, Reactor1.svg, reactor3.svg, shaker.svg, stocker.svg, switch.svg, Switch2.svg, tank.svg, Tank1.svg, Temperature.svg, Test.svg, test1.svg

File Name: NewReactor

Files of Type: ".svg" or ".svgz"

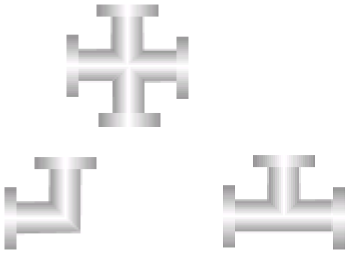
Your new symbol is now created. You can use it as many times as you want in your projects.

Note: Make sure to save your Symbols inside the directory **C:\Users\<*CurrentUser*>\Documents\ViewOn Projects\Symbols**.

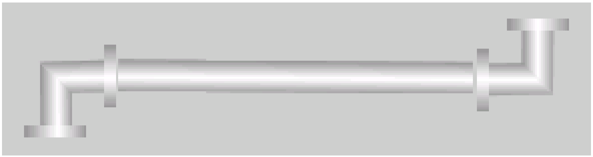
Symbols without animation

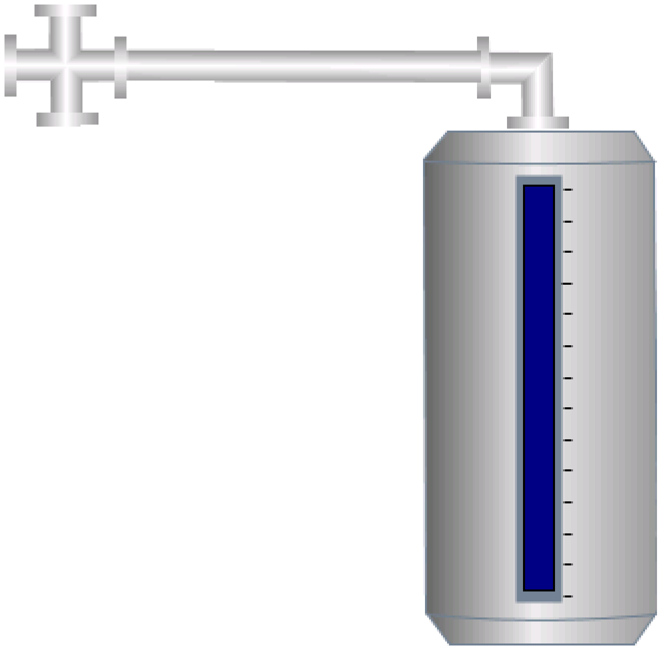
It is not required that a symbol needs to be animated.

Inside the preexisting **Symbols** list in viewON you'll find some **Symbols** that are not animated. .



Those **Symbols** do not have any predefined animation. But they will help you to design easily a more complex object, as for example:

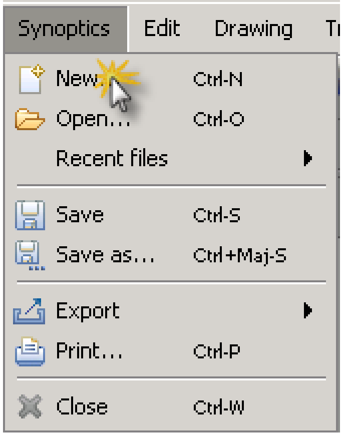


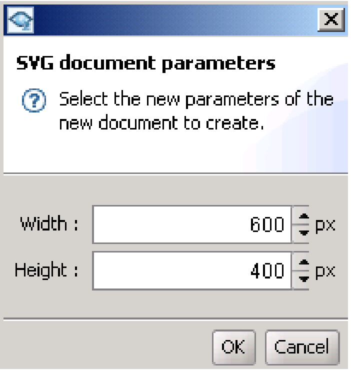


Create your own symbol

You can create your own symbol in order to reuse the object in different projects without recreating the same object over and over again.

To create a new symbol, inside the **Synoptics menu,** select **New** and set the size of the **Synoptic**.




SVG document parameters

Select the new parameters of the new document to create.

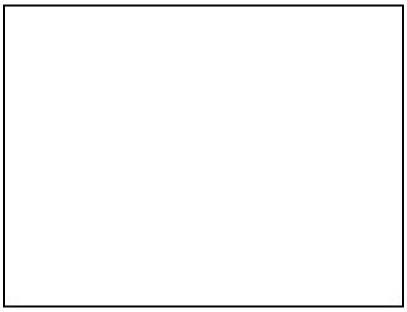
Width : 600 px

Height : 400 px

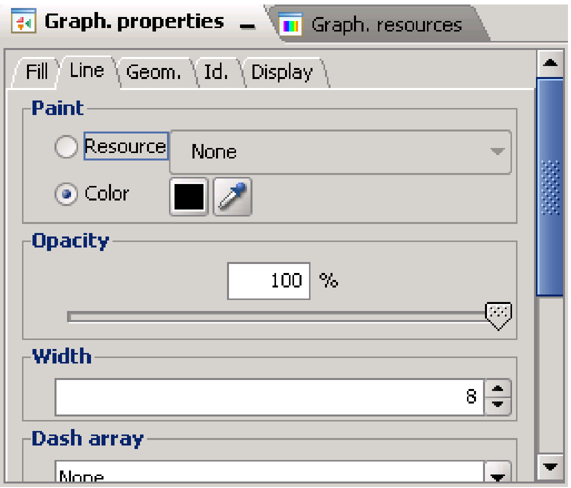
You can now draw the object you want to create as a symbol.

For example, let's create a Level display field.

Click on **Create rectangles and squares.**



Customize the object as you like. Open the **Graph. properties** section and change the different properties (the fill, the line, the geometry, etc.)



Paint

○ Resource None

● Color

Opacity

100 %

Width

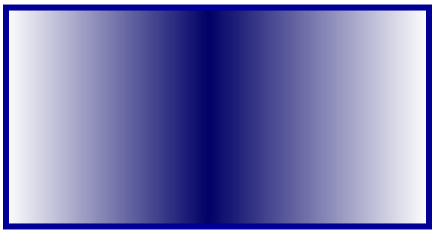
8

Dash array

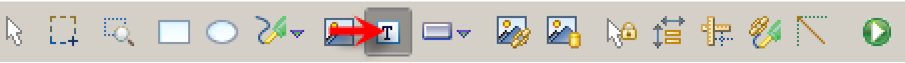
None

For further information on how to customize your object, please refer to the **AUG-006-1-EN** (Animations and Actions) document on our website: http://support.ewon.biz/viewon.html

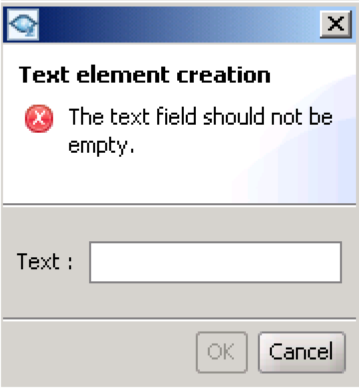
An example:



Let's now add a text on our object. To do so, click on T in the viewON toolbar and then on your object.



The following pop-up window will open.



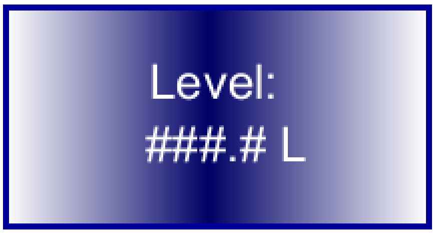
Text element creation

The text field should not be empty.

Text :

Enter your text in the blank field.

Let's assume we created following object.

