

TeslaSCADA IDE

User Manual

Version 2.17

Table of Content

About TeslaSCADA IDE.....	8
Requirements	8
Windows.....	8
Mac OS	8
Linux.....	9
Installation.....	9
Windows.....	9
Mac OS	9
Linux.....	9
Start TeslaSCADA IDE.....	10
Project.....	13
Create project.....	13
Save project	15
Open project.....	15
Edit project properties	15
Screens.....	16
Create screen.....	16
Open screen.....	16
Copy screen	16
Delete screen.....	16
Edit screen properties	16
Export screen	16
Import screen	16
Servers	17
Create server.....	17
Modbus server	17
Siemens server	17
Allen Bradley server	18
OPC UA server	18
MQTT server	18
Open server properties.....	19
Copy server	19
Delete server	19
Scripts.....	20
Create script.....	20
Open script	20

Copy script	20
Delete script	20
Edit script properties.....	20
Export script	20
Import script.....	20
Tags	21
Create tag.....	21
Modbus tag settings	21
Siemens tag settings	22
AllenBradley tag settings	22
Micrologix tag settings	22
OPC UA tag settings	22
MQTT tag settings	23
Copy tag.....	24
Delete tag	24
Edit tag properties	24
Export all tags.....	24
Import tags	24
Users	25
Create user.....	25
Open user properties.....	25
Copy user	25
Delete user	25
Databases.....	26
Create recipe	26
Create history	26
Open database properties.....	27
Copy database	27
Delete database	27
Design screen.....	28
Create graphical object	28
Resize graphical object	28
Move graphical object	28
Open graphical object properties	28
Copy graphical object	29
Cut graphical object.....	29
Paste graphical object	29
Erase graphical object.....	29

Duplicate graphical object	29
Send to back graphical object	29
Bring to front graphical object.....	29
Rotate clockwise graphical object	29
Rotate counterclockwise graphical object	30
Align graphical objects	30
Group graphical objects	30
Ungroup graphical objects.....	30
Graphical objects	31
Simple Objects library	31
Line	31
Rectangle	31
Ellipse	32
Polyline	32
Polygon	32
Sector.....	32
Text	33
Border	33
Image	33
Scale	33
3D Objects library.....	34
Sphere	34
Cylinder.....	34
Cone	34
Sector 3D	35
Polygon 3D	35
Tank	35
Border 3D.....	35
Text 3D	36
Buttons and Switches library	37
Button.....	37
Image Button.....	37
Switch and Apple switch	38
Switch 3 Pos.	38
Lights/Indicators library	39
Light	39
Indicator	39

Pipes library	40
Pipe	40
Valves library	40
Valve	40
Ball valve.....	41
Position valve.....	41
Pumps library	42
Pump.....	42
Fans library.....	42
Fan.....	42
Tanks library	43
Vertical tank	43
Conveyers library	43
Belt conveyer	43
Analog meters library	44
Analog meter.....	44
Range Indicator.....	44
Other analog meters	45
Digital meters library.....	46
Digital meters	46
Controls library	47
Slider	47
Slider vertical and horizontal.....	47
Counter and counter rectangle	48
Selector.....	48
Trends and Charts library	49
Trend	49
Pie Chart	50
Events library.....	50
Events log	50
Recipes library.....	51
Recipe table	51
Recipe selector	51
History DB library	52
History DB table	52
History DB trend.....	52
History Excel report and History Max and Min report	52

Widgets library.....	53
Web View.....	53
Analog clock.....	53
Digital clock.....	53
User-defined library	54
Create user-defined library.....	54
Rename user-defined library	54
Delete user-defined library	54
Export user-defined library	54
Import user-defined library	54
Properties	55
Flash.....	55
Rotation.....	56
Motion.....	56
Visibility	57
Line color.....	57
Fill color.....	58
Filling.....	58
Text color	59
Control (for buttons)	59
Text input	61
Output.....	61
Indicator.....	61
Rotation Indicator	62
Image	62
Color.....	62
Control (slider).....	63
Control (counter)	63
Value (for meters).....	63
Value (for Range indicator and Gauges).....	64
Switch control.....	65
Switch control (3 pos.).....	65
Selector	66
Design script	67
Create script object	67
Connect script objects	67
Bind script object to the tag	67
Enter value to the value script object	67

Duplicate script object	67
Erase script object.....	67
Erase connection line	67
Script objects	68
Input/Output library.....	68
Logical library	68
Bitmap operations library.....	68
Arithmetic library.....	68
Compare library.....	69
Select library.....	69
Arrays library	69
Triggers/Counters library	69
Trigonometric library.....	70
Hex operations library	70
Call screen library.....	70
Date and time library	70
Servers library	70
Recipes library.....	70
Simulation.....	71
Load on device	73
Network method	73
Manual method.....	74
Import for iOS	75
Network method	75
Manual method.....	76
Use Telegram Bot	77

About TeslaSCADA IDE

TeslaSCADA IDE is an integrated development environment used for configuring, developing and managing HMI/SCADA applications. In this manual you will find everything you need to create a full-featured SCADA (Supervisory Control and Data Acquisition) project visualization. With this tool you can create and manage TeslaSCADA projects, configure connections with devices, enter tags, alarms, and trends.

A simple to use interface allows for easy manipulation of the project's configuration and data processing. The project data are stored in a single file (based on xml) for easy backup and restoration.

TeslaSCADA IDE has an integrated GUI (Graphical User Interface) visualization editor for easy creation of professionally looking graphics.

Requirements

TeslaSCADA IDE requires Windows, Mac OS or Linux operating systems.

Windows

Processors: Intel Pentium 4, Intel Centrino, Intel Xeon, or Intel Core Duo (or compatible) 1.8 GHz minimum.

Operating systems: Windows 8 (Modern UI (i.e. Metro Mode) is not supported), Windows 7, Windows Vista, Windows XP (not recommended but supported).

Memory: 512MB of RAM (1 GB recommended).

Disc Space: 256MB of free disc space.

Mac OS

Processors: Dual-Core Intel, PowerPC G5

Operating systems: 10.7.3 or greater

Memory: 512MB of RAM (1 GB recommended).

Disc Space: 256MB of free disc space.

Linux

Processors: Intel Pentium 4, Intel Centrino, Intel Xeon, or Intel Core Duo (or compatible) 1.8 GHz minimum.

Operating systems: Ubuntu 10.4 + gtk2 2.18+

Memory: 512MB of RAM (1 GB recommended).

Disc Space: 256MB of free disc space.

Media: You must install the following in order to support AAC audio, MP3 audio, H.264 video, and HTTP Live Streaming:

libavcodec52 and libavformat52 on Ubuntu Linux 10.04, 10.10, 11.04 or equivalent.

libavcodec53 and libavformat53 on Ubuntu Linux 11.10, 12.04 or equivalent.

Installation

Windows

To install TeslaSCADA IDE download EXE package for your operating system. Run installation file and go through installation procedure.

Mac OS

To install TeslaSCADA IDE download DMG package for your operating system. DMG package provides a simple drag-and-drop installation experience.

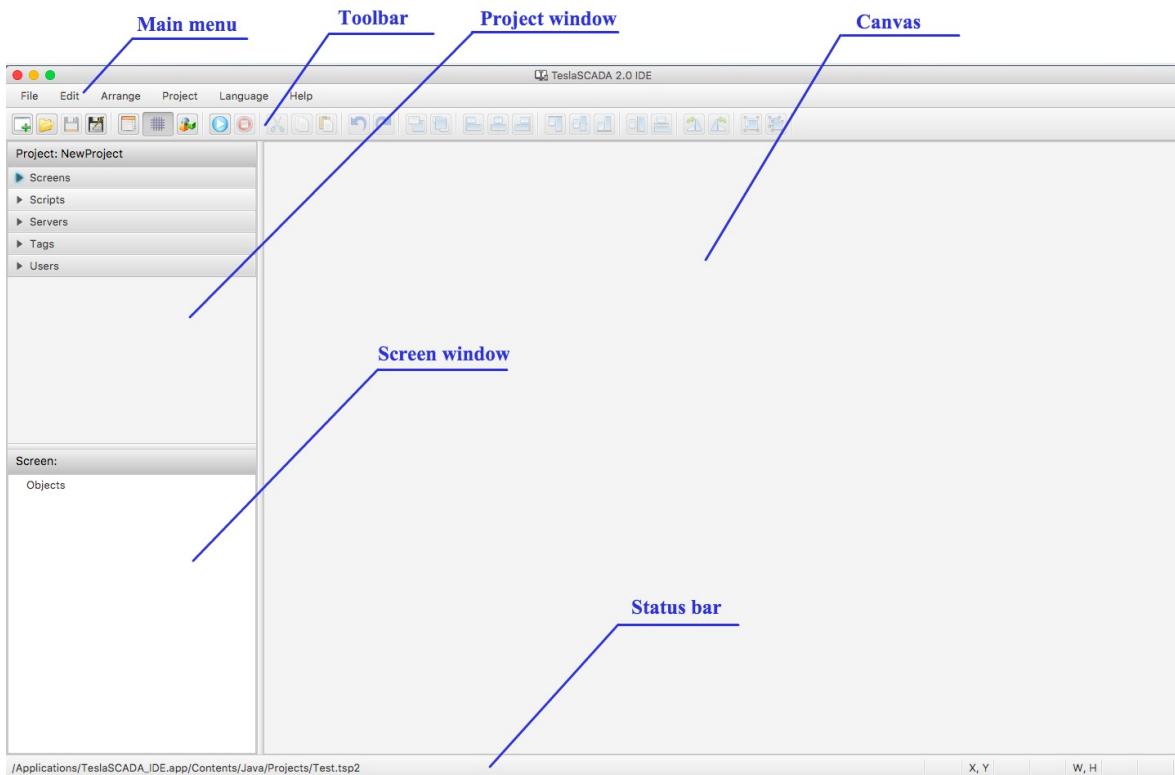


Linux

To install TeslaSCADA IDE download RPM package for your operating system. By default RPM package will install the application to /opt, add a shortcut to the application menu. RPM package does not have any UI for installation (normal behavior for Linux)

Start TeslaSCADA IDE

After opening the application you will see the start screen. Look at the picture below to briefly get to know the TeslaSCADA IDE interface:



Main menu

File - manipulation with project files.

Edit - manipulation with objects (cut, copy, paste and etc.).

Arrange - arrange manipulation with objects (align, rotate and etc.).

Project - possibility to create new objects of the project, change its properties and run/stop simulation.

Language - possibility to change language of the interface.

Help - opens the help menu

Toolbar

The toolbar consists of the following functions:



New project – creates a new project.



Open project – opens an existing project.

-  **Save** – saves your project.
-  **Save as** – saves your project with a new name.
-  **Properties** – properties of your project.
-  **Snap to Grid** – ON/OFF snap to grid.
-  **New object** – creates a new graphical object.
-  **Run simulation** – start simulation of your project.
-  **Stop simulation** – stop simulation of your project.
-  **Cut** – cut selected object(s).
-  **Copy** – copy selected object(s).
-  **Paste** – paste selected object(s).
-  **Undo** – undo the last operation.
-  **Redo** – redo the last operation.
-  **Send to Back** – send to back selected object.
-  **Bring to Front** – bring to front selected object.
-  **Align Left** – align to the left side the selected objects.
-  **Align Center** – align the vertical center of the selected objects.
-  **Align Right** – align to the right side the selected objects.
-  **Align Top** – align on top of the selected objects.
-  **Align Middle** – align the horizontal center of the selected objects.
-  **Align Bottom** – align to the bottom of the selected objects.
-  **Space Horizontal** – align the horizontal spacing between the selected objects.
-  **Space Vertical** – align the vertical spacing between the selected objects.
-  **Rotate Clockwise** – rotate clockwise selected object(s).
-  **Rotate CounterClockwise** – rotate counterclockwise selected object(s).
-  **Group Objects** – group selected objects.
-  **Ungroup Objects** – ungroup selected objects.
-  **Lock Object** – lock object to the position.
-  **Unlock Object** – unlock object from the position.

Project window

Project window contains all the information about the project and consists:

Screens - contains all screens of the project.

Scripts - contains all scripts of the project.

Servers - contains all servers of the project.

Tags - contains all tags of the project.

Users - contains all users of the project.

Screen window

Screen window contains all objects of the current screen.

Status bar

Status bar contains information about path of the current project, information about selected object (x,y coordinates and dimension) and information about run or not simulation mode.

Canvas

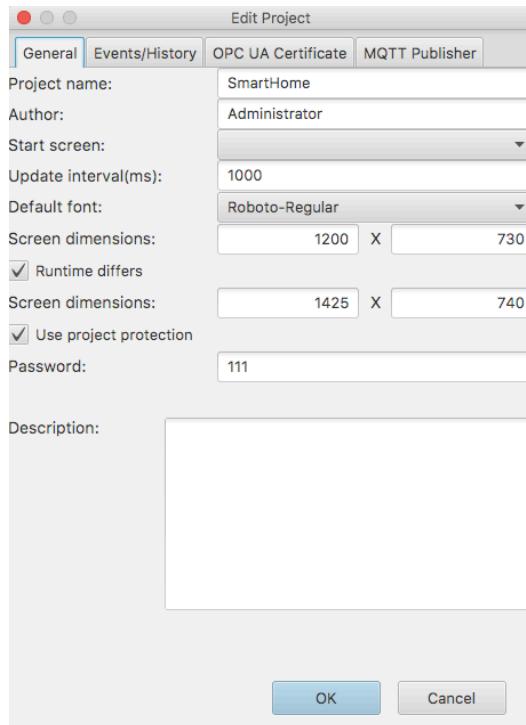
Place for the design screen or script.

Project

Create project

To create a new project TeslaSCADA IDE must be started.

1. Click on the **New** icon in the toolbar or use the command **New** from the main menu **File**. You'll see the following window:



2. On the *General* tab:

- 2.1. In the **Project name** enter the name of the project.
- 2.2. In the **Author** write the author of the project if you want.
- 2.3. When you create a new project the **Start screen** combobox is empty. You can choose the start screen after creating screens of the project.
- 2.4. Choose **Default font**. System font let you use Chinese, Arabian and etc language symbols.
- 2.5. In the **Update interval(ms)** enter update interval of the project. It's an interval of updating objects of the current screen.
- 2.6. Enter default dimensions of your design screen in the **Screen dimensions** fields.
- 2.7. If the screen dimensions of your target device differs check **Runtime differs** and enter its **Screen dimensions**.
- 2.8. Optionally, specify a meaningful **Description** yet.
- 2.9. If you want to protect your project from opening by non-authorised person check **Use project protection**.
- 2.10. Enter **Password** for protecting your project.

3. On the *Events/History* tab:

3.1. Select the time period during which data will be stored in databases in the **Storage DB period** combobox.

3.2. Enter databases names in the **Events DB name** and **History DB name**. If you choose the simple names like *events* or *history* application will create SQLite database in the application directory. If you choose names beginning with **jdbc:mysql:** like **jdbc:mysql://192.168.0.104:3306/test** the application will connect to MySQL database and create events or history table. **Don't create big MySQL databases for connecting from Android devices (MySQL databases need a wide network bandwidth for sending and receiving data).**

3.3. Enter **Username** and **Password** if you use MySQL database.

3.4. Enter **Notifications(Priority)**. Events with a priority lower than this will be notified about it by using the pop-up window and sound.

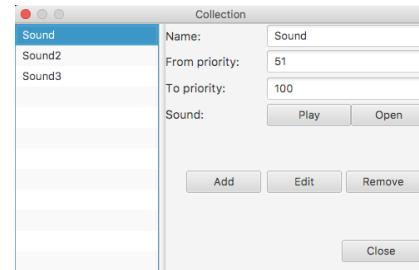
3.5. Click Sounds **Collection** to set up sounds of events notifications depending on priority. After clicking you'll see the window. Where:

3.5.1. **Name** of sound.

3.5.2. **From priority** and **To priority** - priority interval within which sound will play.

3.5.3. Button to **Play** sound and button to **Open** sound (wav) file.

3.5.4. Buttons to **Add**, **Edit** and **Remove** sounds.



3.6. If you check **Show servers events** you'll get information about disconnection, lost or restore servers.

3.7. If you want to use Telegram bot in your project check **Use Telegram Bot**.

For more information about using telegram bot in your project see the chapter below.

3.8. Enter **Bot's name**. You get Telegram Bot's name from BotFather when you creating your bot.

3.9. Enter **Bot's token**. You get Telegram Bot's token from BotFather when you creating your bot.

3.10. Check **Use E-mail client** if you want to use E-mail notifications about Alarms. All event messages that have priority < **Notifications(Priority)** will be sent by E-mail.

3.11. Enter E-mail **Host**.

3.12. Enter E-mail **Port**.

3.13. Choose **Type** of the connection - TLS or SSL.

3.14. Enter **From** which **E-mail address** the mail will be sent.

3.15. Check **Authentication** if you use Username and Password.

3.16. Enter **Username** of the E-mail.

3.17. Enter **Password** of the E-mail.

3.18. Enter **To** which **E-mail addresses** the mail will be sent. Use commas to separate addresses.

4. If you use OPC UA server in your project on the *OPC UA certificate* tab enter **Name** of used/created certificate and **Period(days)** of validation if you create certificate. The certificate stored in the `{app}/private` directory.

5. If you want to use **MQTT Publisher** check **Enable MQTT Publisher**.

5.1. Enter **Broker URL** of the MQTT server.

5.2. Enter **Username** and **Password** of the MQTT server.

5.3. Choose **QoS** of MQTT messages.

5.4. Check **Enable TLS/SSL** if you want to use server certificate for encryption messages.

5.5. Enter **Certificate filename**. File should be placed in `/private/` folder in the directory where TeslaSCADA2 Runtime execution file.

5.6. Check **Enable Client Certificate** if you want to use client certificate for encryption messages.

5.7. Enter **Client certificate** filename. File also should be placed in `/private/` folder.

5.8. Enter **Client Private key** filename. File also should be placed in `/private/` folder.

5.9. Enter **Private key password**.

5.10. Check **PEM formatted** if your certificate and key files are PEM formatted.

Publisher's topics are consists of the «name of the project +/Tags/+tagname» for tags and «name of the project+/Events/+tagname» for events.

Save project

To save project:

1. Click on the **Save** icon in the toolbar or select the menu item *File* and *Save*. The first time you save a new project, you will be asked for a location.

2. Now select the location and click the button *Save* (TeslaSCADA project extension .tsp2).

Open project

To open project:

1. Click on the **Open** icon in the toolbar or select the menu item *File* and *Open*.

2. Now select the project and click *Open* (TeslaSCADA project extension .tsp2).

Edit project properties

To edit project properties:

1. Click on the **Properties** icon in the toolbar or select the menu item *Project* and *Properties*.

Screens

Create screen

To create a new screen select the menu item **Project** and **New Screen** or choose **Screens** on the **Project Window**, click right button on it and choose **New Screen** item.

You'll see the following window:

1. In the **Name** enter the name of the screen.
2. Optionally, specify a meaningful **Comment**.
3. Choose **Background** color.
4. Select **Screen type**: *General* or *Popup*.
5. Add **Collection of Scripts** for this screen if you want.
6. Enter **Screen dimension**.
7. Check **Use password** if you want to use screen security.
8. Enter **Password** of the screen.

Open screen

To open screen:

1. Right click on the screen you want to open and choose *Open* item.
or
2. Double click on the screen you want to open.

Copy screen

To copy screen:

1. Right click on the screen you want to copy and choose *Copy* item.

Delete screen

To delete screen:

1. Right click on the screen you want to delete and choose *Delete* item.

Edit screen properties

To edit screen properties:

1. Right click on the screen you want to edit and choose *Screen properties* item.

Export screen

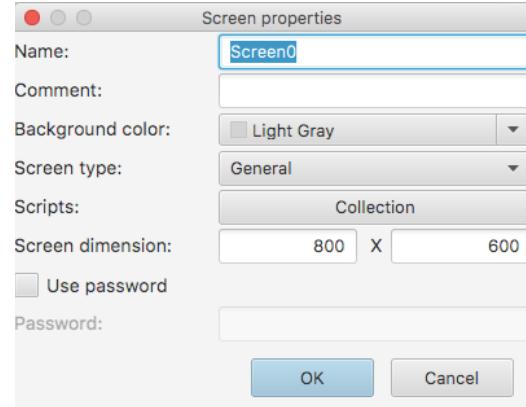
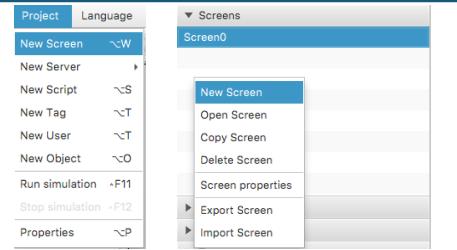
To export screen:

1. Right click on the screen you want to export and choose *Export screen* item.
2. Now select the location and click the button *Save* (TeslaSCADA screen extension .tsp2screen).

Import screen

To import screen:

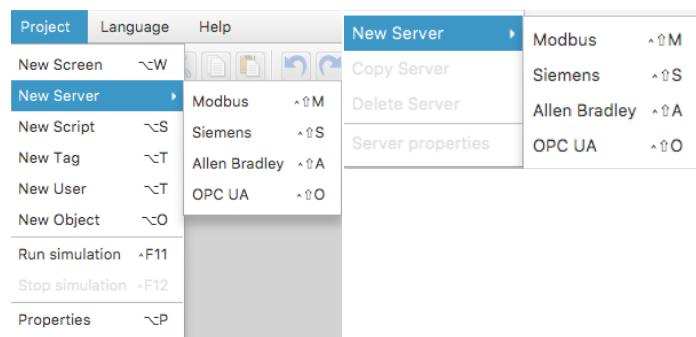
1. Right click on the screen window and choose *Import screen* item.
2. Now select the screen file and click *Open* (TeslaSCADA screen extension .tsp2screen).



Servers

Create server

To create a new server select the menu item **Project** and **New Server** or choose **Servers** on the **Project Window**, click right button on it and choose **New Server** item. Choose server you want to add to your project.



Modbus server

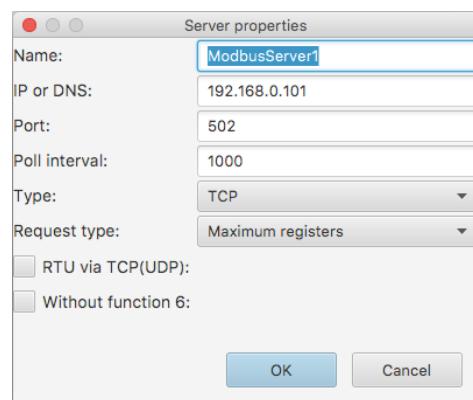
To create a new Modbus server select the menu item **Modbus**. You'll see the following window:

1. In the **Name** enter the name of the Modbus server.
2. Write IP address or DNS in the **IP or DNS** field.
3. Enter Modbus server port in the **Port**.
4. Define the polling interval of the server in the **Poll interval** field.
5. Choose communication protocol in the **Type**.
6. Choose **Request type**:

- *Maximum registers* - if you choose this type the application during polling will send maximum modbus pointers in 1 polling request.

- *Consecutive registers* - if you choose this type the application during polling will send only consecutive modbus pointers in 1 polling request.

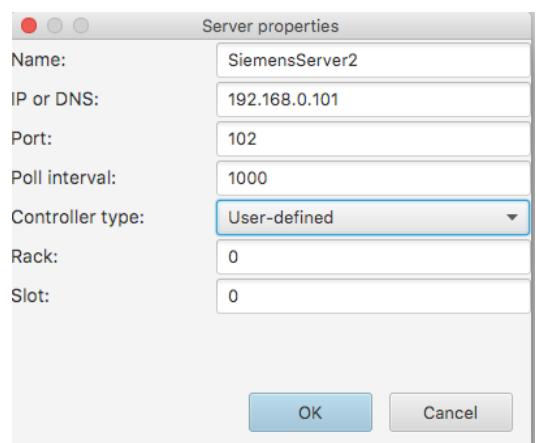
- *1 pointer registers* - if you choose this type the application during polling will send only registers used by 1 pointer in 1 polling request.
7. Check **RTU via TCP(UDP)** if you user Modbus converter from serial into TCP(UDP) protocol.
 8. Check **Without function 6** if your controller doesn't support Modbus writing function 6.



Siemens server

To create a new Siemens server select the menu item **Siemens**. You'll see the following window:

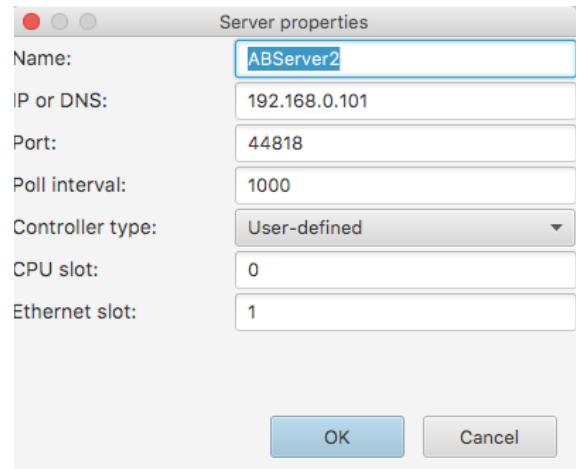
1. In the **Name** enter the name of the Siemens server.
2. Write IP address or DNS in the **IP or DNS** field.
3. Enter Siemens server port in the **Port**.
4. Define the polling interval of the server in the **Poll interval** field.
5. Choose type of the Siemens PLC in the **Controller type**.
6. Enter rack number in the **Rack** field.
7. Enter slot number in the **Slot** field.



Allen Bradley server

To create a new Allen Bradley server select the menu item **Allen Bradley**. You'll see the following window:

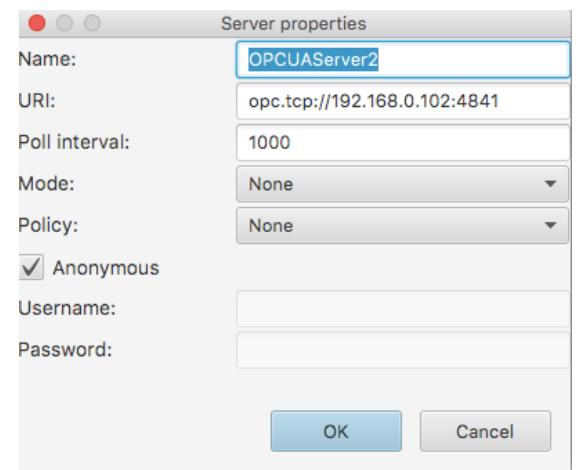
1. In the **Name** enter the name of the Allen Bradley server.
2. Write IP address or DNS in the **IP or DNS** field.
3. Enter Allen Bradley server port in the **Port**.
4. Define the polling interval of the server in the **Poll interval** field.
5. Choose type of the Allen Bradley PLC in the **Controller type**.
6. Enter PLC's cpu slot number in the **CPU slot** field.
7. Enter PLC's backplane number in the **Backplane** field.



OPC UA server

To create a new OPC UA server select the menu item **OPC UA**. You'll see the following window:

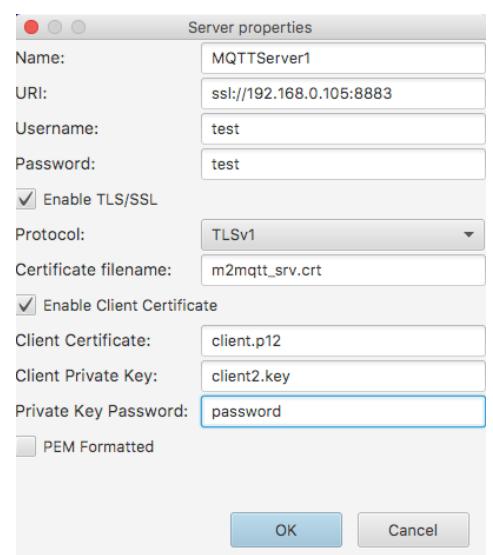
1. In the **Name** enter the name of the OPC UA server.
2. Write OPC UA server address in the **URI** field.
3. Define the polling interval of the server in the **Poll interval** field.
4. Choose security mode in the **Mode**.
5. Choose security policy in the **Policy**.
6. Check **Anonymous** if you don't use User token.
7. Enter **Username** and **Password** into relevant fields if you use User token.



MQTT server

To create a new MQTT server select the menu item **MQTT**. You'll see the following window:

1. In the **Name** enter the name of the MQTT server.
2. Write MQTT server address in the **URI** field.
3. Enter **Username** and **Password** into relevant fields.
4. Check **Enable TLS/SSL** if you want to use server certificate for encryption messages.
5. Enter **Certificate filename**. File should be placed in /private/ folder in the directory where TeslaSCADA2 Runtime execution file.
6. Check **Enable Client Certificate** if you want to use client certificate for encryption messages.
7. Enter **Client certificate*** filename. File also should be placed in /private/ folder.
8. Enter **Client Private key*** filename. File also should be placed in /private/ folder.
9. Enter **Private key password***.



10. Check **PEM formatted*** if your certificate and key files are PEM formatted.

* If you use this project for iOS (iPhone or iPad) you should use .p12 format for the file of the certificate. To create .p12 file you should in openssl utility use this type of command:

```
openssl pkcs12 -export -out [your file name].p12 -in [your file name].crt -inkey [your file name].key
```

For example:

```
openssl pkcs12 -export -out client.p12 -in client.crt -inkey client.key
```

The name of your .p12 you should place in the **Client certificate** field (client.p12 in our example). **Client Private Key** you can left empty. In the **Private key password** you should enter password of the .p12 file. **PEM formatted** you can left uncheck. All .p12 files are PEM formatted.

Open server properties

To open server properties:

1. Double click on the server properties which you want to open.

or

2. Right click on the server properties which you want to open and choose *Server properties* item.

Copy server

To copy server:

1. Right click on the server you want to copy and choose *Copy server* item.

Delete server

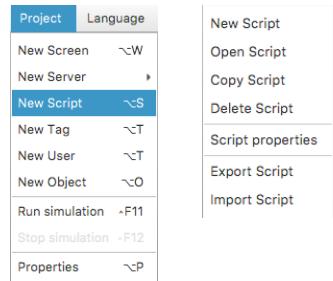
To delete server:

1. Right click on the server you want to delete and choose *Delete server* item.

Scripts

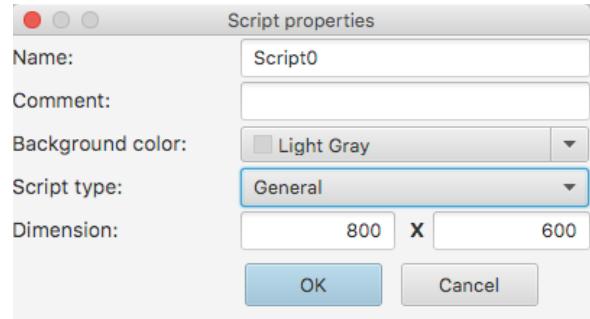
Create script

To create a new script select the menu item **Project** and **New Script** or choose **Scripts** on the **Project Window**, click right button on it and choose **New Script** item.



You'll see the following window:

1. In the **Name** enter the name of the screen.
2. Optionally, specify a meaningful **Comment**.
3. Choose **Background** color.
4. Select **Script type**: *General* or *Screen*. General script bind to the whole project. Screen script bind to the Screen.
5. Enter **Dimension** of the script's design screen.



Open script

To open script:

1. Right click on the script you want to open and choose *Open script* item.
- or
2. Double click on the script you want to open.

Copy script

To copy script:

1. Right click on the script you want to copy and choose *Copy script* item.

Delete script

To delete script:

1. Right click on the script you want to delete and choose *Delete script* item.

Edit script properties

To edit script properties:

1. Right click on the script you want to edit and choose *Script properties* item.

Export script

To export script:

1. Right click on the script you want to export and choose *Export script* item.
2. Now select the location and click the button *Save* (TeslaSCADA script extension .tsp2script).

Import script

To import script:

1. Right click on the script window and choose *Import script* item.
2. Now select the script file and click *Open* (TeslaSCADA screen extension .tsp2script).

Tags

Create tag

To create a new tag select the menu item **Project** and **New Tag** or choose **Tags** on the **Project Window**, click right button on it and choose **New Tag** item.

You'll see the following window:

On the General tab:

1. In the **Name** enter the name of the tag. The name should be unique for the project.
2. Choose **Data type**.
3. If you select *String* or *Array* data types enter **Number of elements** (letters).
4. If you select *String* or *Array* data types choose data type of **1 element** (letter).
5. Choose **Access mode** to the tag: *Read*, *Write* or *ReadWrite*.
6. Enter default tag's value into **Initial PV**.
7. In the **Input/Output** section bind tag to the server's tag. In the **PV Input server** choose server you want to bind. Then click «...» button to set up server's tag settings or enter it into the **PV Input tag**.
8. If the output server's tag differs from the input server's tag check **Output differs from input** and select **PV Output server** and enter **PV Output tag**.

Project		Language
New Screen		
New Server		
New Script		
New Tag	~T	
New User		
New Object		
Run simulation		
Stop simulation		
Properties		

Tag properties

General	Scaling	Alarms	History
Name:	Tag0		
Data type:	Boolean		
Number of elements:	10		
1 element:			
Access mode:	ReadWrite		
Initial PV:	false		
Input/Output			
PV Input server:	ModbusServer1		
PV Input tag:			
<input type="checkbox"/> Output differs from Input:			
PV Output server:	ModbusServer1		
PV Output tag:			
<input type="button" value="OK"/> <input type="button" value="Cancel"/>			

Depending on the type of **PV Input server** or **PV Output server** you'll see different server's tag (pointer) settings window:

Modbus tag settings

You'll see the following window:

1. Enter **SlaveID** of the modbus device.
2. Choose **Point type** of the register.
3. Write offset of the register into **Offset**.
4. Choose **Data type** of the modbus tag.
5. Choose number of **Bit** if the point type is boolean.

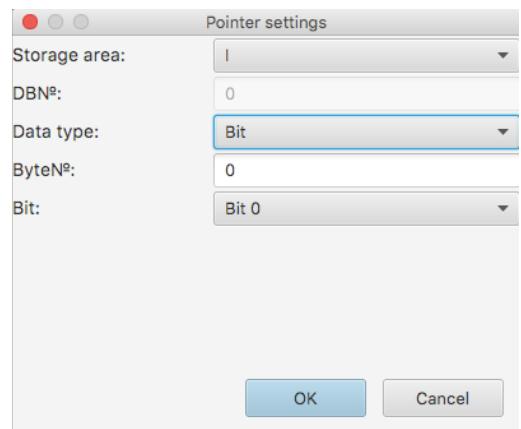
Pointer settings

SlaveID:	1
Point type:	Holding Registers
Offset:	0
Data type:	Unsigned Integer(16bit)
Bit:	none
<input type="button" value="OK"/> <input type="button" value="Cancel"/>	

Siemens tag settings

You'll see the following window:

1. Choose **Storage area** of the siemens tag: *I,Q,M* or *DB*.
2. Write DB number in the **DBNº** field if you choose DB storage area.
3. Choose **Data type** of the siemens tag.
4. Enter byte number of the area into **ByteNº** field.
5. Choose number of **Bit** if the data type is *Bit*.



AllenBradley tag settings

You'll see the following window:

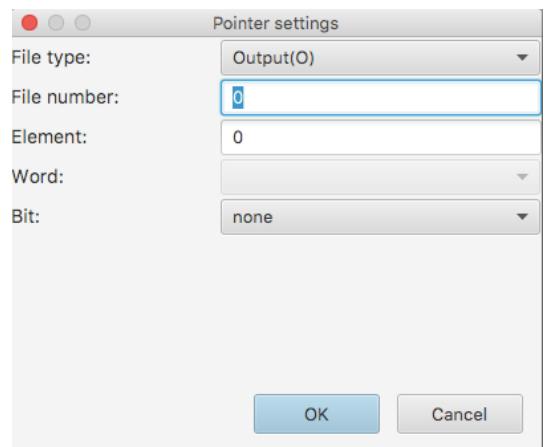
1. Enter **Tag name**.
2. Choose **Data type** of the allen bradley tag.



Micrologix tag settings

If you choose Micrologix or SLC500 controller type in the Allen Bradley server settings you'll see the following window:

1. Choose **File type** of the server's tag.
2. Write **File number** in the field.
3. Enter **Element** of the servers tag.
4. Choose **Word** for some file types.
5. Choose number of **Bit**.



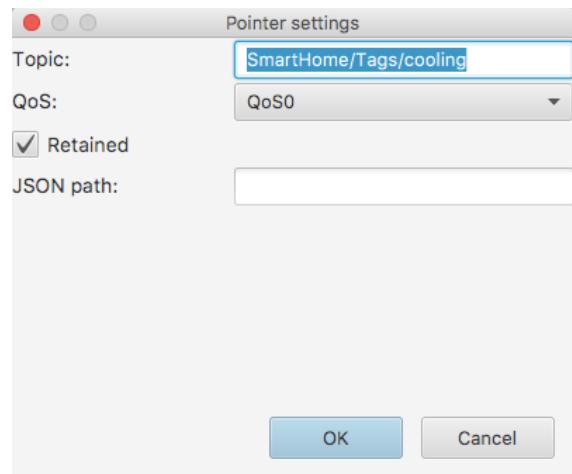
OPC UA tag settings

After clicking «...» button when you choose OPC UA server you'll get into the Address Space window. Browse through the address space by double clicking on the nodes and choose the tag(node) you need by clicking right button on it and choosing *Select* menu item on the popup window.