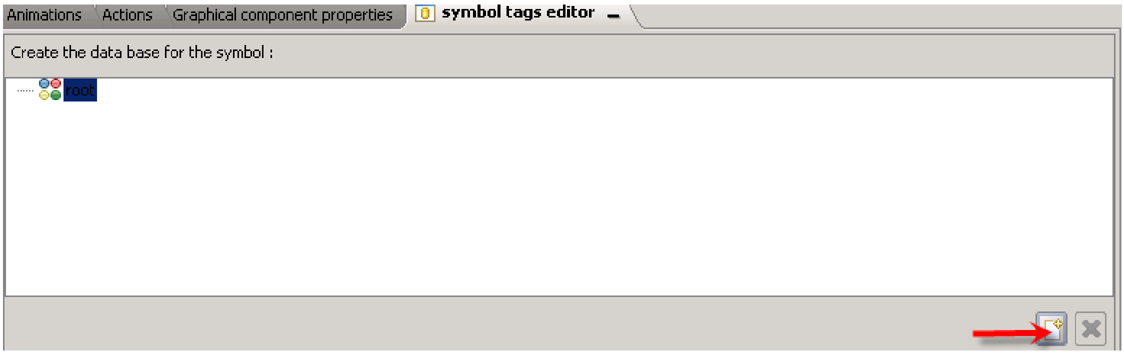


Now we want to link an animation to the “###.# L” text, to be replaced by the eWON Tag value when the symbol will be used.

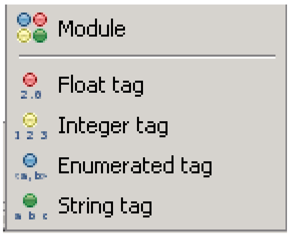
For this, we will first need to create a Symbol Tag inside our Symbol. We will then link this Symbol tag to the animation.

Later, when the symbol will be used inside the viewon project, we can then link an eWON tag to the corresponding Symbol Tag.

To create a Symbol tag, go to the **Symbol tags editor** tab (at the second part of the screen), and click on the **Insert** button.

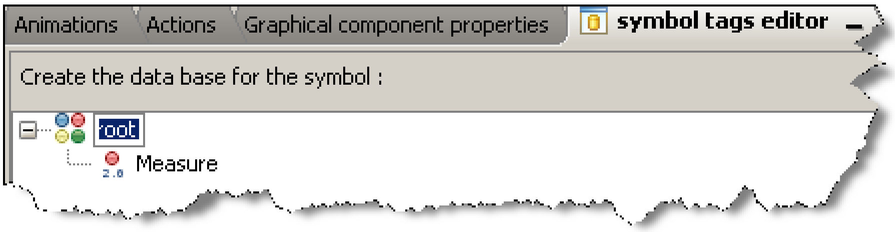

Animations
Actions
Graphical component properties
symbol tags editor
Create the data base for the symbol :
root

The following window will appear, asking for the type of tag to create.



Here you'll now need to select the type of tag corresponding to the eWON Tag you want to link to the animation later.

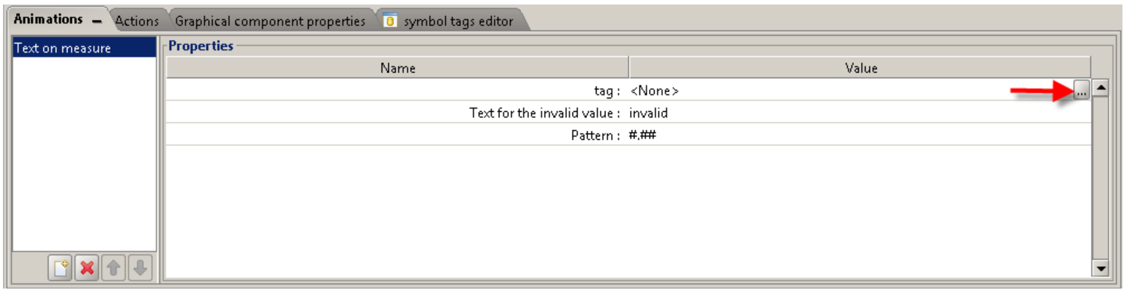
Let us select a **Float Tag** and call it “Measure” (to name it, double-click on the float tag, type the name and then press the *enter* key).



Create the data base for the symbol :

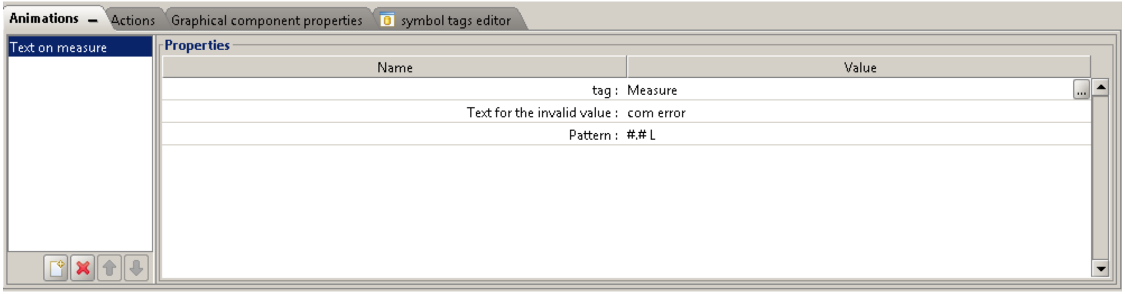
- root
  - Measure

Now we can add the animation to the text field “###.# L”. We will use the **Text/ Measure** animation.



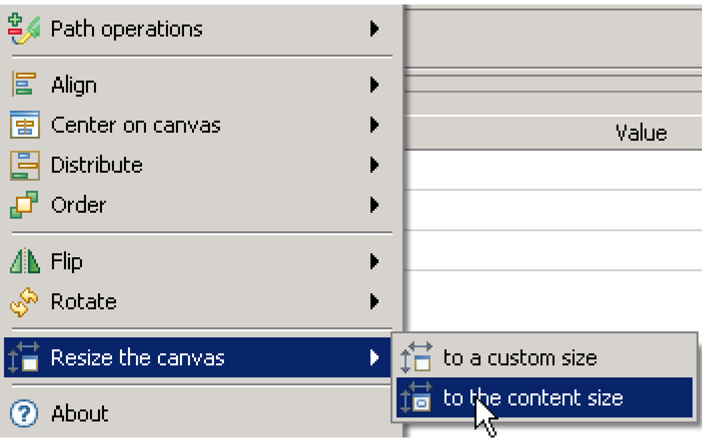
Name	Value
tag :	<None>
Text for the invalid value :	invalid
Pattern :	#.##

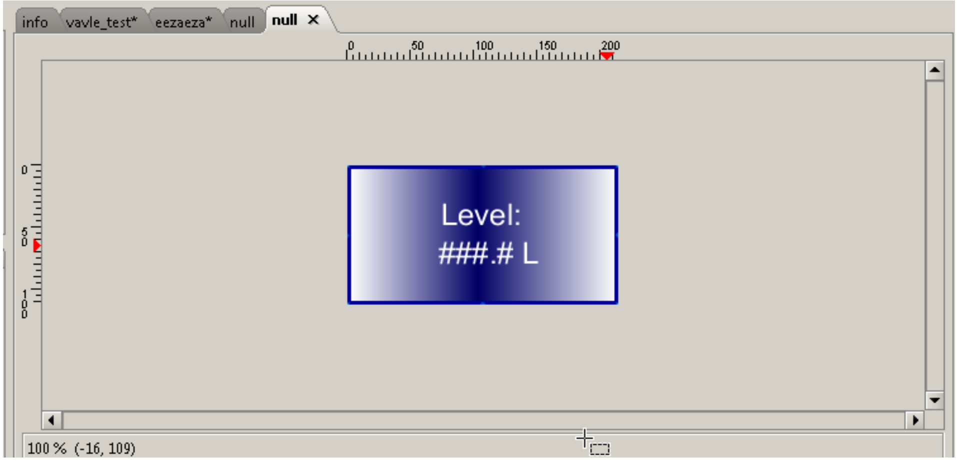
To link the animation to the symbol Tag that we just created, click on the Tag selection button.
Choose a text for the invalid value and define the display pattern (the number of decimals that you need).