MQTT tag settings

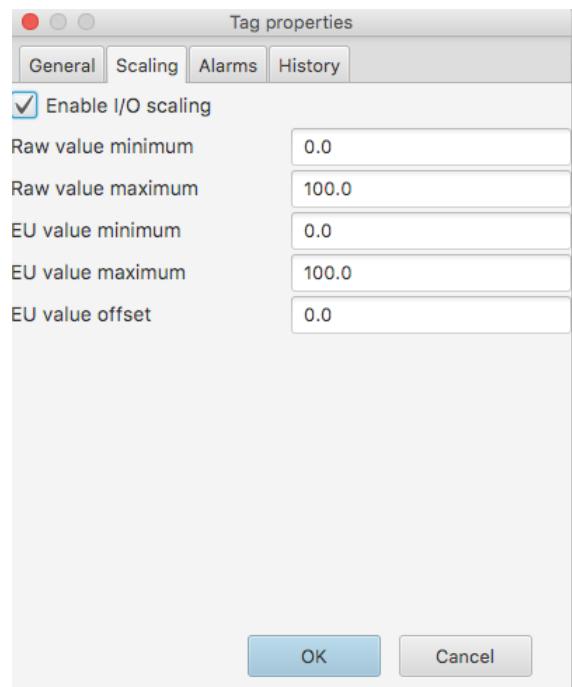
You'll see the following window:

1. Enter **Topic**.
2. Choose **QoS** of the MQTT tag.
3. Check **Retained** if you want to use this property.
4. If MQTT response contains JSON array enter **JSON path** to parse the value. For example if response is: «{foo: bar, lat: 0.23443, long: 12.3453245}» to get long value enter «long» in the field. If response is not JSON format left field empty.



On the *Scaling* tab of the *Tag properties* window:

1. Check **Enable I/O scaling** if you want to scale a value get from the server.
2. Enter minimum server tag's value into **Raw value minimum**.
3. Enter maximum server tag's value into **Raw value maximum**.
4. Enter minimum tag's value in engineer units into **EU value minimum**.
5. Enter maximum tag's value in engineer units into **EU value maximum**.
6. Write tag's value offset int **EU value offset**.

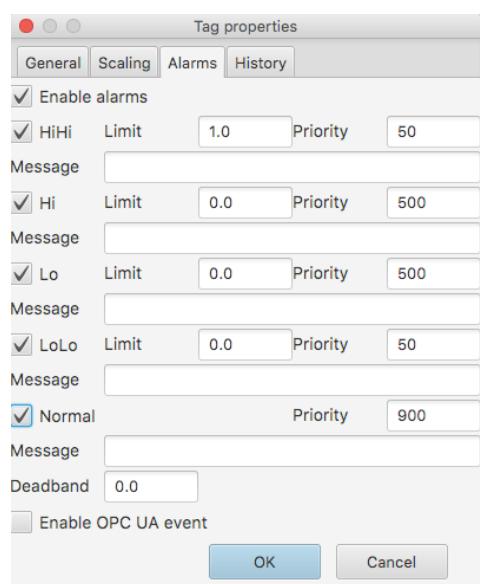


When you get some value from the server application use this formula:

$$\text{value} = (\text{value}-\text{rawmin}) * (\text{eumax}-\text{eumin}) / (\text{rawmax}-\text{rawmin}) + \text{eumin} + \text{offset}$$

On the *Alarms* tab of the *Tag properties* window:

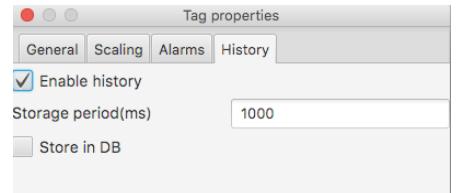
1. Check **Enable alarms** if you want to use alarms for this tag.
2. Check **HiHi**, **Hi**, **Lo**, **LoLo** or **Normal** if you want to use the correspondent alarm(event).
3. Write **Limit** for the correspondent alarm(event). If the value of the tag plus **Deadband** will be more than *HiHi* or *Hi* limit the correspondent alarm will be called and be written into Event database. If the value of the tag minus **Deadband** will be less than *LoLo* or *Lo* limit the correspondent alarm will be raised and be written into Event database.



4. Enter **Priority** for the correspondent alarm(event). If the priority of the alarm(event) is less than value of **Notifications(Priority)** you set in the project properties the notification dialog will be called.
5. Enter **Message** for the correspondent alarm(event).
6. Check **Enable OPC UA event** if you bind this tag to the OPC UA server tag(node) and you want to use EventNotifier of this tag(node).

On the *History* tab of the *Tag properties* window:

1. Check **Enable history** if you want to storage values of this tag.
2. Enter **Storage period(ms)**.
3. Check **Store in DB** if you want to store data in history database.



Copy tag

To copy tag:

1. Right click on the tag you want to copy and choose *Copy tag* item.

Delete tag

To delete tag:

1. Right click on the tag you want to delete and choose *Delete tag* item.

Edit tag properties

To edit tag properties:

1. Right click on the script you want to edit and choose *Tag properties* item.
or
2. Double click on the tag you want to edit.

Export all tags

To export all tags:

1. Right click on the tags window and choose *Export all tags* item.
2. Now select the location and click the button *Save* (TeslaSCADA tags extension .tsp2tags).

Import tags

To import tags:

1. Right click on the tags window and choose *Import tags* item.
2. Now select the tags file and click *Open* (TeslaSCADA screen extension .tsp2tags).

Users

Create user

User is not a mandatory element of the project. You can use or not use in it. To create a new user select the menu item *Project* and *New User* or choose **Users** on the **Project Window**, click right button on it and choose *New User* item.

You'll see the following window:

1. In the **Name** enter the name of the user.
2. Write **Password** for the current user.
3. Check **Control functions** if you want that current user can write values into the server's tags.
4. Check **Acknowledge events** if you want that current user can acknowledge events in events database.
5. Check **Delete events** if you want that current user can delete events from events database.
6. Check **Insert events** if you want that runtime application insert events into events database when current user is logged in.
7. Check **Insert history** if you want that runtime application insert history data into history database when current user is logged in.
8. Check **Settings** if you want current user can enter *Settings* menu of runtime application.
9. Check **Edit recipes** if you want current user can Add, Edit and delete recipes fields.
10. Check **Save control operation** if you want to save user control operations in Events database.
(it will be saved if you check Enable alarms in Tag properties).
11. **Priority** of the user control operations events.

Open user properties

To open user properties:

1. Right click on the user you want to open and choose *User properties* item.
- or
2. Double click on the user properties which you want to open.

Copy user

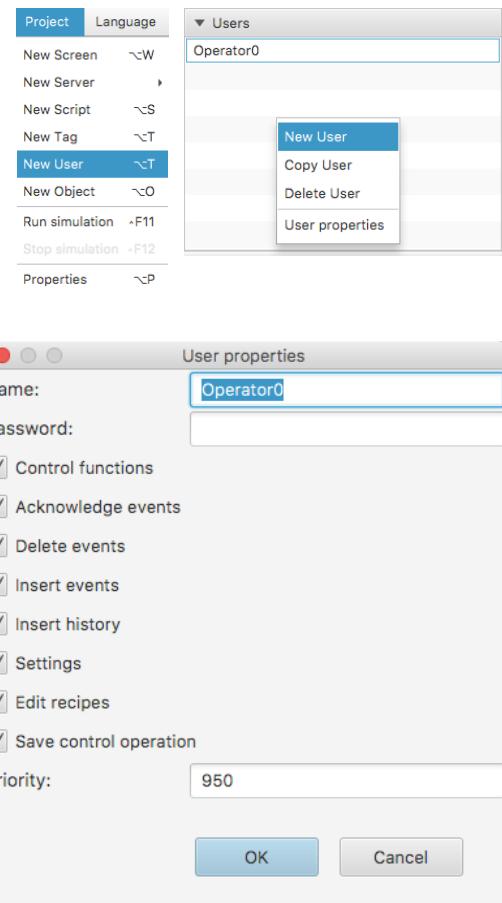
To copy user:

1. Right click on the user you want to copy and choose *Copy user* item.

Delete user

To delete user:

1. Right click on the user you want to delete and choose *Delete user* item.

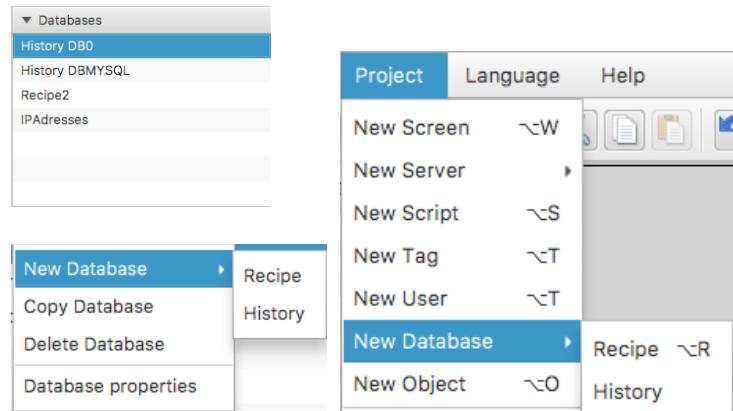


Databases

Database is not a mandatory element of the project. You can use or not use in it. Database consists of 2 types: Recipe and History.

Create recipe

To create a new recipe select the menu item *Project* and *New Database ->Recipe* or choose **Databases** on the **Project Window**, click right button on it and choose *New Database->Recipe* item.

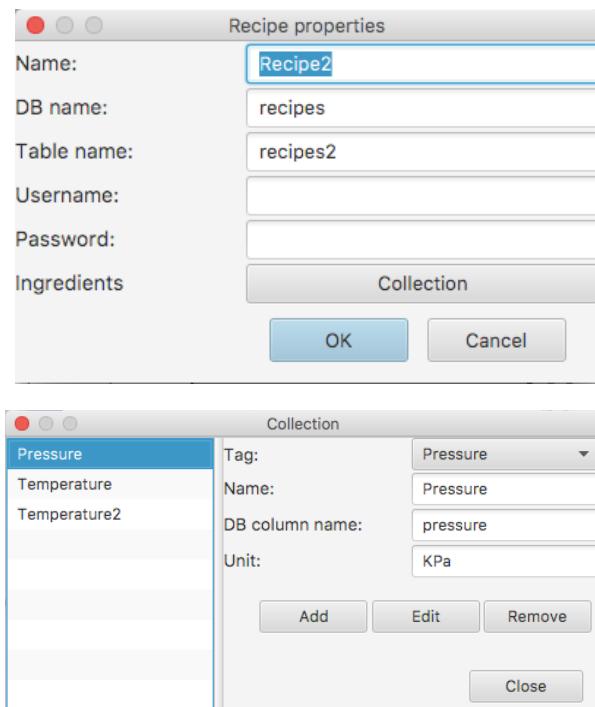


You'll see the following window of recipe properties:

1. In the **Name** enter the name of the recipe.
2. Write **DB name** for the current recipe. If you enter the simple name like *recipes* for example you will connect to the SQLite database. The SQLite database file .db will be created in /DB/ folder. If you choose names beginning with **jdbc:mysql:** like *jdbc:mysql:/192.168.0.104:3306/test* the application will connect to MySQL database (for iOS you can't use MySQL databases at this time).
3. Write **Table name** of the current database.
4. Write **Username** if needed for MySQL databases.
5. Write **Password** if needed for MySQL database.
6. Click **Collection** to fill up ingredients of the recipe.

After clicking **Collection** button you'll see the following window. Where:

1. Choose **Tag** you want to bind to the ingredient.
2. Enter **Name** of the ingredient.
3. Enter **DB column name** for the database.
4. Enter **Unit** of the DB ingredient.

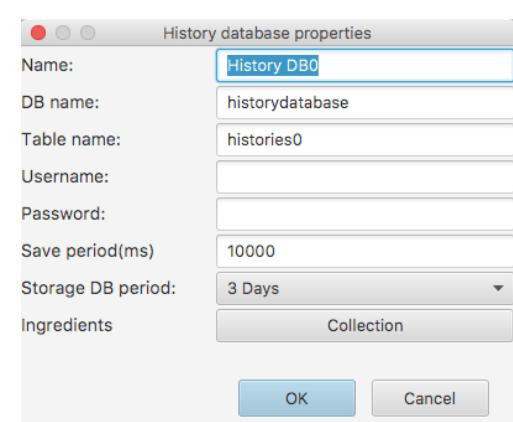


Create history

To create a new history select the menu item *Project* and *New Database ->History* or choose **Databases** on the **Project Window**, click right button on it and choose *New Database->History* item.

You'll see the following window of recipe properties:

1. In the **Name** enter the name of the history db.
2. Write **DB name** for the current history. If you enter the simple name like *history* for example you will connect to the SQLite database. The SQLite database file .db will be

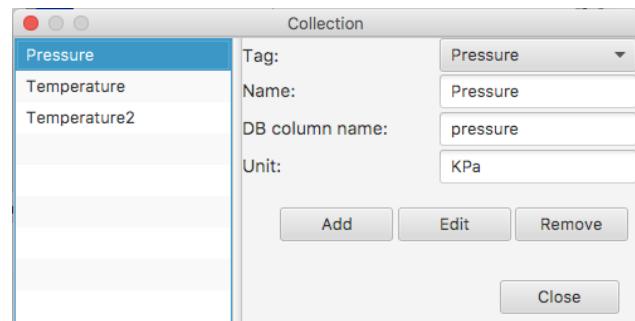


created in /DB/ folder. If you choose names beginning with **jdb:mysql:** like **jdb:mysql:/192.168.0.104:3306/test** the application will connect to MySQL database (for iOS you can't use MySQL databases at this time).

3. Write **Table name** of the current database.
4. Write **Username** if needed for MySQL databases.
5. Write **Password** if needed for MySQL database.
6. Enter **Save Period(ms)**. The retention period of the selected tags in the database.
7. Choose **Storage DB period**. It's a period of storing history information.
8. Click **Collection** to fill up ingredients of the recipe.

After clicking **Collection** button you'll see the following window. Where:

1. Choose **Tag** you want to bind to the ingredient.
2. Enter **Name** of the ingredient.
3. Enter **DB column name** for the database.
4. Enter **Unit** of the DB ingredient.



Open database properties

To open database properties:

1. Right click on the recipe you want to open and choose *Database properties* item.
or
2. Double click on the database properties which you want to open.

Copy database

To copy database:

1. Right click on the database you want to copy and choose *Copy database* item.

Delete database

To delete database:

1. Right click on the database you want to delete and choose *Delete database* item.

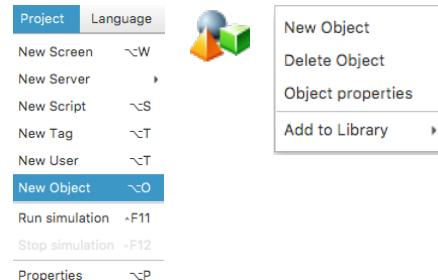
Design screen

To start designing the screen you want, you should double click on it or click right button on the **Project window->Screens** and choose *Open screen*.

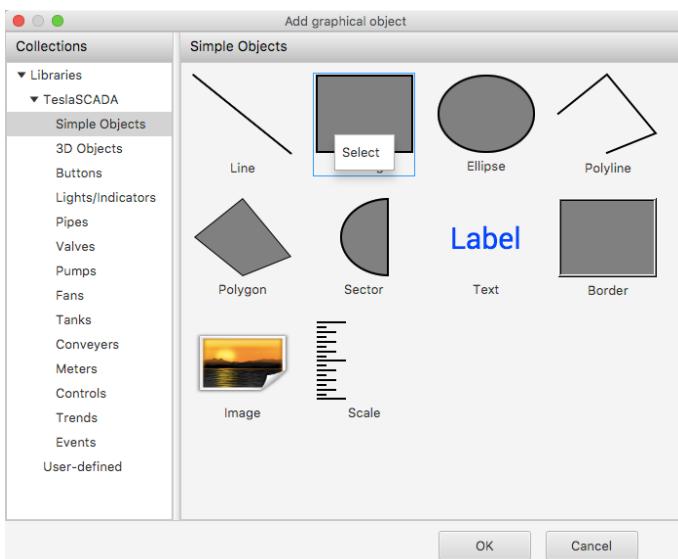
Create graphical object

Add new graphical object object on the screen you can in several ways:

1. Select the menu item *Project* and *New Object*.
2. Click **New Object** button on the Toolbar.
3. Click right button on the **Screen window** and choose *New object* item.
4. Click right button on the **Canvas** and choose *New object* item.



You'll see the **Add graphical object** window:



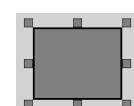
Select library which object you want to use in your project (all libraries and their objects described below). Select object you can in several ways:

1. By double clicking on the object.
2. By clicking on the object (select rectangle will appear) and then clicking OK button.
3. By clicking right button and choosing *Select* item.

Add graphical object window will disappear and you can select the location on the screen where you want to place an object.

Object information about its dimensions and coordinates you can find in the status bar on the right.

X, Y | 260 | 240 | W, H | 100 | 80



appear and

Resize graphical object

You can resize graphical object by clicking on it. Resize squares will you can change dimensions of your object as you want.

Move graphical object

You can move graphical objects by Drag and Drop technology.

Open graphical object properties

You can open graphical object properties on the **Screen Window** or on the **Canvas**. To open graphical object properties:

1. Right click on the object you want to open and choose *Object properties* item.
or
2. Double click on the object properties which you want to open.

Copy graphical object

You can copy graphical object:

1. Right click on the object you want to copy and choose *Copy* item.
2. Select the object you want to copy and choose *Edit->Copy* menu item.
3. Select the object you want to copy and click *Copy* button on the **Toolbar**.

Cut graphical object

You can cut graphical object:

1. Right click on the object you want to cut and choose *Cut* item.
2. Select the object you want to cut and choose *Edit->Cut* menu item.
3. Select the object you want to cut and click *Cut* button on the **Toolbar**.

Paste graphical object

You can paste (before cut or copied) graphical object:

1. Right click on the **Canvas** and choose *Paste* item.
2. Choose *Edit->Paste* menu item.
3. Click *Paste* button on the **Toolbar**.

Erase graphical object

You can erase graphical object:

1. Right click on the object you want to erase and choose *Erase* item.
2. Select the object you want to erase and choose *Edit->Erase* menu item.
3. Right click on the object in the **Screen Window** and choose *Delete object* item.

Duplicate graphical object

You can duplicate graphical object:

1. Right click on the object you want to duplicate and choose *Duplicate* item.
2. Select the object you want to duplicate and choose *Edit->Duplicate* menu item.

Send to back graphical object

You can send to back graphical object relative to other objects of the screen:

1. Right click on the object you want to send to back and choose *Send to Back* item.
2. Select the object you want to send to back and choose *Arrange->Send to Back* menu item.
3. Select the object you want to send to back and click *Send to Back* button on the **Toolbar**.

Bring to front graphical object

You can bring to front graphical object relative to other objects of the screen:

1. Right click on the object you want to bring to front and choose *Bring to Front* item.
2. Select the object you want to bring to front and choose *Arrange->Bring to Front* menu item.
3. Select the object you want to bring to front and click *Bring to Front* button on the **Toolbar**.

Rotate clockwise graphical object

You can rotate clockwise graphical object clockwise:

1. Select the object you want to rotate clockwise and click *Rotate Clockwise* button on the **Toolbar**.

2. Select the object you want to rotate clockwise and choose *Arrange->Rotate Clockwise* menu item.

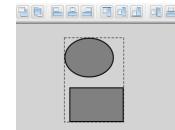
Rotate counterclockwise graphical object

You can rotate counterclockwise graphical object clockwise:

1. Select the object you want to rotate counterclockwise and click *Rotate CounterClockwise* button on the **Toolbar**.
2. Select the object you want to rotate counterclockwise and choose *Arrange->Rotate CounterClockwise* menu item.

Align graphical objects

You can align objects relative to each other on the screen. Choose objects you want to align by selecting square. And:



1. Choose *Arrange->Align* menu items.
2. Click *Align* buttons on the **Toolbar**.
3. Right click on selecting square and choose *Align* item.

For more information about each alignment operation you can read above in section **Start TeslaSCADA IDE ->Toolbar**.

Group graphical objects

You can group objects. Choose objects you want to group by selecting square. And:

1. Select *Arrange->Group objects* menu item.
2. Click *Group objects* button on the **Toolbar**.
3. Right click on selecting square and choose *Group objects* item.

Ungroup graphical objects

You can ungroup objects. Choose group of objects you want to ungroup by clicking on it . And:

1. Select *Arrange->Ungroup objects* menu item.
2. Click *Ungroup objects* button on the **Toolbar**.
3. Right click on selecting square and choose *Ungroup objects* item.

Graphical objects

Each graphical object has several group of properties. The description of each group of properties you can find below in the chapter - **Properties**. In this chapter we describe one group for every object - **General**.

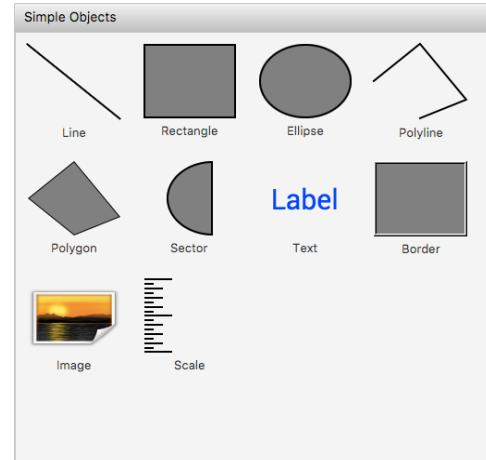
This group is responsible for the appearance of the object. Each object has the following properties:

1. **Name** - write name of the object in this field.
2. **Dimensions** - dimensions of the graphical object. Enter width of the object in the **W** field and enter height of the object in the **H** field.
3. **Coordinates** - coordinates of the graphical object. Write x coordinates of the object in the **X** field and enter y coordinates of the object in the **Y** field.
4. **Angle** - select the angle of rotation of the object.
5. Almost all objects has **Type** property to select the type of the object - 2D or 3D.

Simple Objects library

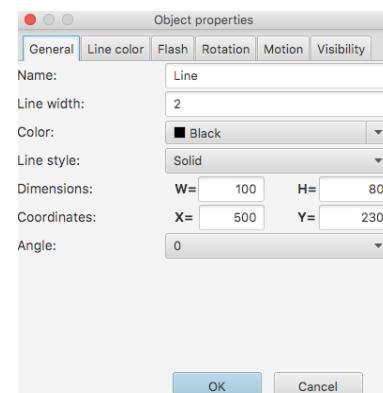
Simple objects library contains the following objects:

Line, Rectangle, Ellipse, Polyline, Polygon, Sector,
Text, Border, Image and Scale.



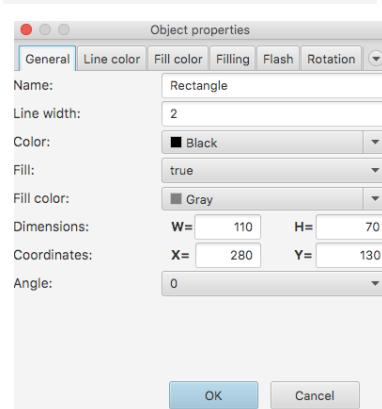
Line

1. Write width of the line in the **Line width** field.
2. Choose **Color** of the line.
3. Choose **Line style**: *Solid, Dash, Dot* or *DashDot*.



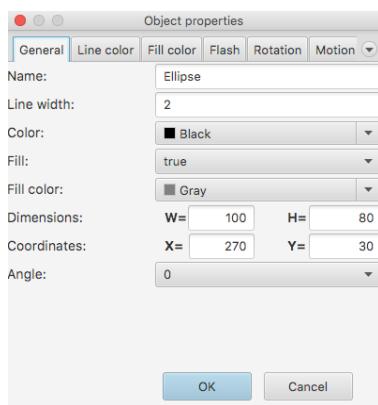
Rectangle

1. Write width of the line in the **Line width** field.
2. Choose **Color** of the border line.
3. Select fill or not this rectangle in the **Fill** combobox.
4. Choose **Fill color** of the rectangle.



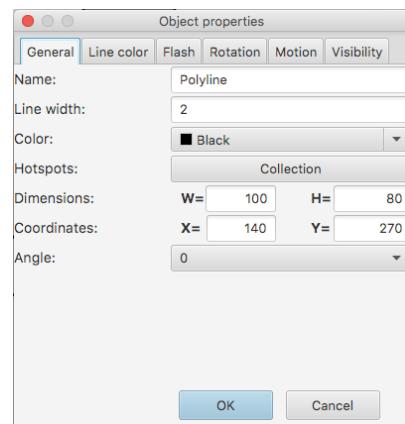
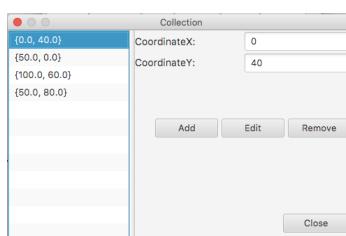
Ellipse

1. Write width of the line in the **Line width** field.
2. Choose **Color** of the border line.
3. Select fill or not this ellipse in the **Fill** combobox.
4. Choose **Fill color** of the ellipse.



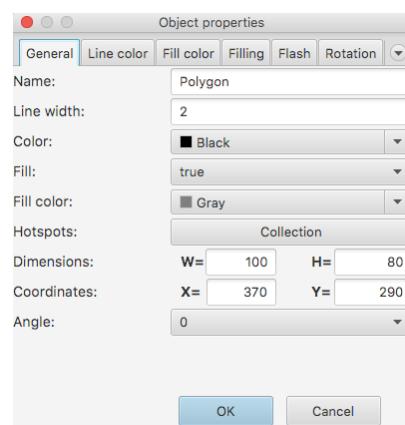
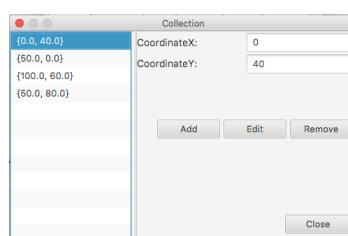
Polyline

1. Write width of the line in the **Line width** field.
2. Choose **Color** of the line.
3. When you click **Collection** button the Collection window will appear. You can *Add*, *Edit* or *Remove* nodes of the polyline.



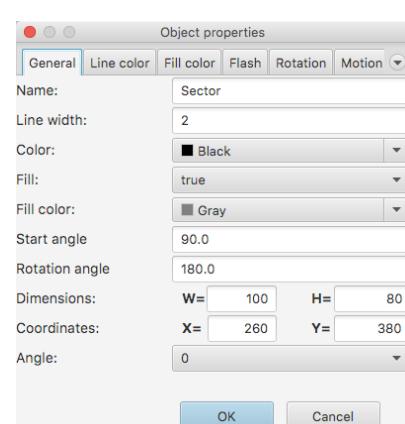
Polygon

1. Write width of the line in the **Line width** field.
2. Choose **Color** of the border line.
3. Select fill or not this polygon in the **Fill** combobox.
4. Choose **Fill color** of the polygon.
5. When you click **Collection** button the Collection window will appear. You can *Add*, *Edit* or *Remove* nodes of the polygon.



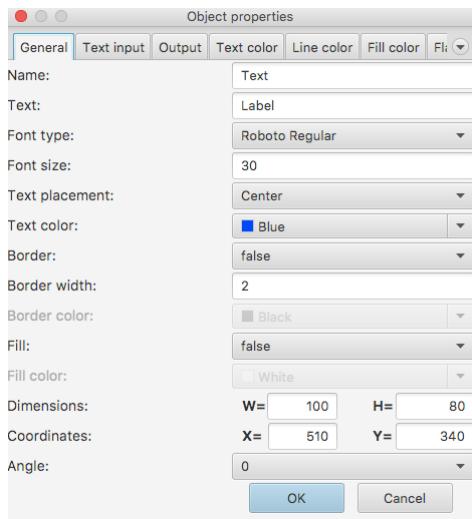
Sector

1. Write width of the line in the **Line width** field.
2. Choose **Color** of the border line.
3. Select fill or not this sector in the **Fill** combobox.
4. Choose **Fill color** of the sector.
5. Enter **Start angle** of the sector in the field. 0 degrees is right middle point of the dimensions rectangle.
6. Write **Rotation angle** in the field. Counterclockwise rotation.



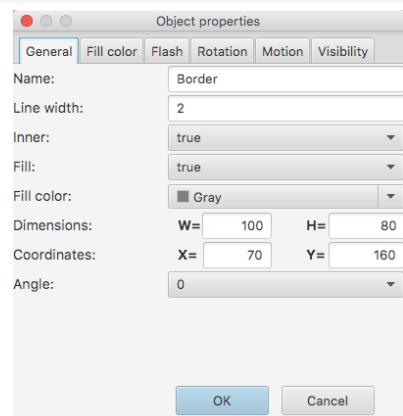
Text

1. Write text in the **Text** field.
2. Choose **Font type** of the text.
3. Enter **Font size** in the field.
4. Select **Text placement**: *Left, Center or Right*.
5. Choose **Text color**.
6. Select use or not **Border** around text.
7. Write width of the border in the **Border width** field.
8. Choose **Border color**.
9. Select fill or not text background in the **Fill** combobox.
10. Choose **Fill color** of the text background.



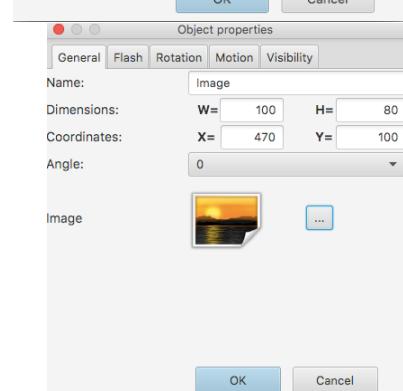
Border

1. Write width of the border in the **Line width** field.
2. Select **Inner** or not border style.
3. Select fill or not this border in the **Fill** combobox.
4. Choose **Fill color** of the border.



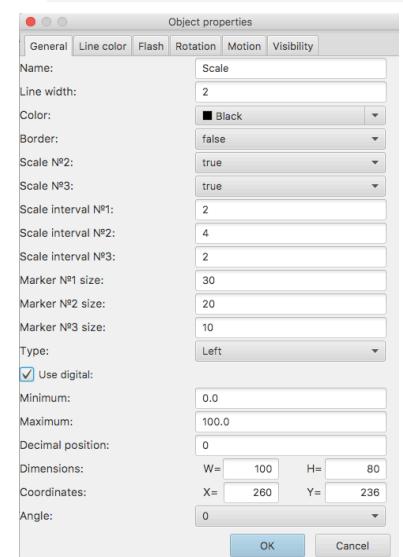
Image

1. Select **Image** you want to add to the project by clicking «...» button. Open file dialog will appear. Choose file with image you want to add to the project and click Open button.



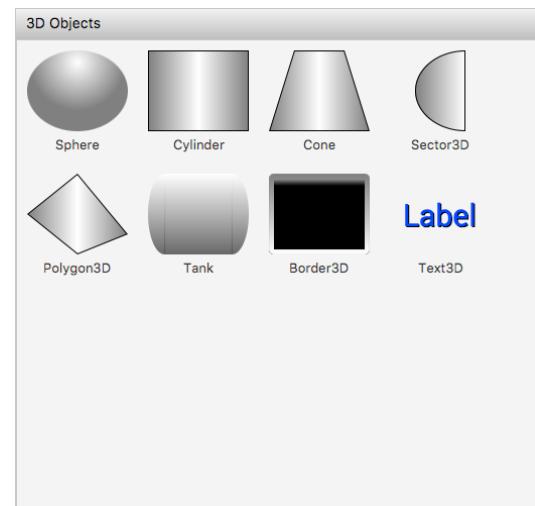
Scale

1. Write width of the line in the **Line width** field.
2. Choose **Color** of the border and scale lines.
3. Select use or not **Border** for this scale.
4. Select use or not **Scale №2** for this scale.
5. Select use or not **Scale №3** for this scale.
6. Enter **Scale interval №1** in the field.
7. Enter **Scale interval №2** in the field.
8. Enter **Scale interval №3** in the field.
9. Write width of the scale №1 in the **Marker №1 size** field.
10. Write width of the scale №2 in the **Marker №2 size** field.
11. Write width of the scale №3 in the **Marker №3 size** field.
12. Choose **Type** of the scale: Left, Right, Top or Bottom.
13. Check **Use digital** if you want to bind numeration to Scale №1.
14. Enter **Minimum** value of Scale №1.
15. Enter **Maximum** value of Scale №1.
16. Enter **Decimal position** of scale's numbers.



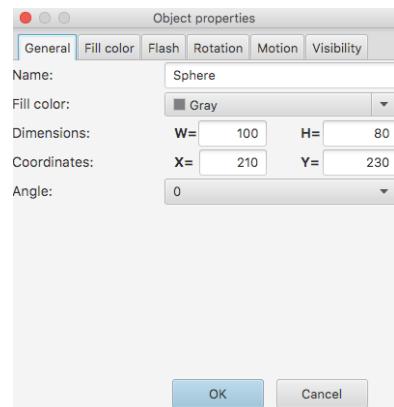
3D Objects library

3D objects library contains the following objects: Sphere, Cylinder, Cone, Sector 3D, Polygon 3D, Tank, Border 3D, Text 3D.



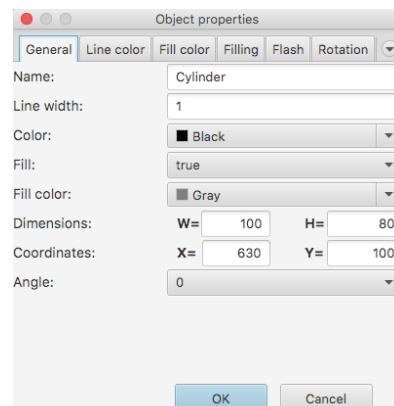
Sphere

1. Choose **Fill color** of the sphere.



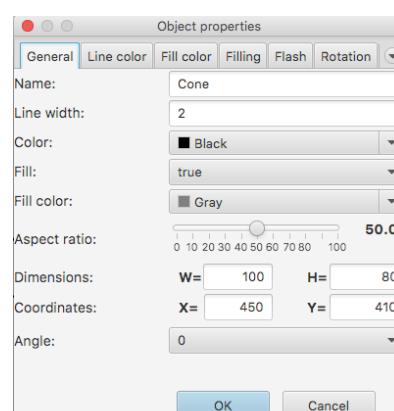
Cylinder

1. Write width of the line in the **Line width** field.
2. Choose **Color** of the border line.
3. Select fill or not this cylinder in the **Fill** combobox.
4. Choose **Fill color** of the cylinder.



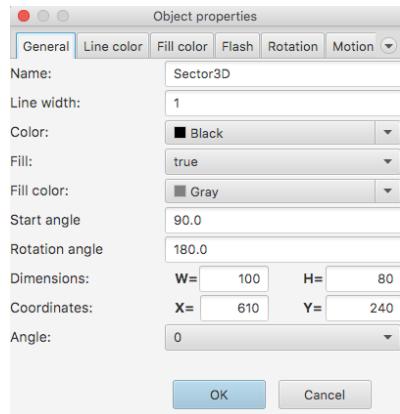
Cone

1. Write width of the line in the **Line width** field.
2. Choose **Color** of the border line.
3. Select fill or not this cone in the **Fill** combobox.
4. Choose **Fill color** of the cone.
5. Select **Aspect ratio** of the cone.



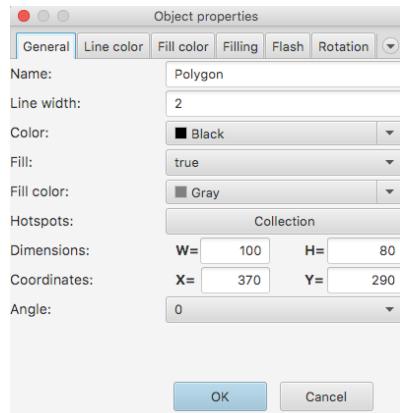
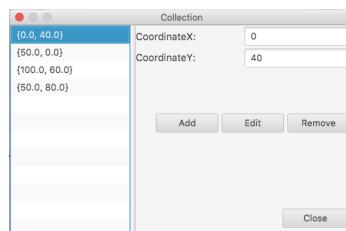
Sector 3D

1. Write width of the line in the **Line width** field.
2. Choose **Color** of the border line.
3. Select fill or not this sector in the **Fill** combobox.
4. Choose **Fill color** of the sector.
5. Enter **Start angle** of the sector in the field. 0 degrees is right middle point of the dimensions rectangle.
6. Write **Rotation angle** in the field. Counterclockwise rotation.



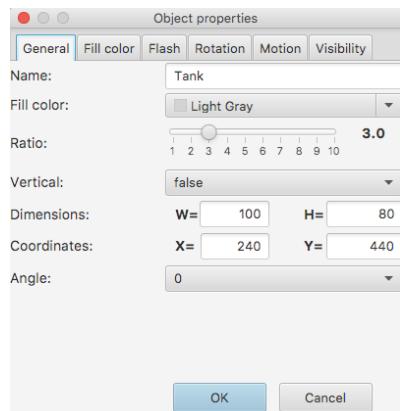
Polygon 3D

1. Write width of the line in the **Line width** field.
2. Choose **Color** of the border line.
3. Select fill or not this polygon in the **Fill** combobox.
4. Choose **Fill color** of the polygon.
5. When you click **Collection** button the Collection window will appear. You can *Add*, *Edit* or *Remove* nodes of the polygon.