Name	Value
tag :	Measure
Text for the invalid value :	com error
Pattern :	#.# L

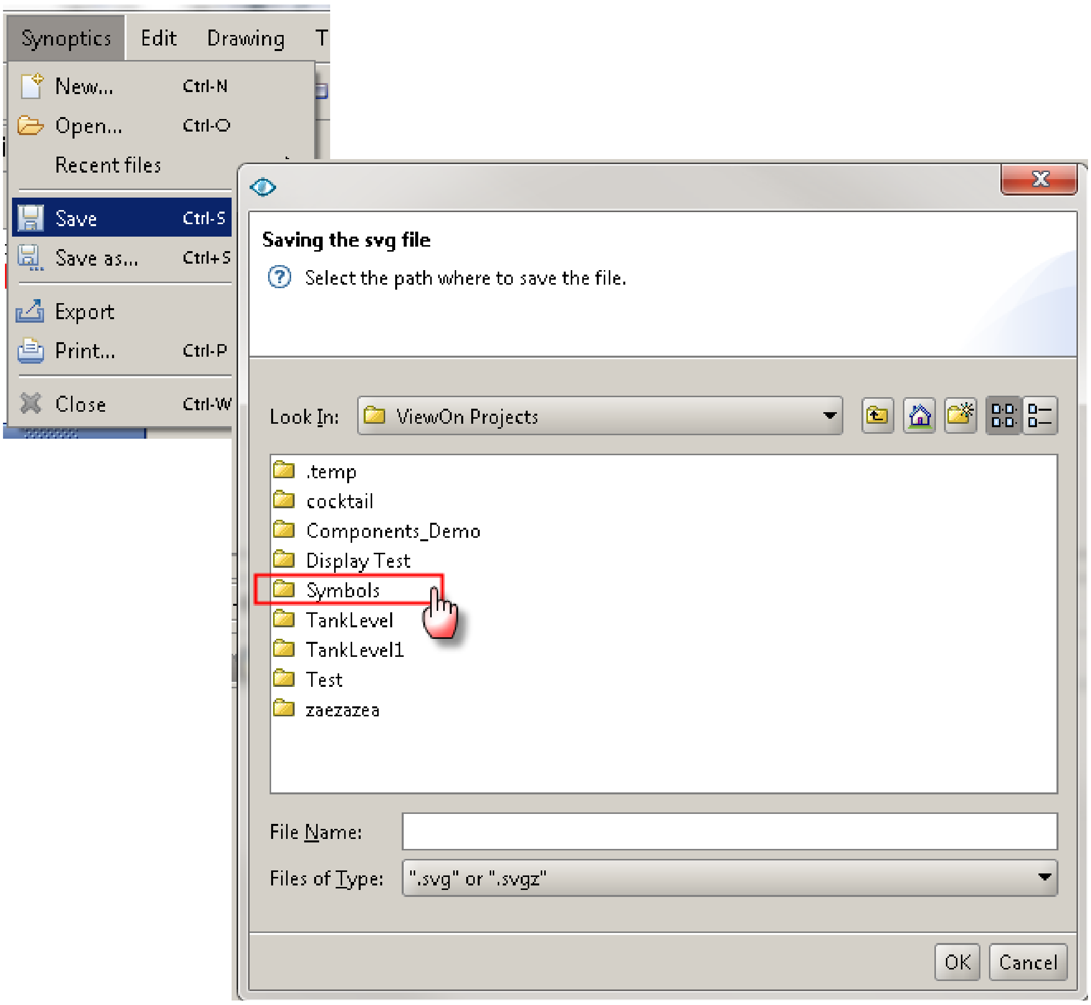
Next, we will need to re-size the canvas to the content size. Select the square, right click, and inside the “Resize the canvas” menu select the “to the content size” option.





To finish the symbol, we will need to save it.

Inside the **Synoptics menu,** click on the **Save** option



Saving the svg file

Select the path where to save the file.

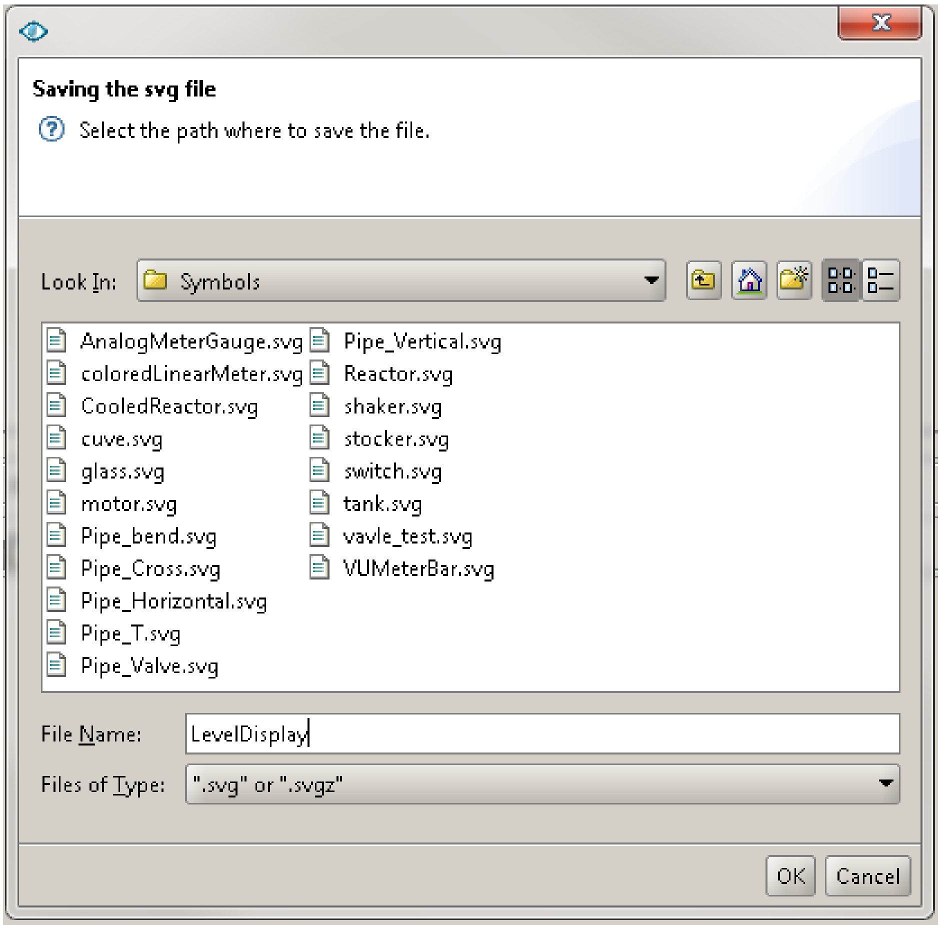
Look In: ViewOn Projects

.temp
cocktail
Components_Demo
Display Test
Symbols
TankLevel
TankLevel1
Test
zaezazea

File Name:

Files of Type: ".svg" or ".svgz"

and click on the **Symbols** directory.



Saving the svg file

Select the path where to save the file.

Look In: Symbols

AnalogMeterGauge.svg
coloredLinearMeter.svg
CooledReactor.svg
cuve.svg
glass.svg
motor.svg
Pipe_bend.svg
Pipe_Cross.svg
Pipe_Horizontal.svg
Pipe_T.svg
Pipe_Valve.svg
Pipe_Vertical.svg
Reactor.svg
shaker.svg
stocker.svg
switch.svg
tank.svg
vavle_test.svg
VUMeterBar.svg

File Name: LevelDisplay

Files of Type: ".svg" or ".svgz"

Give a name to your symbol (for example: LevelDisplay) and click on the OK button to save the synoptic.

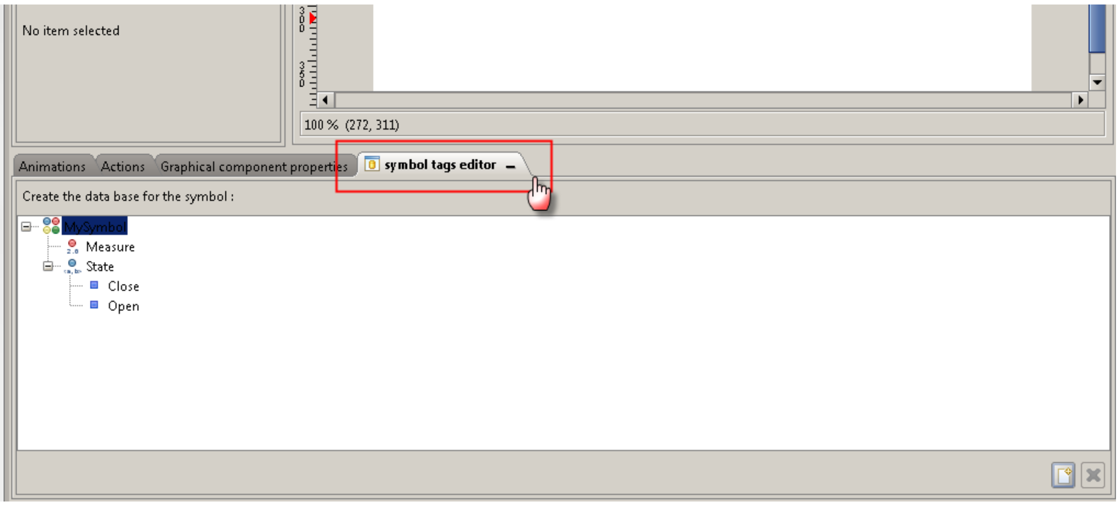
Your new symbol is now created. You can now use it inside your viewON projects simply by inserting the symbol called “LevelDisplay”.

NOTE

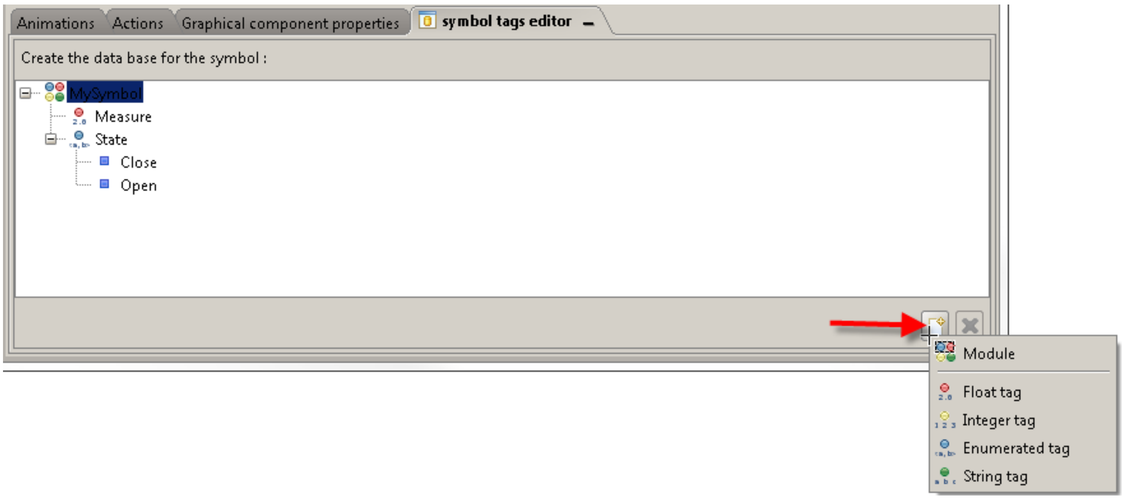
It is very important to save your Synoptic inside the directory **C:\Users\<CurrentUser>\Documents\ViewOn Projects\Symbols.** Otherwise your object will not be considered as a **Symbol**.

Symbol Tag Editor

As explained in the previous pages, to link an animation or action to your symbol, you need to create your symbol database inside the **symbol tag editor** section.



To add a symbol tag, click on the Insert button on the right lower corner.



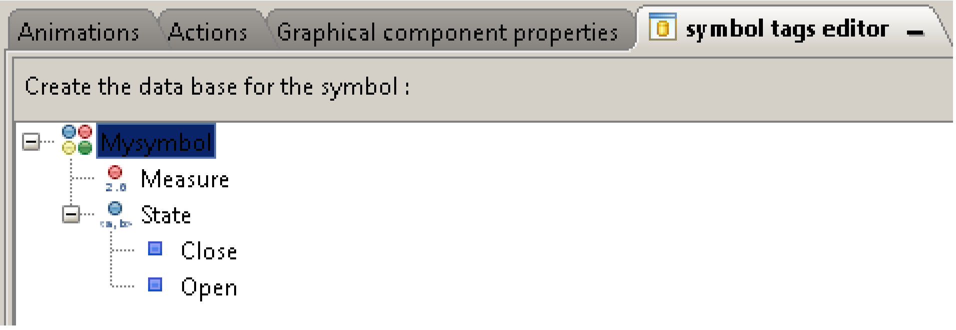
Define now the type of tag you want to insert.

The different possibilities are:

Symbol Tag type	Used for
Float Tag	eWON integer and floating point tags
~~Integer Tag~~	Currently not implemented
Enumerated Tag	eWON boolean tags
String Tag	eWON does not support string type tags. To retrieve a string out of the eWON you'll need to associate a viewON script to this symbol tag.
Module	This will add a new module inside your Symbol tag database. If your symbol is composed of several different objects and animations, the module will help to identify clearly each tag. See chapter Symbol Database module page 25

Symbol database example

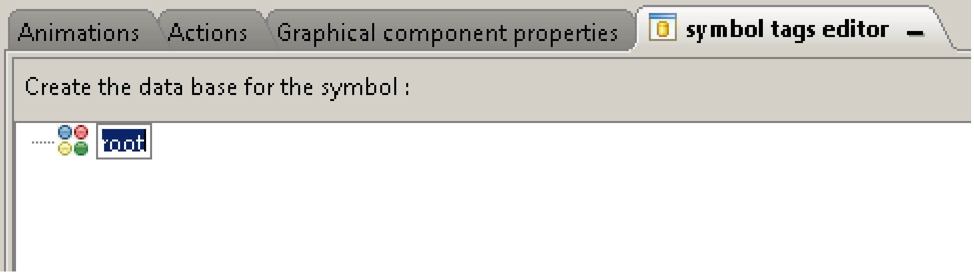
Let's for example create the following Symbol tag database structure:



Create the data base for the symbol :

- Mysymbol
  - Measure
  - State
    - Close
    - Open

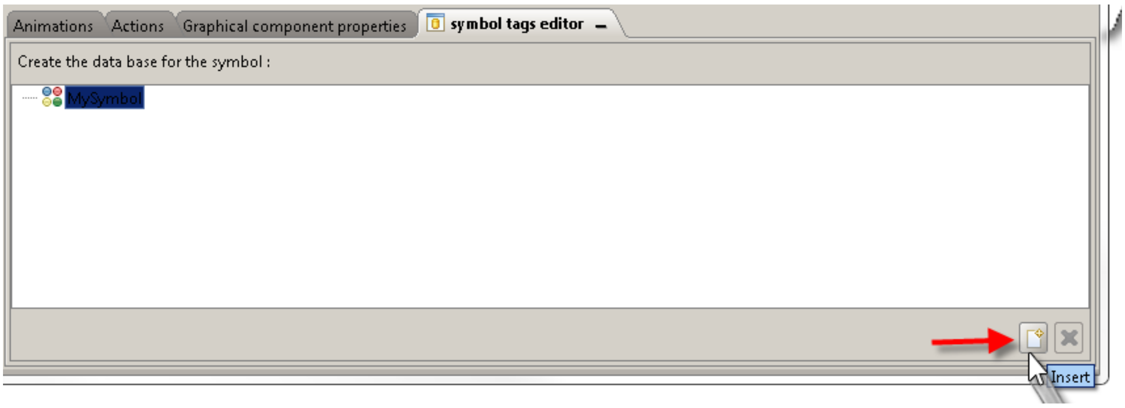
Inside the Symbol tags editor first double click the “root” text, to give a name to your Symbol database. (MySymbol for example)



Create the data base for the symbol :

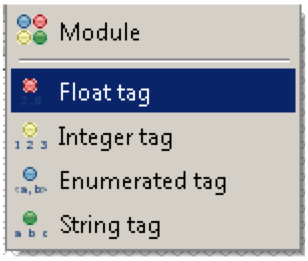
root

Then click on the **Insert** button.