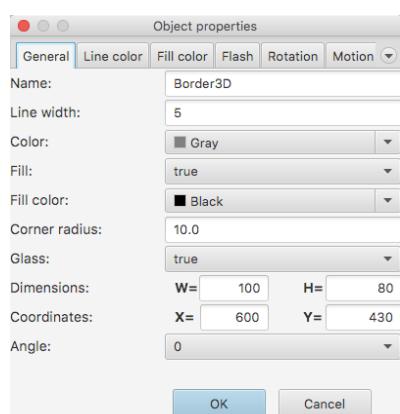
Tank

1. Choose **Fill color** of the tank.
2. Select **Ratio** of the tank.
3. Select vertical or horizontal will be tank in **Vertical** combobox.



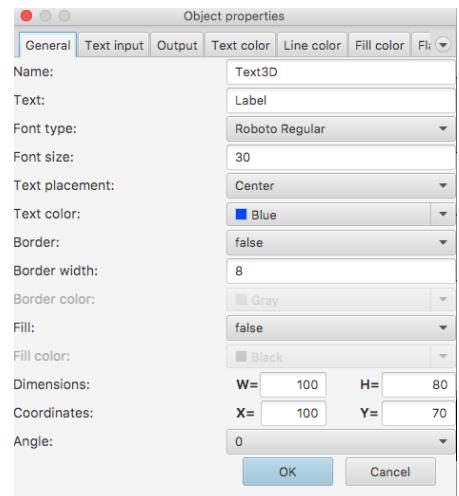
Border 3D

1. Write width of the border in the **Line width** field.
2. Choose **Color** of the border.
3. Select fill or not this border in the **Fill** combobox.
4. Choose **Fill color** of the border.
5. Enter radius of the border's corner in the **Corner radius** field.
6. Select use or not **Glass** effect.



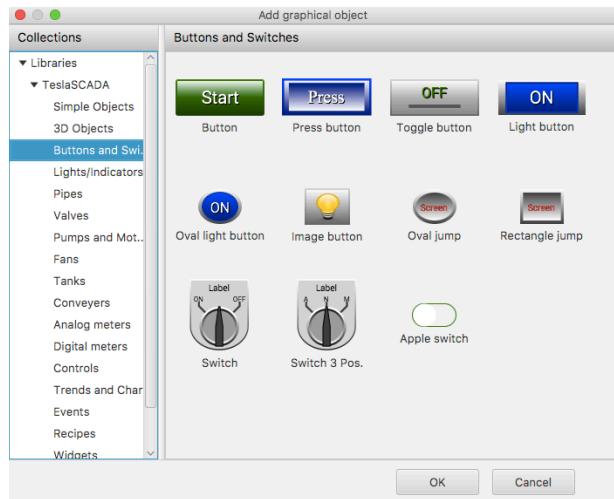
Text 3D

1. Write text in the **Text** field.
2. Choose **Font type** of the text.
3. Enter **Font size** in the field.
4. Select **Text placement: Left, Center or Right**.
5. Choose **Text color**.
6. Select use or not **Border** around text.
7. Write width of the border in the **Border width** field.
8. Choose **Border color**.
9. Select fill or not text background in the **Fill** combobox.
10. Choose **Fill color** of the text background.



Buttons and Switches library

Buttons and Switches library contains the following objects: Button, Press button, Toggle button, Light button, Oval light button, Image button, Oval jump, Rectangle jump, Switch, Switch 3 Pos and Apple switch. All buttons except Image button have the same General group properties. Below we describe there only for 5 graphical objects - **Button**, **Image button**, **Switch**, **Switch 3 Pos** and **Apple switch**.



Button

1. Write text of the button in the **Text** field.
2. Choose **Text color**.
3. Choose **Fill color** of the button.
4. Choose **Font type** of the button's label.

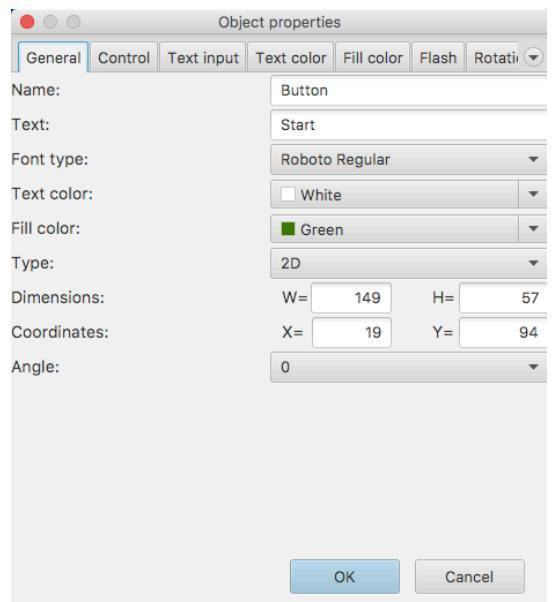
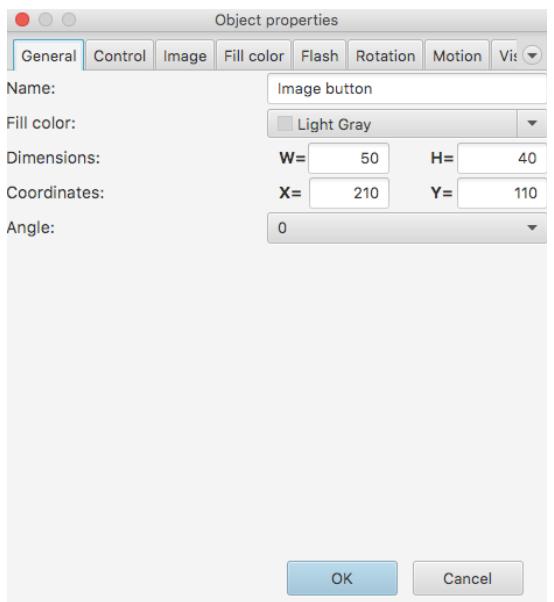


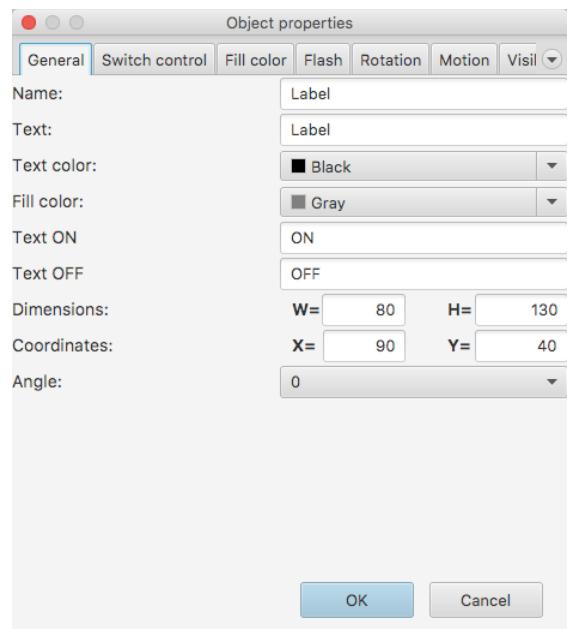
Image Button

1. Choose **Fill color** of the button's background.



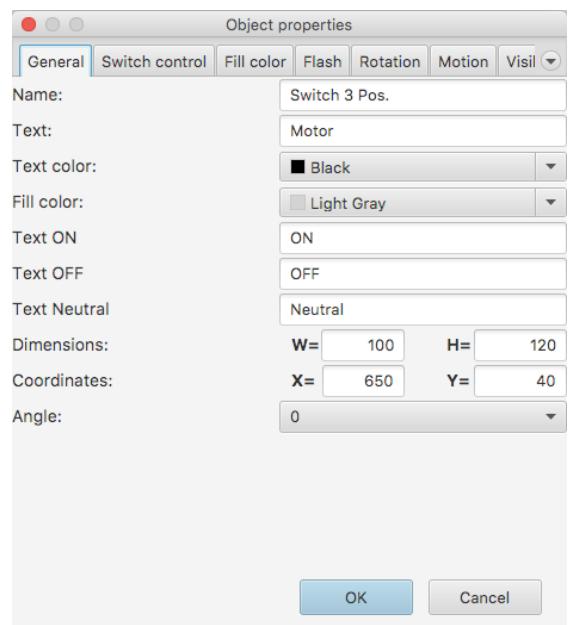
Switch and Apple switch

1. Write label of the switch in the **Text** field.
2. Choose **Text color**.
3. Choose **Fill color** of the switch.
4. Write label for ON position of the switch in **Text ON**.
5. Write label for OFF position of the switch in **Text OFF**.



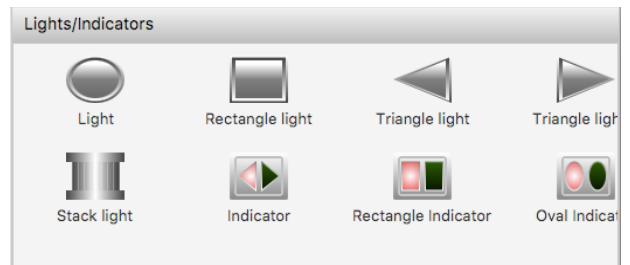
Switch 3 Pos.

1. Write label of the switch in the **Text** field.
2. Choose **Text color**.
3. Choose **Fill color** of the button.
4. Write label for ON position of the switch in **Text ON**.
5. Write label for OFF position of the switch in **Text OFF**.
6. Write label for Neutral position of the switch in **Text Neutral**.



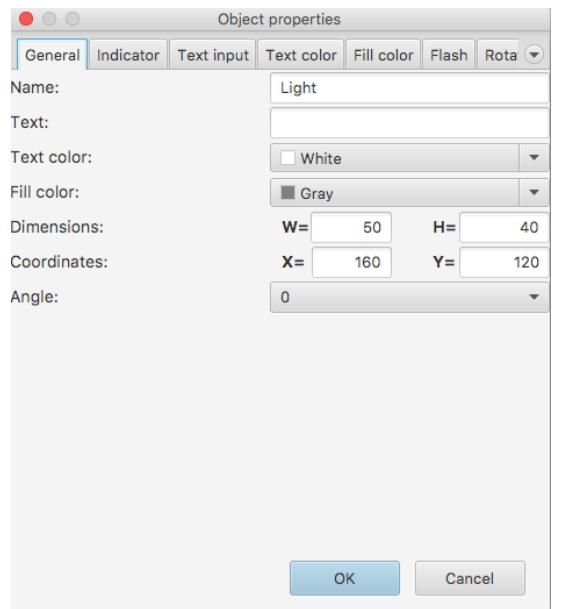
Lights/Indicators library

Lights/Indicators library contains the following objects:
Light, Rectangle light, Triangle light, Triangle light 2,
Stack light, Indicator, Rectangle Indicator and Oval
Indicator. All lights have the same General group
properties and all indicators have the same General group
properties. Below we describe it only for two graphical
objects - **Light** and **Indicator**.



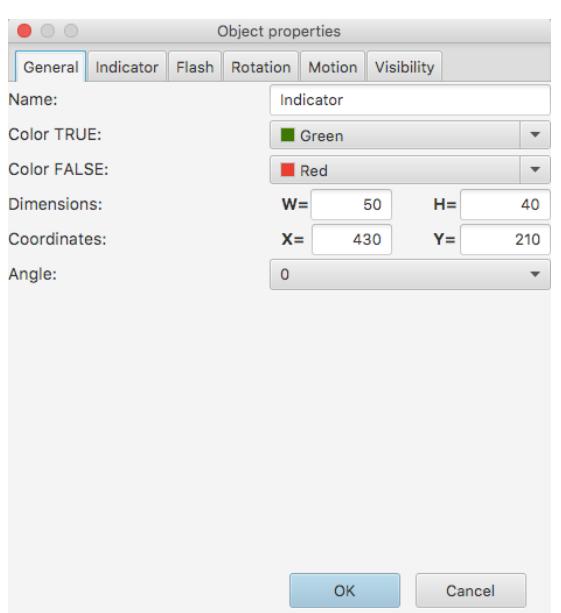
Light

1. Write text of the light in the **Text** field.
2. Choose **Text color**.
3. Choose **Fill color** of the light.



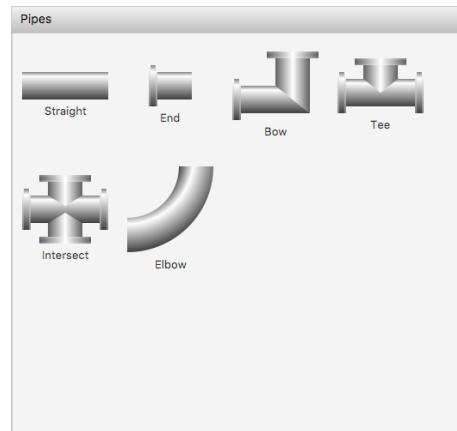
Indicator

1. Choose **Color TRUE** of the indicator.
2. Choose **Color FALSE** of the indicator.



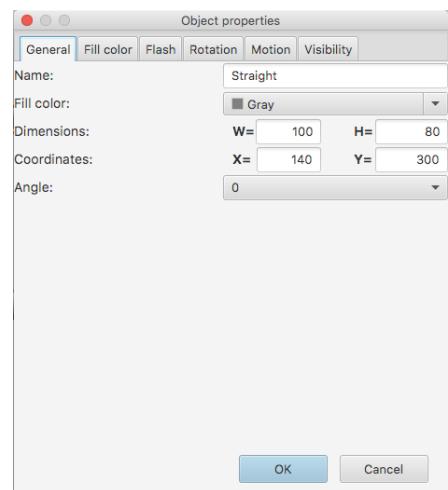
Pipes library

Pipes library contains the following pipes objects: Straight, End, Bow, Tee, Intersect and Elbow. All pipes have the same General group properties. Below we describe it only for one graphical object - **Straight**.



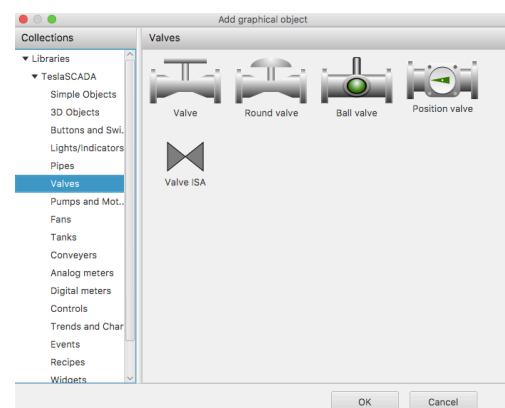
Pipe

1. Choose **Fill color** of the pipe.



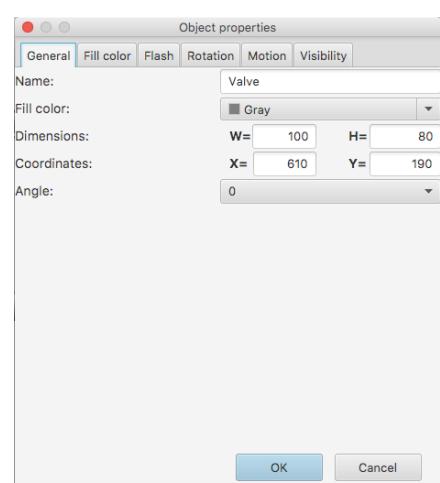
Valves library

Valves library contains the following objects: Valve, Round valve, Ball valve, Position valve and Valve ISA. Valve, Round valve and Valve ISA have the same General properties:



Valve

1. Choose **Fill color** for the valve.

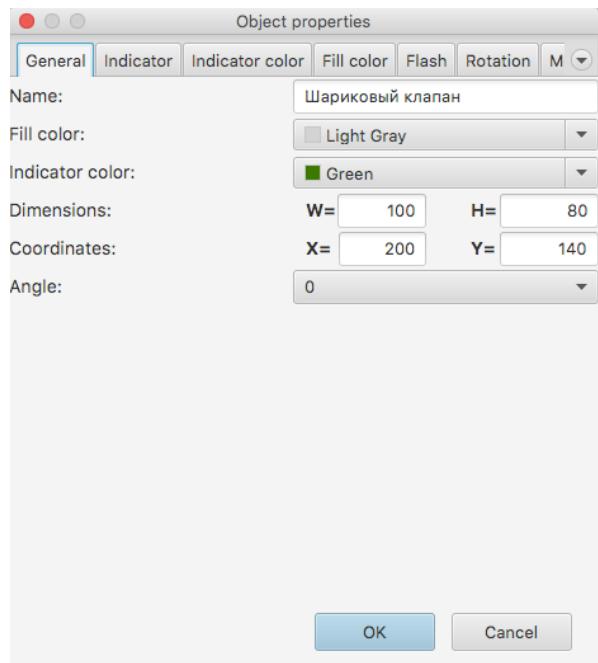


Ball valve

1. Choose **Fill color** for the valve.
2. Choose **Indicator color**.

Indicator property like indicator property for other graphical object.

Indicator color property like other color properties.

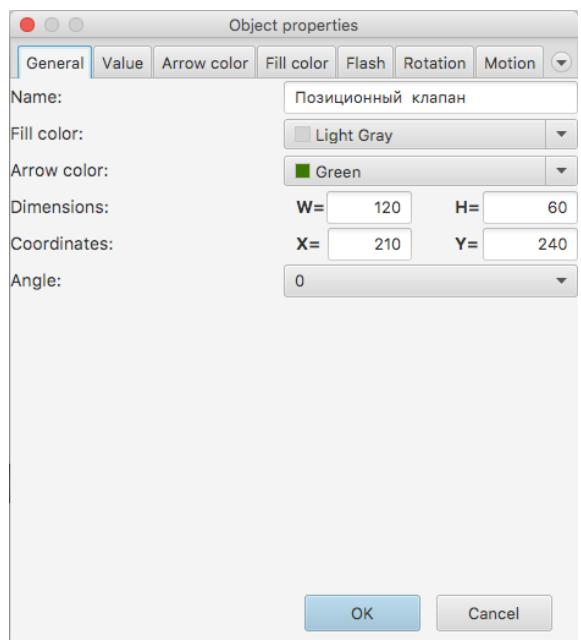


Position valve

1. Choose **Fill color** for the valve.
2. Choose **Arrow color**.

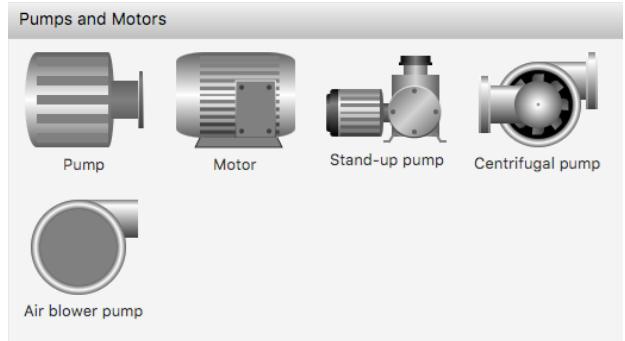
Value property like value property for analog meters.

Arrow color property like other color properties.



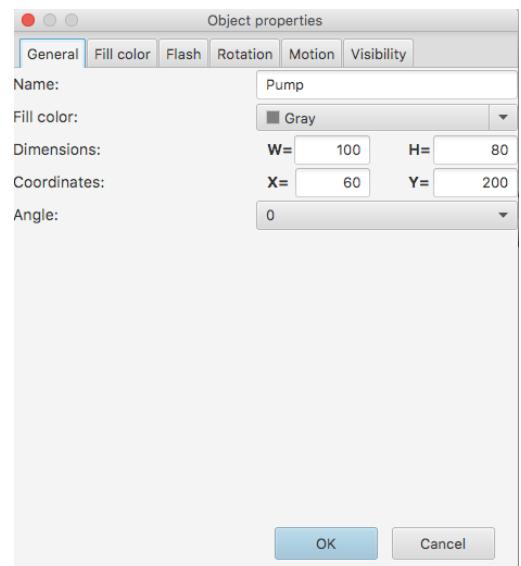
Pumps library

Pumps library contains the following objects: Pump, Motor, Stand-up pump, Centrifugal pump and Air blower pump. All pumps have the same General group properties. Below we describe there only for one graphical object - **Pump**.



Pump

1. Choose **Fill color** of the pump.



Fans library

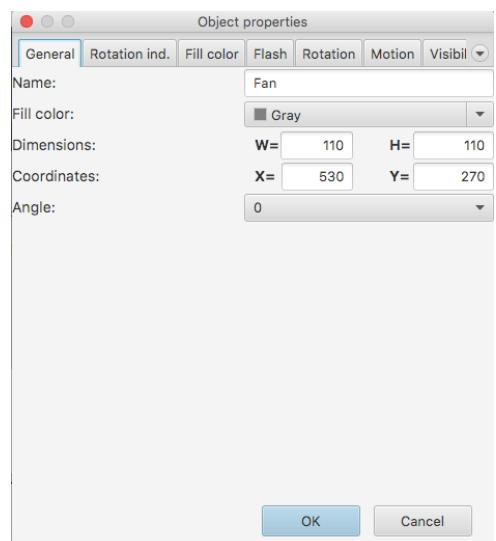
Fans library contains the following objects: Fan, Round fan and Square fan. All fans have the same properties:



Fan

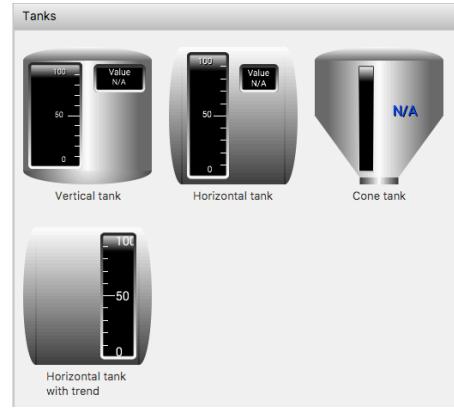
1. Choose **Fill color** of the fan.

Set up **Rotation ind.** properties to rotate fan.



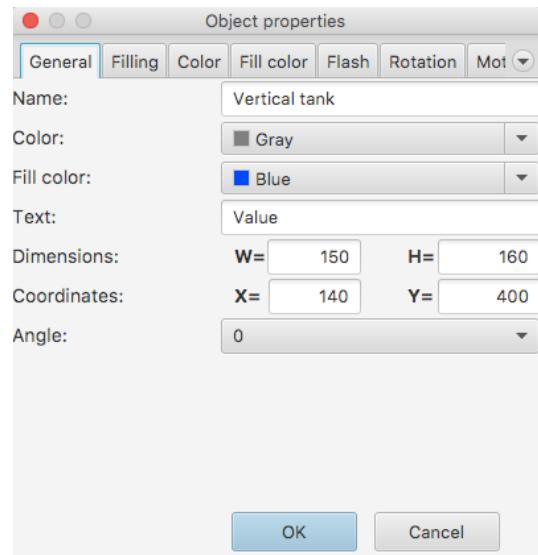
Tanks library

Tanks library contains the following objects: Vertical tank, Horizontal tank, Vertical tank with trend, Horizontal tank with trend and Cone tank. All tanks have the same General group properties. Below we describe it only for one graphical object - **Vertical tank**.



Vertical tank

1. Choose background color of the tank in **Color**.
2. Choose filling color of the tank in **Fill color**.
3. Enter text in the **Text** field.



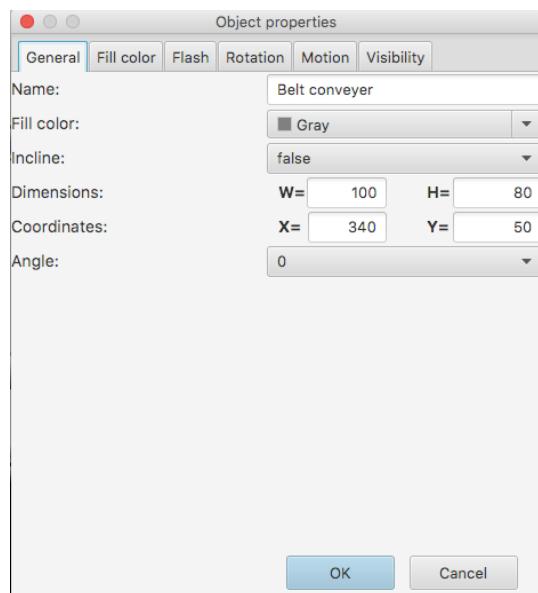
Conveyers library

Conveyers library contains the following objects: Screw conveyer and Belt conveyer. All conveyers have the same General group properties. Below we describe it only for one graphical object - **Belt conveyer**.



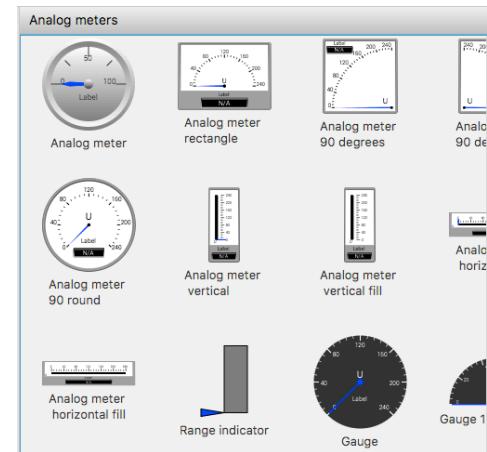
Belt conveyer

1. Choose **Fill color** of the conveyer.
2. Select incline or not in **Incline** combobox.



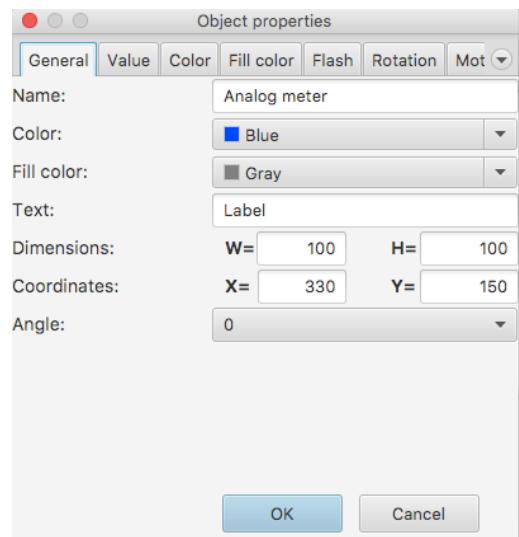
Analog meters library

Analog meters library contains the following objects:
Analog meter, Analog meter rectangle, Analog meter 90 degrees, Analog meter 90 degrees 2, Analog meter 90 round, Analog meter vertical, Analog meter vertical fill, Analog meter horizontal, Analog meter horizontal fill, Range Indicator, Gauge, Gauge 180 degrees, Gauge 90 degrees and Gauge 90 degrees 2.



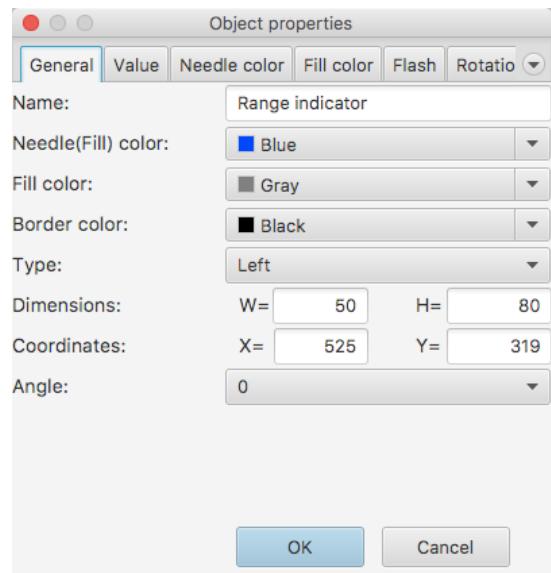
Analog meter

1. Write text of the label in the **Text** field.
2. Choose color of the arrow in the **Color**.
3. Choose filling color of the meter in **Fill color**.



Range Indicator

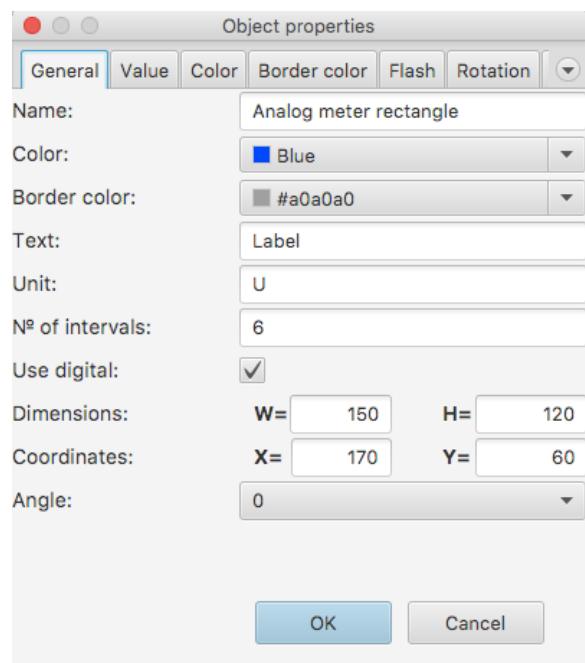
1. Choose **Needle(Fill) color**.
2. Choose **Fill color**.
3. Choose **Border color**.
4. Choose type of the indicator: Left or Right.



Other analog meters

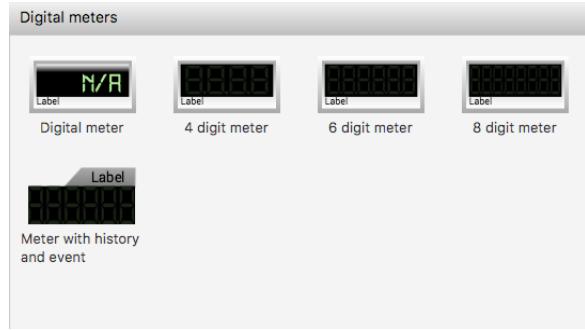
Analog meter rectangle, Analog meter 90 degrees, Analog meter 90 degrees 2, Analog meter 90 round, Analog meter vertical, Analog meter vertical fill, Analog meter horizontal and Analog meter horizontal fill have the same General properties:

1. Write text of the label in the **Text** field.
2. Choose color of the arrow in the **Color**.
3. Choose border color of the meter in **Border color**.
4. Write unit text in the **Unit** field.
5. Enter the number of intervals of the meter in the **Nº of intervals** field.
6. Check **Use digital** if you want to use digital meter.



Digital meters library

Digital meters library contains the following objects: Digital meter, 4 digit meter, 6 digit meter, 8 digit meter and Meter with history and event.

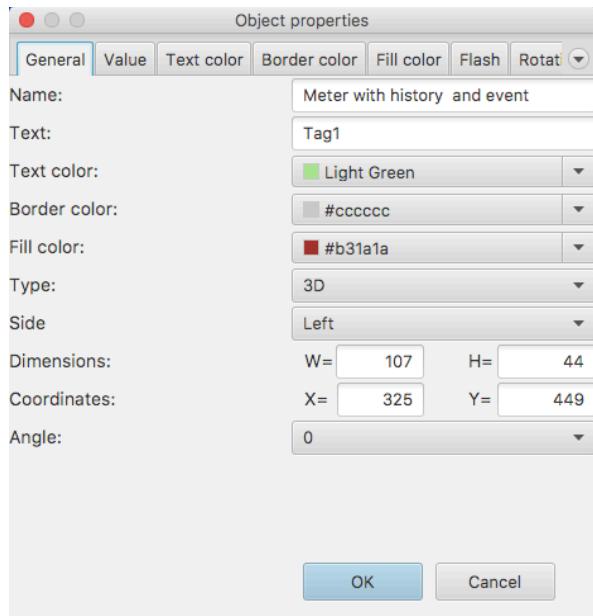


Digital meters

All digital meters have the same general properties:

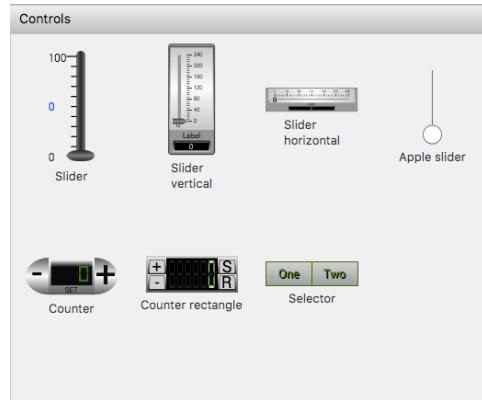
1. Write text of the label in the **Text** field.
2. Choose **Text color**.
3. Choose color of the border in the **Border color**.
4. Choose filling color of the meter in **Fill color**.

For Meter with history and event also you can choose **Side** where history trend or event table will appear after clicking on meter.



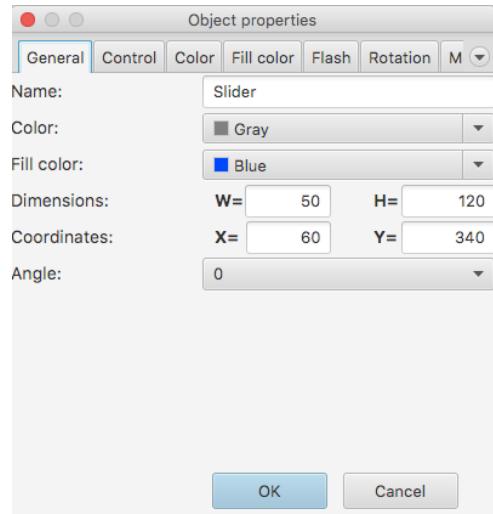
Controls library

Controls library contains the following objects: Slider, Slider vertical, Slider horizontal, Counter, Counter rectangle, Apple slider and Selector.



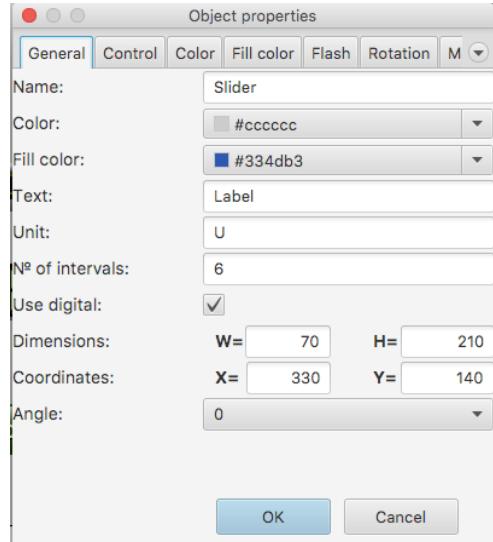
Slider

1. Choose color of the background in the **Color**.
2. Choose filling color of the slider in **Fill color**.



Slider vertical and horizontal

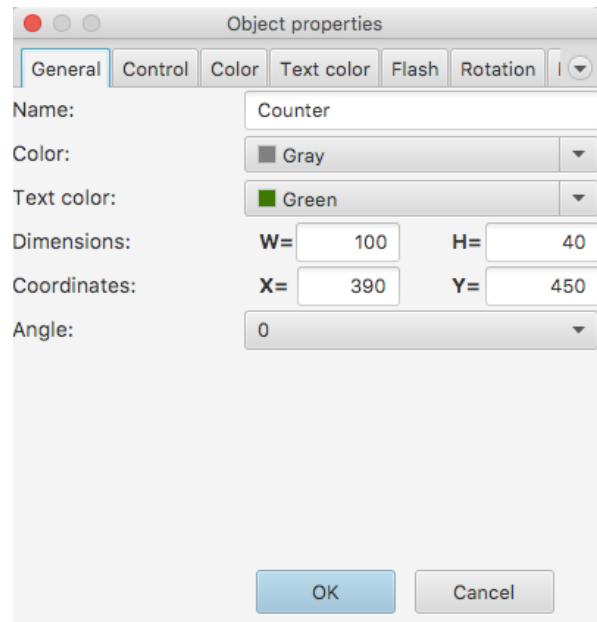
1. Choose color of the background in the **Color**.
2. Choose filling color of the slider in **Fill color**.
3. Enter label text in the **Text** field.
4. Write unit text in the **Unit** field.
5. Enter the number of intervals of the slider in the **No of intervals** field.
6. Check **Use digital** if you want to use digital meter.



Counter and counter rectangle

General properties for counter and counter rectangle are the same:

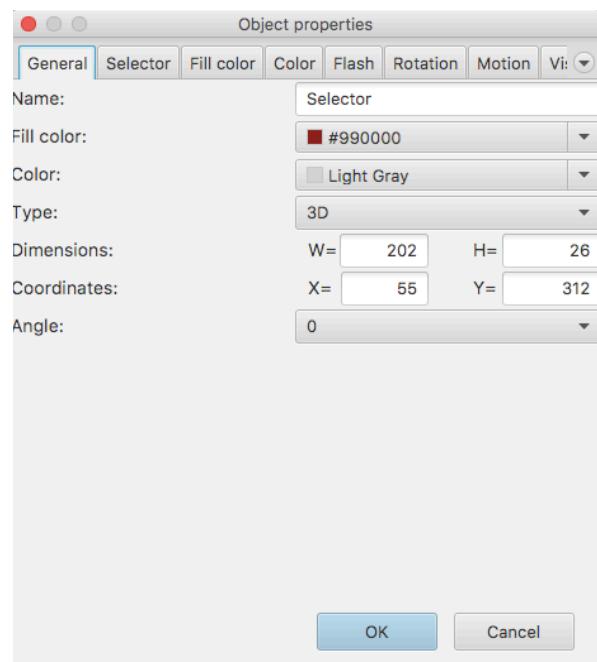
1. Choose color of the background in the **Color**.
2. Choose **Text color**.