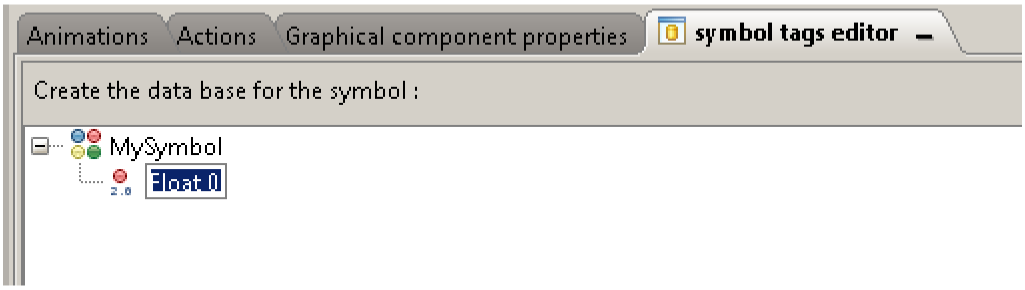


Animations
Actions
Graphical component properties
symbol tags editor
Create the data base for the symbol :
MySymbol
Insert

And select the Float tag type.

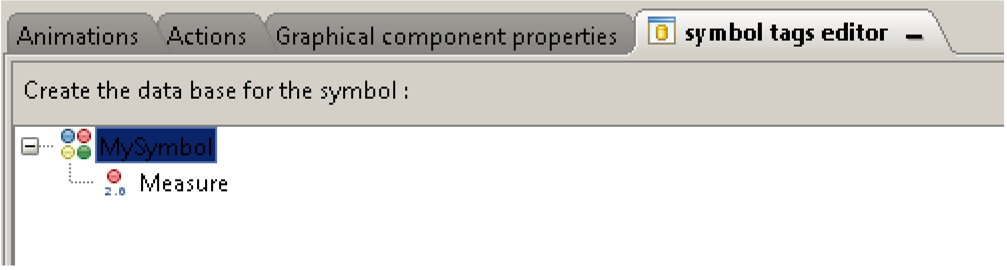


The float Tag called “Float 0” will be added to your database.



Animations
Actions
Graphical component properties
symbol tags editor
Create the data base for the symbol :
MySymbol
Float 0

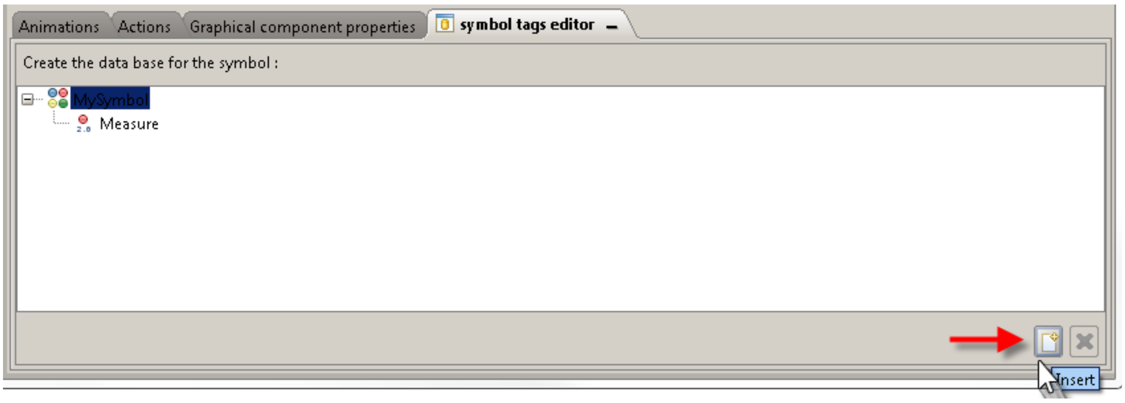
Double click the “Float 0” to change its name, as for example into “Measure”



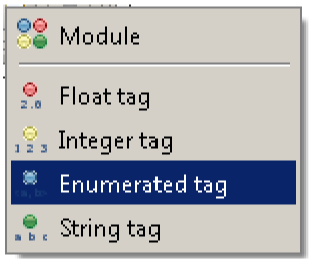
Animations
Actions
Graphical component properties
symbol tags editor
Create the data base for the symbol :
MySymbol
Measure

Now let's add a Tag into the database to link later an eWON boolean tag to the animation.

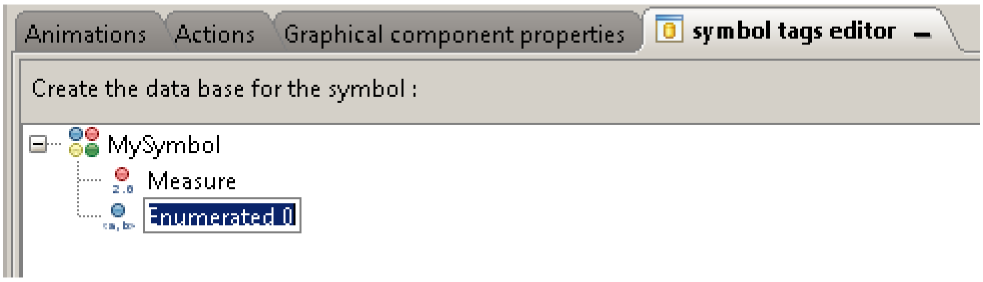
Click again on the **Insert** button


Animations
Actions
Graphical component properties
symbol tags editor
Create the data base for the symbol :
MySymbol
Measure
Insert

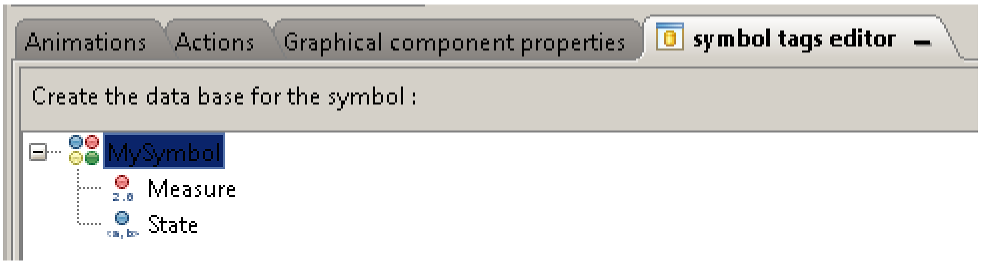
and select the **Enumerated tag** type



The Tag called "Enumerated 0" will be added to your database.