Selector

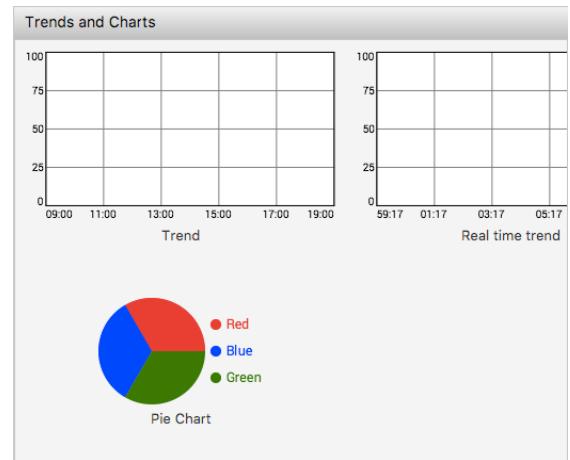
General properties for the selector:

1. Choose colour of the background in the **Color**.
2. Choose **Fill color** for the background of the selected button.



Trends and Charts library

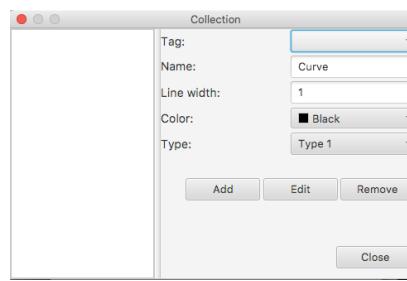
Trends library contains the following objects: Trend, Real time trend, Trend DB, Trend OPC UA History and Pie Chart. *Trend* and *Real time trend* draw curves based on tags that used history data collection (check **Enable history** in Tags properties). *Trend DB* draw curves based on tags that used data stored in database (check **Store in DB** in Tags properties). *Trend OPC UA History* draw curves based on tags that bind to OPC UA nodes supported Historyzing property. All trends have the same General and Grid group properties. Below we describe their only for one graphical object - **Trend**.



Trend

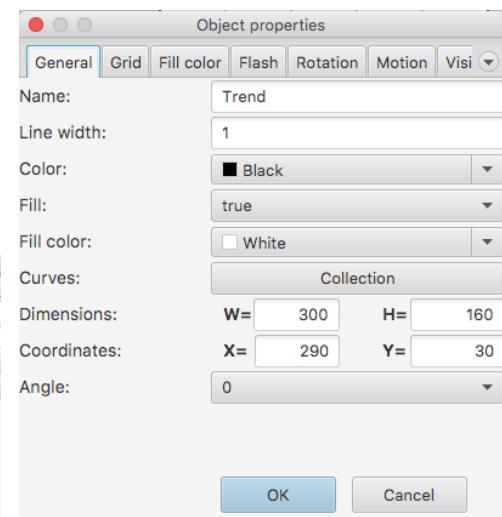
1. Enter width of line in the **Line width**.
2. Choose background color of the trend in **Color**.
3. Select fill or not in the **Fill** combobox.
4. Choose filling color of the trend in **Fill color**.
5. To add curve click

Collection button.



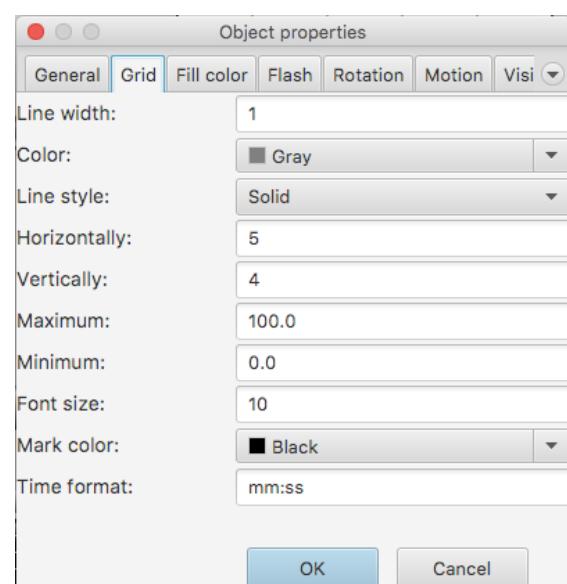
Collection window will appear:

1. Select tag that you want to bind to this curve in the **Tag**.
2. Enter curve's name in the **Name** field.
3. Write width of curve's line in **Line width** field.
4. Choose **Color** of the curve.
5. Select **Type** of the curve. *Type 1* just draw the line. *Type 2* draw line with filling till axis X.



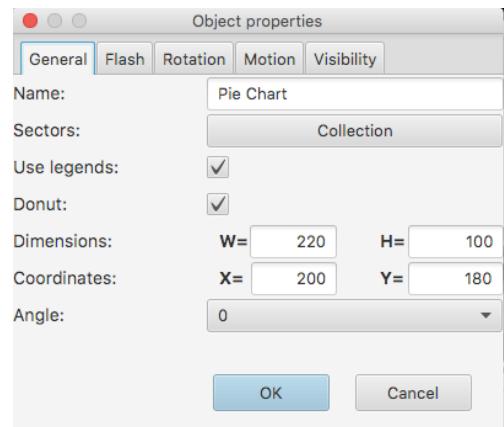
On the **Grid** tab:

1. Enter width of grid's lines in the **Line width** field.
2. Choose **Color** of grid's lines.
3. Select **Line style**: *Solid*, *Dash*, *Dot* or *DashDot*.
4. Write number of horizontal grid's lines in **Horizontally** field.
5. Write number of vertical grid's lines in **Vertically** field.
6. Enter **Maximum** value of the trend.
7. Enter **Minimum** value of the trend.
8. Write **Font size** of the marks.
9. Choose **Mark color**.
10. Enter **Time format** of trend's time axis.



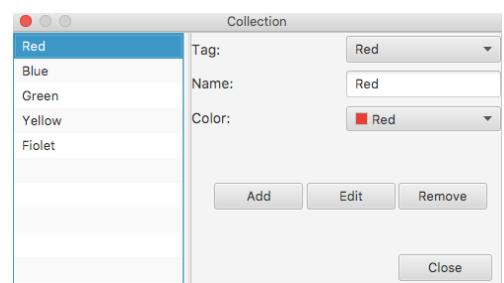
Pie Chart

1. Check **Use legends** if you want to add legends to the chart.
2. Check **Donut** if you want to use ring type chart.
3. To add, edit and remove sector(s) click **Collection**.



Collection window will appear:

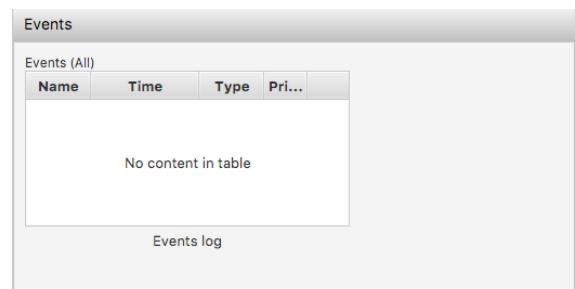
1. Select tag that you want to bind to this sector in the **Tag**.
2. Enter sector's name in the **Name** field.
3. Choose **Color** of the sector.



Events library

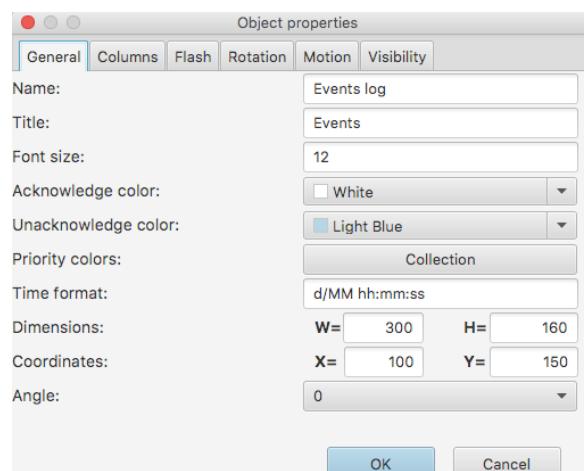
Events library contains the following object: Events log.

Events log collects tag's events (check **Enable alarms** and check events you want to collect in Tags properties).



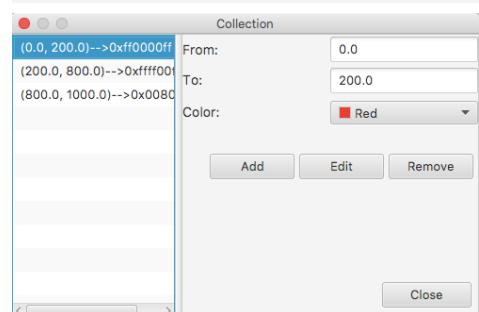
Events log

1. Enter title of the table in the **Title** field.
2. Write size of text in the **Font size**.
3. Choose row's background color of acknowledged events in the **Acknowledge color**.
4. Choose row's background color of not acknowledged events in the **Unacknowledge color**.
5. Choose color of the event's text by clicking **Collection** button.
6. Enter **Time format** of the time's text.



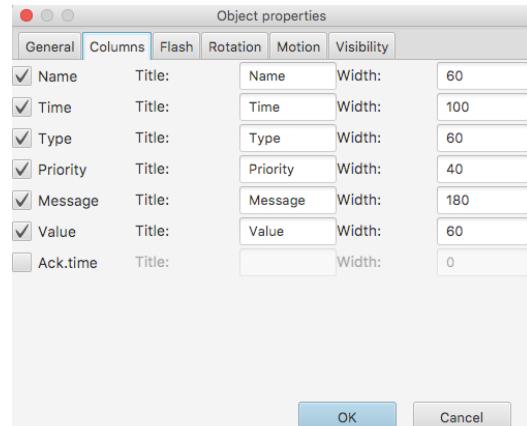
After clicking **Collection** button you'll see the window:

1. Enter the priority of the event from which be used this color in the **From** field.
2. Enter the priority of the event to which be used this color in the **To** field.
3. Choose **Color** of the event's text.



On the *Columns* tab:

1. Select columns that you want to use in the table.
2. Write titles of the columns in correspondent **Title** field.
3. Enter **Width** of the correspondent column.



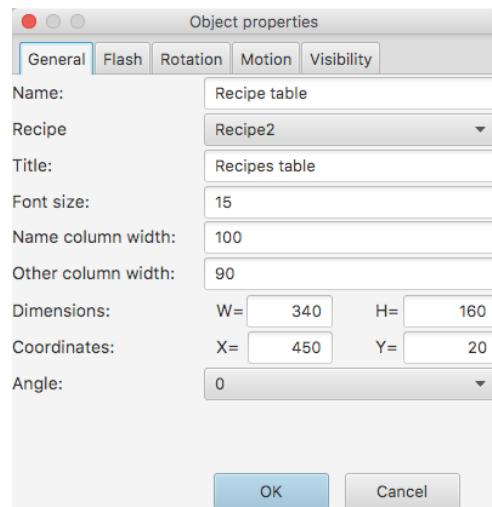
Recipes library

Recipes library contains the following objects: Recipe table and Recipe selector.



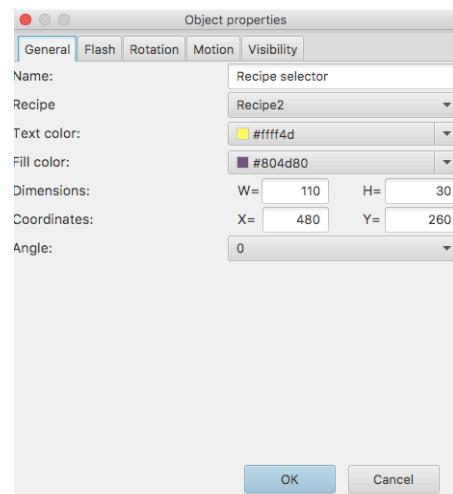
Recipe table

1. Choose **Recipe** you want to bind to the table. During running you can *add*, *edit* and *delete* fields of the recipe database by clicking right button on the table and choosing operation.
2. Enter title of the table in the **Title** field.
3. Write size of text in the **Font size**.
4. Enter **Name column width**. Set width of the name's column.
5. Enter **Other column width**. Set width of other columns.



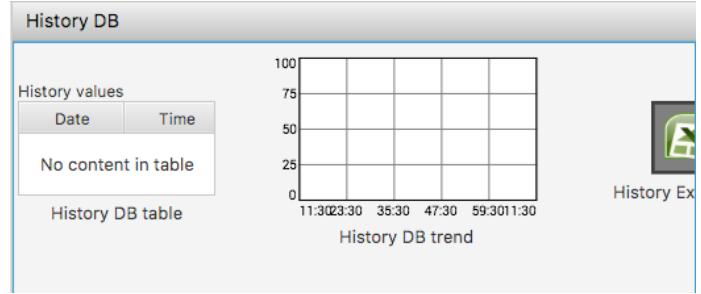
Recipe selector

1. Choose **Recipe** you want to bind to the selector. During running you can *select* fields of the recipe database by clicking on the recipe selector.
2. Choose **Text color** of the selector.
3. Choose **Fill color** of the selector.



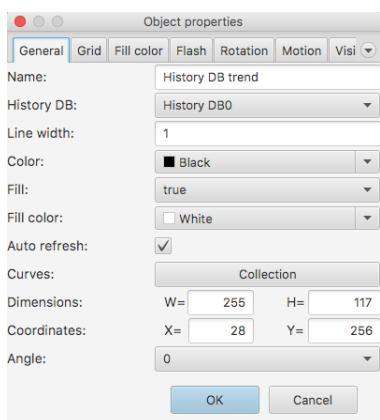
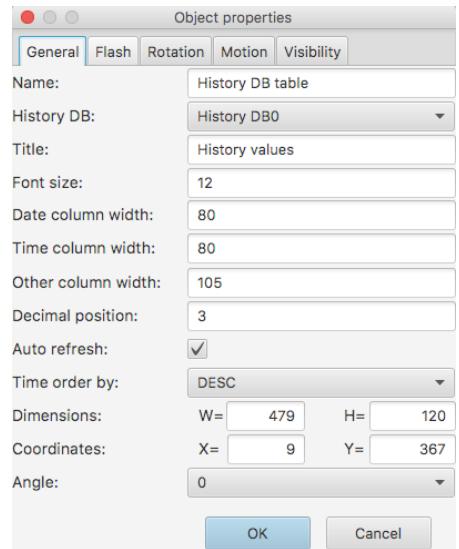
History DB library

History DB library contains the following objects: History DB table, History DB trend, History Excel Report and History Max and Min Report.



History DB table

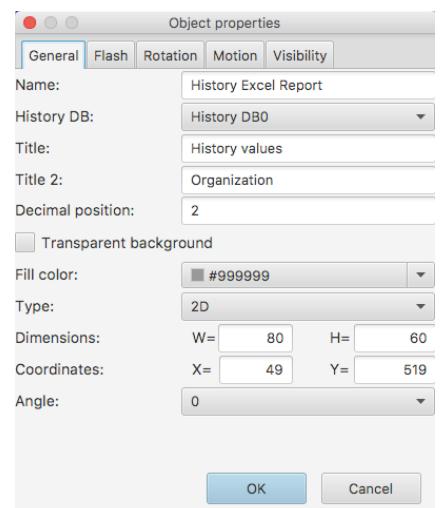
1. Choose **History DB** you want to bind to the table.
2. Enter title of the table in the **Title** field.
3. Write size of text in the **Font size**.
4. Enter **Date column width**. Set width of the date's column.
5. Enter **Time column width**. Set width of the time's column.
6. Enter **Other column width**. Set width of other columns.
7. Enter **Decimal position** of tag's values entered in the table.
8. Check **Auto refresh** if you want update table every time when new tag's value added into database.
9. Choose **Time order by** of the database rows ASC or DESC.



History DB trend

All properties of History DB trend are similar to properties of Trend object. And additional properties:

1. Check **Auto refresh** if you want update table every time when new tag's value added into database..
2. Choose **History DB** you want to bind to the trend.

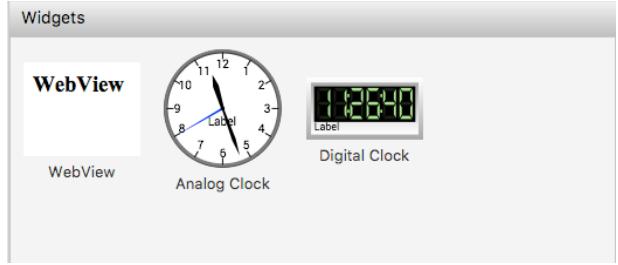


History Excel report and History Max and Min report

1. Choose **History DB** you want to bind to the report.
2. Enter title of the table of the report in the **Title** field.
3. Enter second title of the table in the **Title 2** field.
4. Enter **Decimal position** of tag's values entered in the table.
5. Check **Transparent background** if you want to make background of the button invisible.
6. Choose **Fill colour** of the button.
7. Choose **Type** of the button.

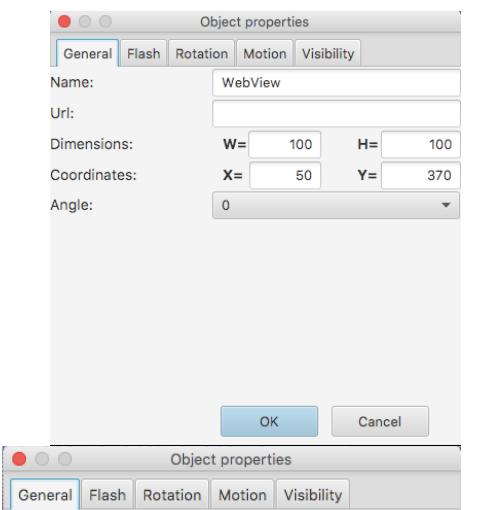
Widgets library

Widgets library contains the following object:
WebView, Analog Clock and Digital Clock.