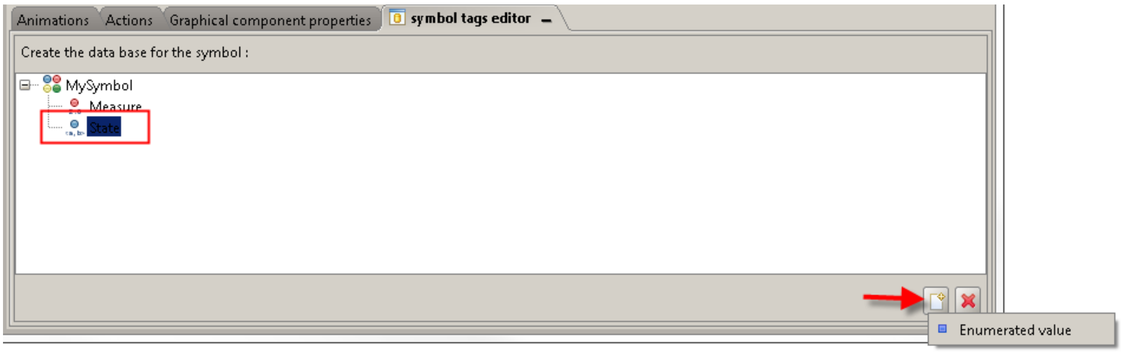

Animations
Actions
Graphical component properties
symbol tags editor
Create the data base for the symbol :
MySymbol
Measure
Enumerated 0

Double click the tag to change its name, as for example "State"


Animations
Actions
Graphical component properties
symbol tags editor
Create the data base for the symbol :
MySymbol
Measure
State

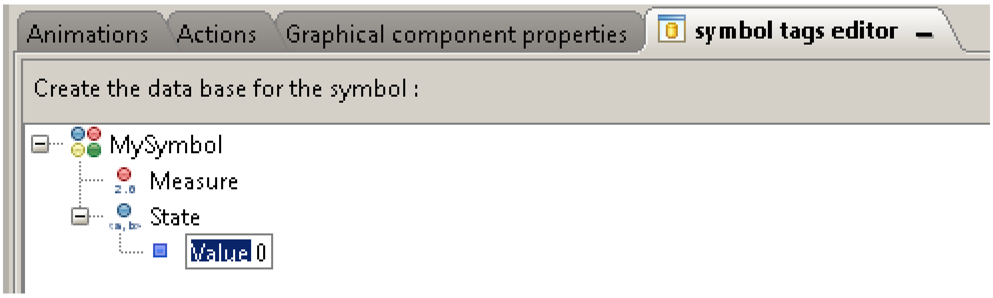
Now we need to add enumerated values for this Tag. As it is for a boolean tag we will need to add 2 states, one for value 0 and one for value 1.

Select the “State” tag and click on the Insert button.

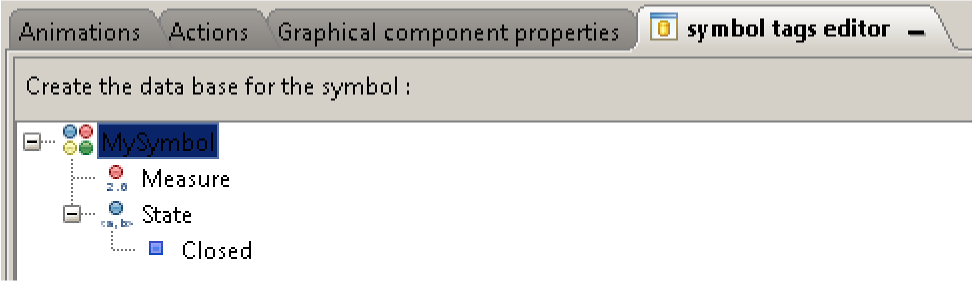


Click on the **Enumerated value** option.

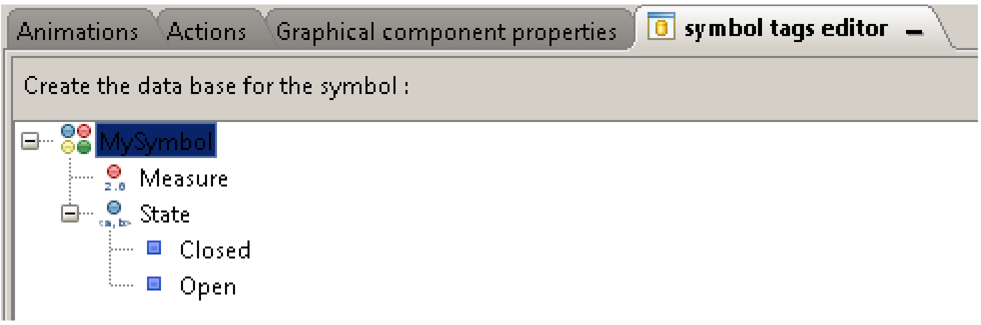
Under your State tag, you'll now see that “Value 0” has been added.


Animations
Actions
Graphical component properties
symbol tags editor
Create the data base for the symbol :
MySymbol
Measure
State
Value 0

Double click the text “Value 0” to change its name, as for example : “Closed”

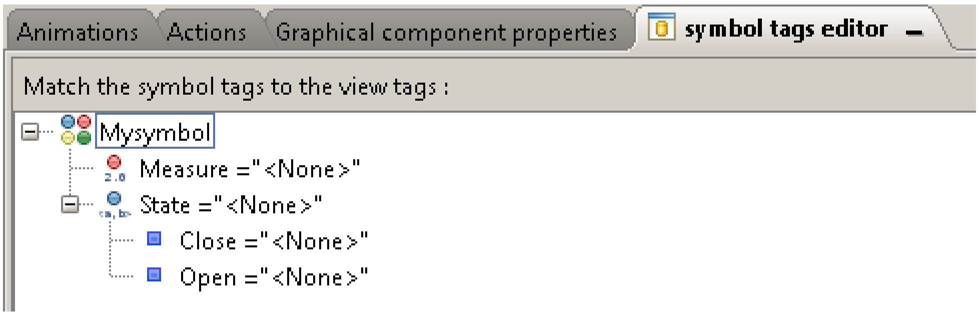

Animations
Actions
Graphical component properties
symbol tags editor
Create the data base for the symbol :
MySymbol
Measure
State
Closed

Repeat to add an “Open” state:


Animations
Actions
Graphical component properties
symbol tags editor
Create the data base for the symbol :
MySymbol
Measure
State
Closed
Open

You'll now be able to use these Symbol tags for the animation inside your symbol animations and actions.

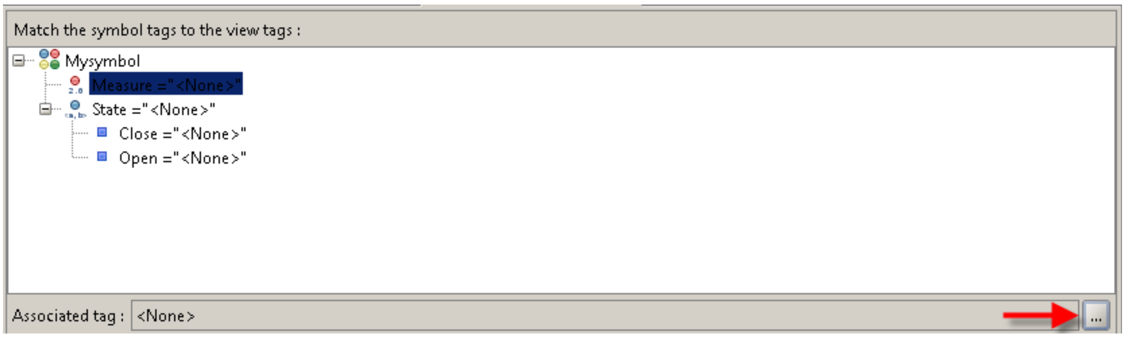
When this symbol is used inside a view, you'll need to link eWON tags to the symbol tags. For this, select the symbol and open the symbol tag editor tab.



Match the symbol tags to the view tags :

- Mysymbol
  - Measure ="<None>"
  - State ="<None>"
    - Close ="<None>"
    - Open ="<None>"

Select the Symbol tag and click on the browse button to link the eWON tag.



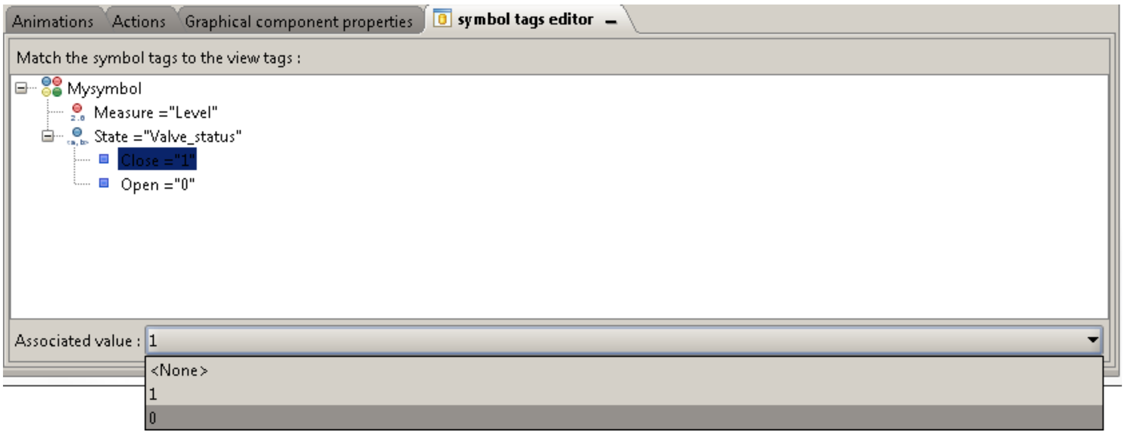
Match the symbol tags to the view tags :

- Mysymbol
  - Measure ="<None>"
  - State ="<None>"
    - Close ="<None>"
    - Open ="<None>"

Associated tag : <None>

For the Enumerated tag, you'll need to specify also the value for the stated “Close” and “Open”.

First link an eWON tag to the symbol tag called “State”,



Match the symbol tags to the view tags :

- Mysymbol
  - Measure ="Level"
  - State ="Valve_status"
    - Close ="1"
    - Open ="0"

Associated value : 1

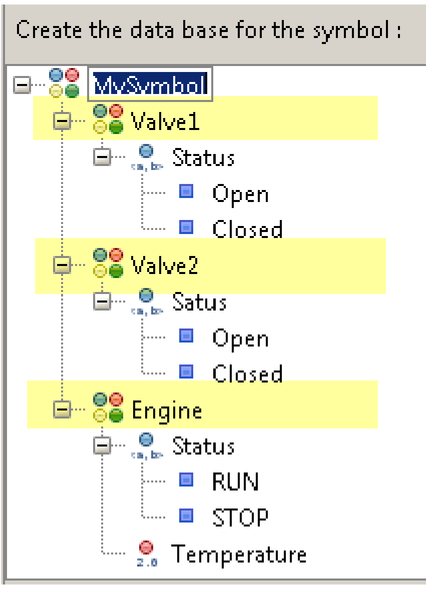
Then for each state, select the value you want to associate (0 or 1).

Symbol Database module

As explained above, the symbol tag editor supports following tag types:
- Float Tags
- Enumerated Tags
- String Tags
- Module

The ***Module*** tag type can be used to add a structure inside your symbol database. It does not represent directly a Tag.
If your symbol is composed of several animated objects, then the module will help to identify clearly to which object the tag belongs.

For example, let's assume we have a symbol regrouping 2 valves and 1 engine. All the 3 objects will have linked an Enumerated Tag called “Status”



Create the data base for the symbol :

```
MySymbol
  Valve1
    Status
      Open
      Closed
  Valve2
    Satus
      Open
      Closed
  Engine
    Status
      RUN
      STOP
    Temperature
```

Highlighted items represents the Module symbol tags. They help to identify clearly to which object the tag will be linked.

Import existing SVG graphics

Instead of designing your own object, you can also import an existing SVG image and animate it according to eWON Tags.
To be able to animate the SVG image, make sure that the image is a real SVG image and not a simple bitmap included in a SVG file (using the element “xlink:href="data:image/png;...”).

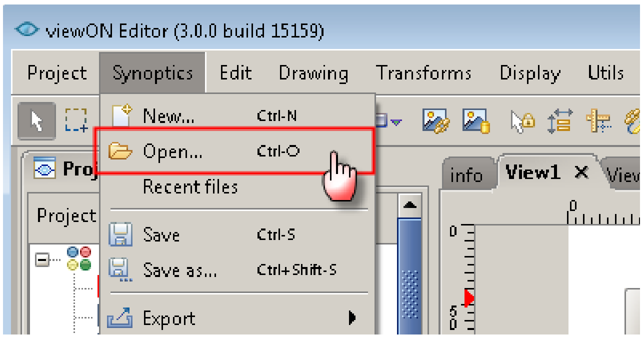
To be able to import the SVG image inside viewON, make sure that the file starts with an SVG element that includes the width and height attributes, as for example

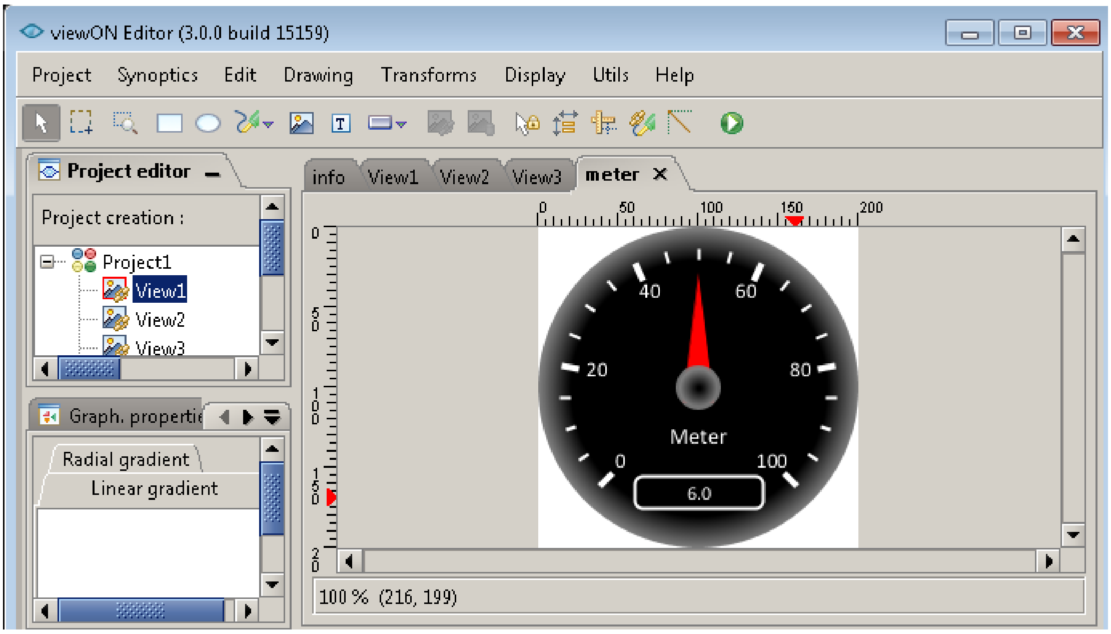
```
<svg xmlns="http://www.w3.org/2000/svg"
xmlns:xlink="http://www.w3.org/1999/xlink" viewBox="0 0 200 200"
width="200px" height="200px">
…
</svg>
```

Here is the SVG code of the example we will use :

```
<svg xmlns="http://www.w3.org/2000/svg" xmlns:xlink="http://www.w3.org/1999/xlink"
viewBox="0 0 200 200" width="200px" height="200px" preserveAspectRatio="none">
  <defs>
    <radialGradient id="circlegradient" cx="50%" cy="50%" r="50%" fx="50%" fy="50%">
      <stop offset="80%" stop-color="black" stop-opacity="1"/>
      <stop offset="100%" stop-color="#808080" stop-opacity="1"/>
    </radialGradient>
    <radialGradient id="centergradient" cx="50%" cy="50%" r="50%" fx="50%" fy="50%">
      <stop offset="10%" stop-color="black" stop-opacity="1"/>
      <stop offset="100%" stop-color="#808080" stop-opacity="1"/>
    </radialGradient>
  </defs>
  <circle id="circleout" cx="100" cy="100" r="100" fill="url(#circlegradient)"/>
  <g stroke="white" stroke-width="2">
      <line x1="100" y1="13" x2="100" y2="25" stroke-width="4" transform="rotate(27
100 100)"/>
      <line x1="100" y1="13" x2="100" y2="20" transform="rotate(40.5 100 100)"/>
      <line x1="100" y1="13" x2="100" y2="20" transform="rotate(54 100 100)"/>
      <line x1="100" y1="13" x2="100" y2="20" transform="rotate(67.5 100 100)"/>
      <line x1="100" y1="13" x2="100" y2="25" stroke-width="4" transform="rotate(81
100 100)"/>
      <line x1="100" y1="13" x2="100" y2="20" transform="rotate(94.5 100 100)"/>
      <line x1="100" y1="13" x2="100" y2="20" transform="rotate(108 100 100)"/>
      <line x1="100" y1="13" x2="100" y2="20" transform="rotate(121.5 100 100)"/>
      <line x1="100" y1="13" x2="100" y2="25" stroke-width="4" transform="rotate(135
100 100)"/>
      <line x1="100" y1="13" x2="100" y2="25" stroke-width="4" transform="rotate(225
100 100)"/>
      <line x1="100" y1="13" x2="100" y2="20" transform="rotate(238.5 100 100)"/>
      <line x1="100" y1="13" x2="100" y2="20" transform="rotate(252 100 100)"/>
      <line x1="100" y1="13" x2="100" y2="20" transform="rotate(265.5 100 100)"/>
      <line x1="100" y1="13" x2="100" y2="25" stroke-width="4" transform="rotate(279
100 100)"/>
      <line x1="100" y1="13" x2="100" y2="20" transform="rotate(292.5 100 100)"/>
      <line x1="100" y1="13" x2="100" y2="20" transform="rotate(306 100 100)"/>
      <line x1="100" y1="13" x2="100" y2="20" transform="rotate(319.5 100 100)"/>
      <line x1="100" y1="13" x2="100" y2="25" stroke-width="4" transform="rotate(333
100 100)"/>
      <line x1="100" y1="13" x2="100" y2="20" transform="rotate(346.5 100 100)"/>
      <line x1="100" y1="13" x2="100" y2="20" transform="rotate(0 100 100)"/>
      <line x1="100" y1="13" x2="100" y2="20" transform="rotate(13.5 100 100)"/>
  </g>
  <g font-family="Calibri" font-size="13px" fill="white"><text x="48"
y="150">0</text>
    <text x="30" y="93"> 20</text><text x="63" y="44"> 40</text><text x="123"
y="44"> 60</text>
    <text x="157" y="93"> 80</text><text x="136" y="150">100</text>
    <text id="comment" x="100" y="135" font-size="14px" text-
anchor="middle">Meter</text>
  </g>
  <path id="needle" d="M 90 110 L 100 28 L 110 110 z " fill="red" style="stroke:
black; stroke-width: 0.1;"/>
  <circle cx="100" cy="100" r="14" fill="gray" fill-opacity="0.9"/>
  <circle cx="100" cy="100" r="12" fill="url(centergradient)"/>
  <rect x="60" y="155" rx="5" ry="5" width="80" height="20" fill="black"
style="stroke:white;stroke-width:2;"/>
  <text x="100" y="170" style="font-family: Calibri;" text-anchor="middle"
fill="white">6.0</text>
</svg>
```

To import an existing SVG symbol in viewON, click “Synoptic → Open” and select your SVG file :

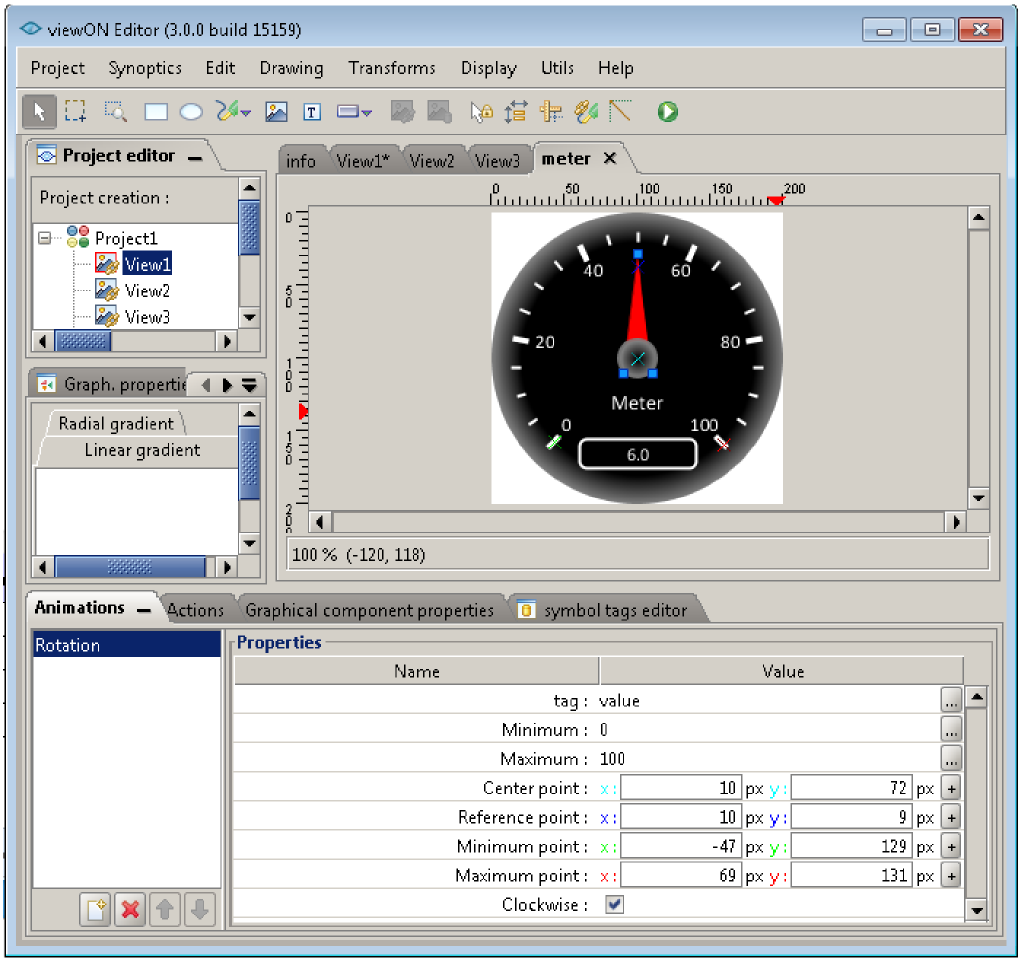




You can now edit and modify every component of the image and add animations/actions (see previous chapter).

Here, for our example, we will add

- a **Rotation** animation in order to make the needle rotate according to the value of the Tag.
- a **Text On Measure** animation to update the Text according to the Tag value.

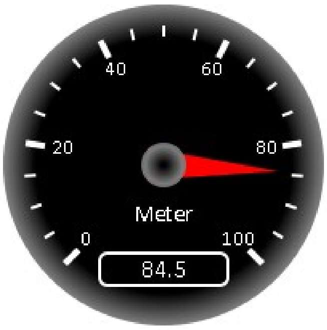


Name	Value
tag :	value
Minimum :	0
Maximum :	100
Center point :	x: 10 px y: 72 px
Reference point :	x: 10 px y: 9 px
Minimum point :	x: -47 px y: 129 px
Maximum point :	x: 69 px y: 131 px
Clockwise :	☑

To make this object become a symbol, proceed as described inside the chapter “Create your own symbol”.

End by resizing the canvas and save the Synoptic inside the directory **C:\Users\<CurrentUser>\Documents\ViewOn Projects\Symbols.**

Here is the result in the browser :



Revisions

Revision Level	Date	Description
3	09/04/14	First release for viewON version 3

i Microsoft, Internet Explorer and Windows are either registered trademarks or trademarks of Microsoft Corporation

ii Firefox is a trademark of the Mozilla Foundation

Document build number: *32*

Note concerning the warranty and the rights of ownership:

The information contained in this document is subject to modification without notice. The vendor and the authors of this manual are not liable for the errors it may contain, nor for their eventual consequences.

No liability or warranty, explicit or implicit, is made concerning quality, the accuracy and the correctness of the information contained in this document. In no case the manufacturer's responsibility could be called for direct, indirect, accidental or other damage occurring from any defect of the product or errors coming from this document.

The product names are mentioned in this manual for information purposes only. The trade marks and the product names or marks contained in this document are the property of their respective owners.