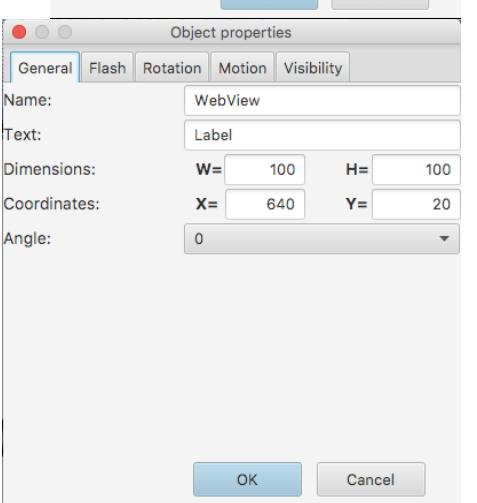
Web View

1. Enter url of the internet resource in the **Url** field.



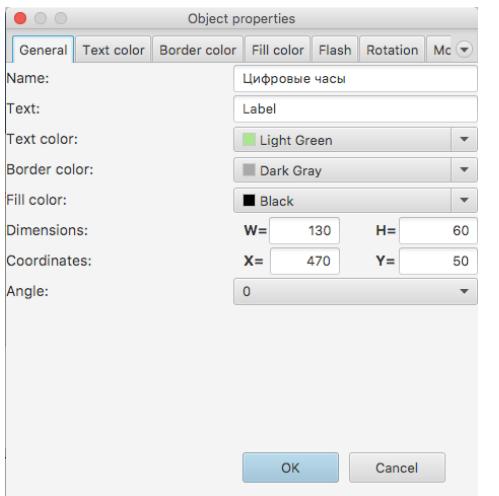
Analog clock

1. Enter label of the clock in the **Text** field.



Digital clock

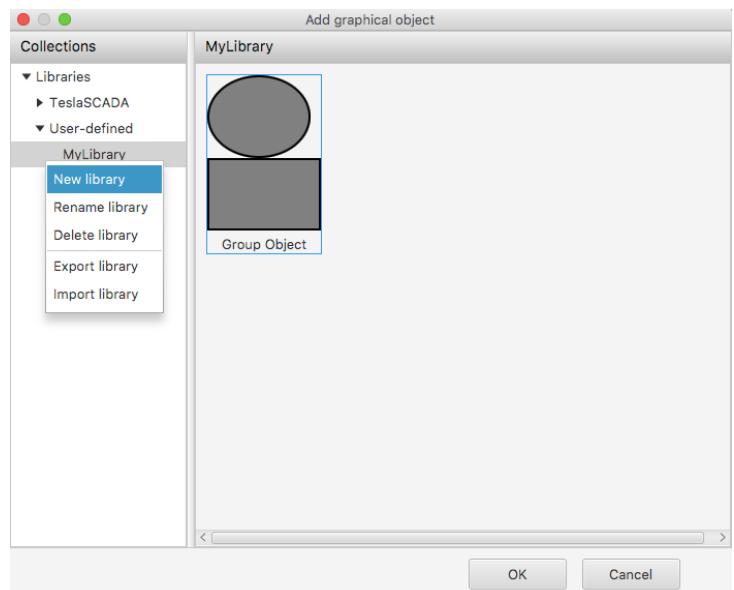
1. Write text of the label in the **Text** field.
2. Choose **Text color**.
3. Choose color of the border in the **Border color**.
4. Choose filling color of the meter in **Fill color**.



User-defined library

Create user-defined library

You can create your own library by clicking right button on *Collections* section of the **Add graphical object** window and choosing *New library* menu item. You can add graphical object in your library by clicking right button on the object on **Canvas** or **Screen window** and choosing *Add to Library->Your library* menu item. You can *Select, Rename or Delete* created object in your library by clicking right button on it and selecting correspondent menu item.



Rename user-defined library

To rename library:

1. Right click on the library you want to rename and choose *Rename library* item.

Delete user-defined library

To delete library:

1. Right click on the library you want to delete and choose *Delete library* item.

Export user-defined library

To export library:

1. Right click on the library you want to export and choose *Export library* item.

2. Now select the location and click the button *Save* (TeslaSCADA library extension .tsp2lib).

Import user-defined library

To import library:

1. Right click on the Collections window and choose *Import library* item.

2. Now select the library file and click *Open* (TeslaSCADA library extension .tsp2lib).

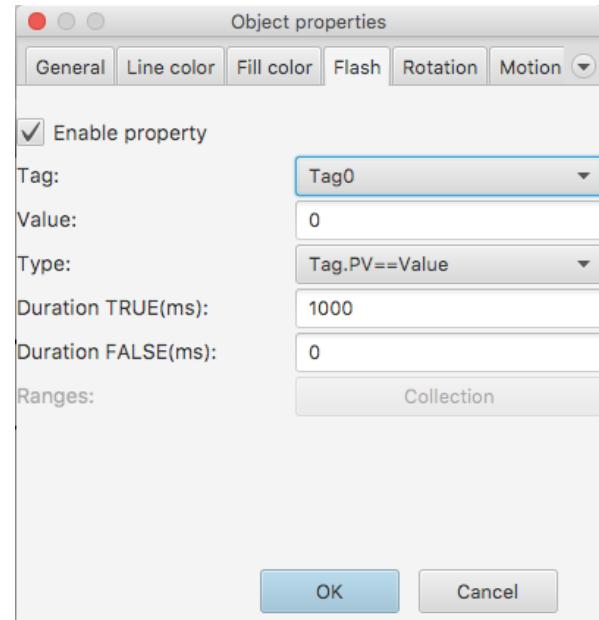
Properties

Each graphical object has several group of properties. To use property of the graphical object check **Enable Property**. Each object has *Flash*, *Rotation*, *Motion* and *Visibility* properties. Other properties depend on the object.

Flash

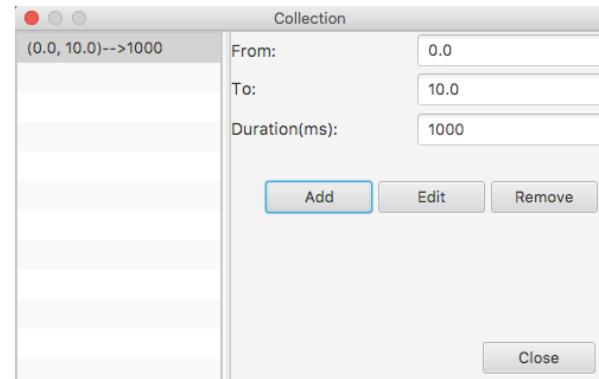
Flash property lets object to flash when conditions is TRUE or FALSE. To edit flash property click **Flash** tab on the object property window.

1. Select the **Tag** value of which will be compared.
2. Enter the comparison **Value**.
3. Select **Type** of comparison.
4. Write period's time in milliseconds of objects flashing if the comparison is true in the **Duration TRUE(ms)** field. If you enter 0 the object will not flashing.
5. Write period's time in milliseconds of objects flashing if the comparison is false in the **Duration FALSE(ms)** field. If you enter 0 the object will not flashing.
6. If you select *Tag.PV in the range* in the **Type** combobox and click **Collection** button. You'll see the window:



1. Enter the value from which the object will flash with this periodicity in the **From** field.
2. Enter the value to which the object will flash with this periodicity in the **To** field.
3. Enter period of flashing in the **Duration(ms)** field.

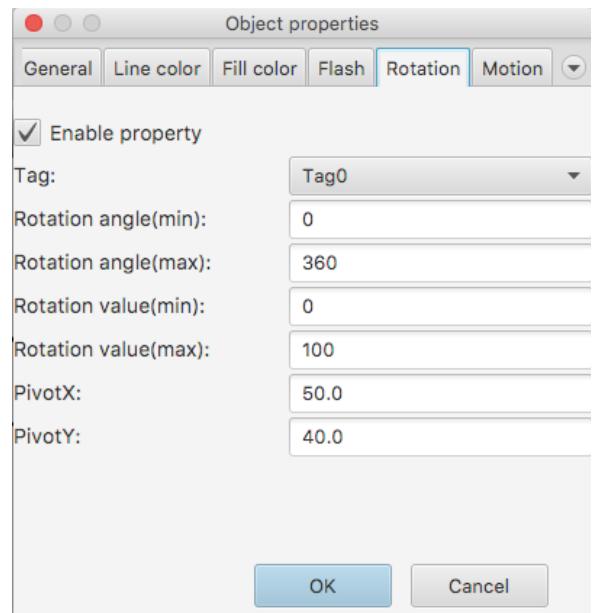
You can *Add*, *Edit* or *Remove* collection element of flashing conditions.



Rotation

Rotation property lets to rotate the object proportional to the value of the tag. To edit rotation property click **Rotation** tab on the object property window.

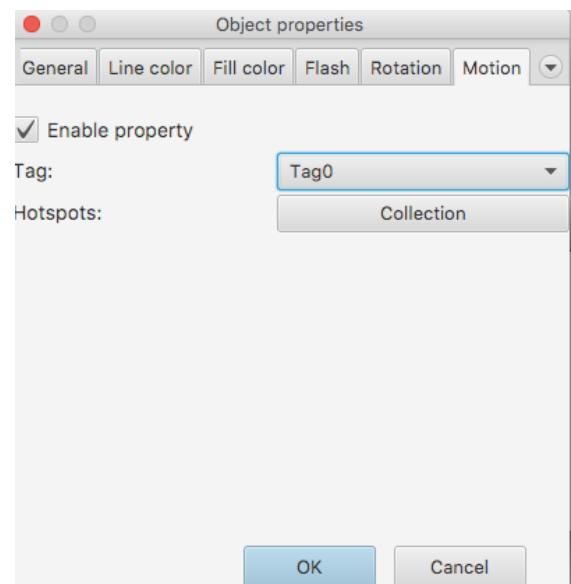
1. Select the **Tag** value of which will be compared.
2. Enter the minimum of rotation angle in the **Rotation angle(min)** field.
3. Enter the maximum of rotation angle in the **Rotation angle(max)** field.
4. Write the minimum of the tag's value in the **Rotation value(min)**.
5. Write the maximum of the tag's value in the **Rotation value(max)**.
6. Enter X coordinate of the pivot in **PivotX** field.
7. Enter Y coordinate of the pivot in **PivotY** field.



Motion

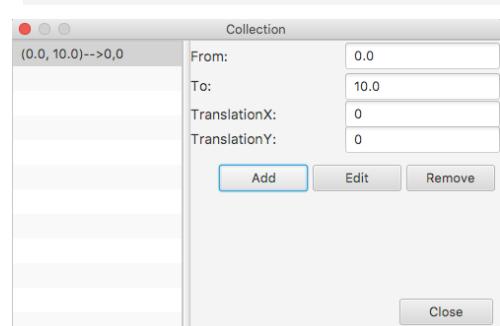
Motion property lets to move the object depending on value of the tag. To edit motion property click **Motion** tab on the object property window.

1. Select the **Tag** depending on value of which the object will change location coordinates.
2. Click Collection button to edit move conditions coordinates.



After clicking you'll see window:

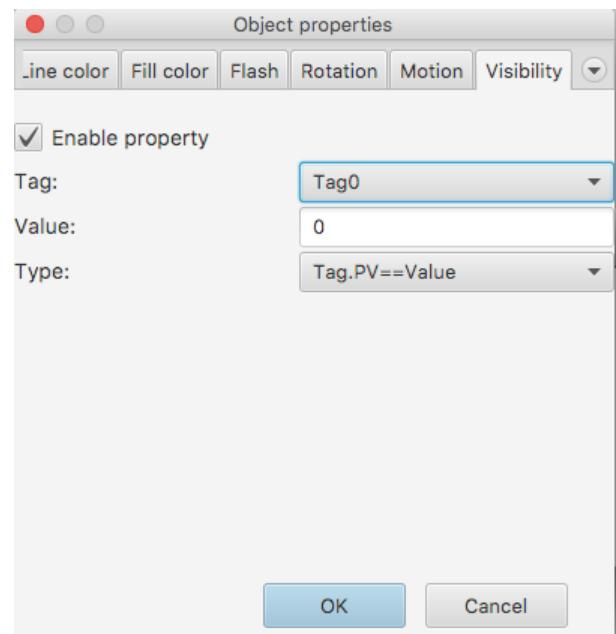
1. Enter the value from which the object will change coordinates in the **From** field.
2. Enter the value to which the object will change coordinates in the **To** field.
3. Write **TranslationX** coordinates.
4. Write **TranslationY** coordinates.



Visibility

Visibility property lets to make the object visible or not depending on the tag's value. To edit visibility property click **Visibility** tab on the object property window.

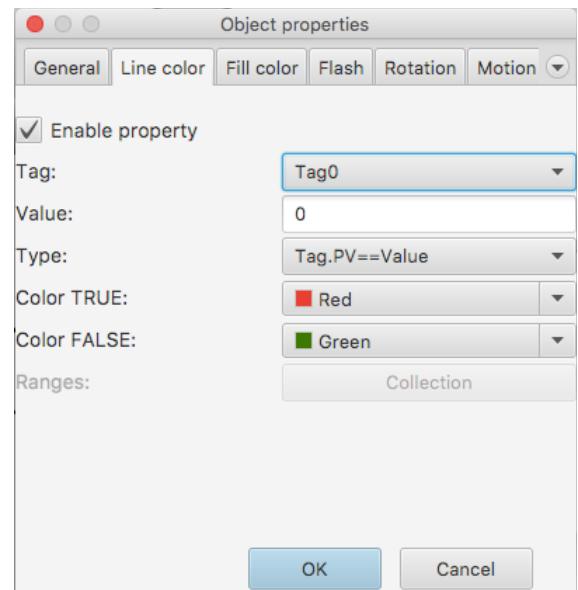
1. Select the **Tag** value of which will be compared.
2. Enter the comparison **Value**.
3. Select **Type** of comparison.



Line color

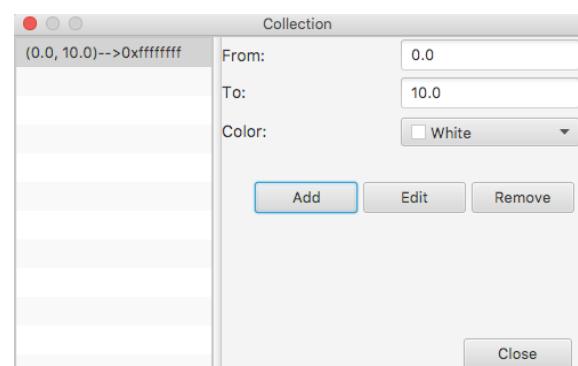
Line color property lets object to change color of its line when conditions is TRUE or FALSE. To edit line color property click **Line color** tab on the object property window.

1. Select the **Tag** value of which will be compared.
2. Enter the comparison **Value**.
3. Select **Type** of comparison.
4. Choose a color that will result if the comparison is true in **Color TRUE**.
5. Choose a color that will result if the comparison is false in **Color FALSE**.
6. If you select *Tag.PV in the range* in the **Type** combobox and click **Collection** button. You'll see the window:



1. Enter the value from which the object will change color in the **From** field.
2. Enter the value to which the object will change color in the **To** field.
3. Choose **Color**.

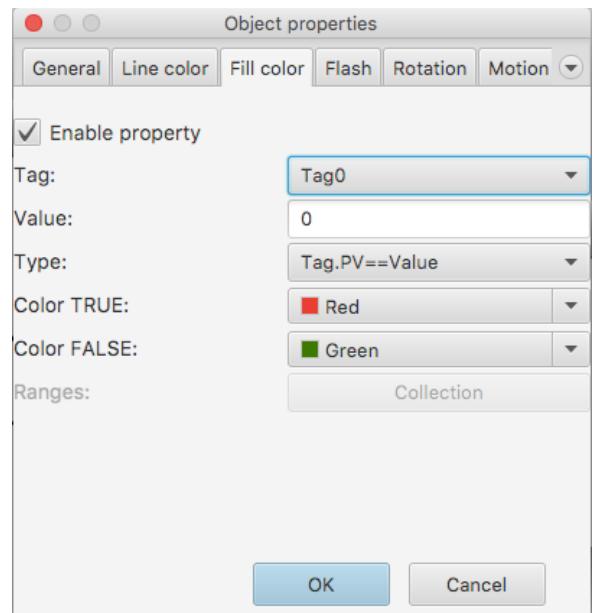
You can *Add*, *Edit* or *Remove* collection element of line color conditions.



Fill color

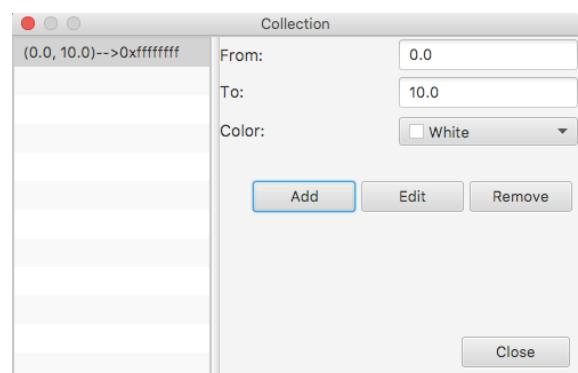
Fill color property lets object to change color of its filling when conditions is TRUE or FALSE. To edit fill color property click **Fill color** tab on the object property window.

1. Select the **Tag** value of which will be compared.
2. Enter the comparison **Value**.
3. Select **Type** of comparison.
4. Choose a color that will result if the comparison is true in **Color TRUE**.
5. Choose a color that will result if the comparison is false in **Color FALSE**.
6. If you select *Tag.PV in the range* in the **Type** combobox and click **Collection** button. You'll see the window:



1. Enter the value from which the object will change color in the **From** field.
2. Enter the value to which the object will change color in the **To** field.
3. Choose **Color**.

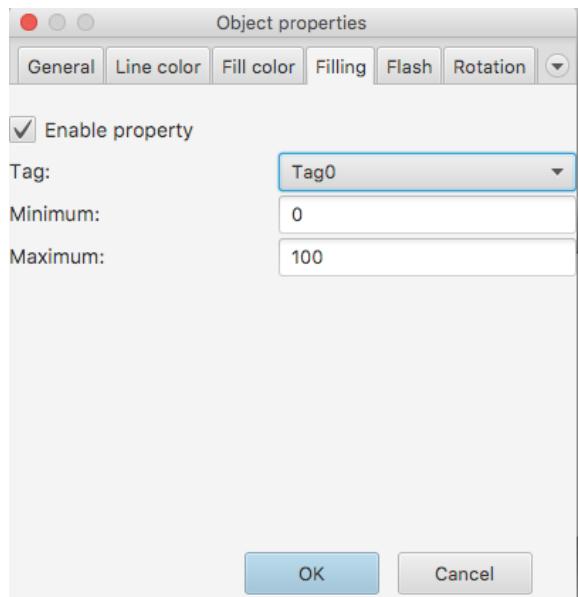
You can *Add*, *Edit* or *Remove* collection element of fill color conditions.



Filling

Filling property lets to control filling of the object depending on value of the tag. To edit filling property click **Filling** tab on the object property window.

1. Select the **Tag** value of which will be used to control filling.
2. Enter minimum value of the object's filling in the **Minimum** field.
3. Enter maximum value of the object's filling in the **Maximum** field.



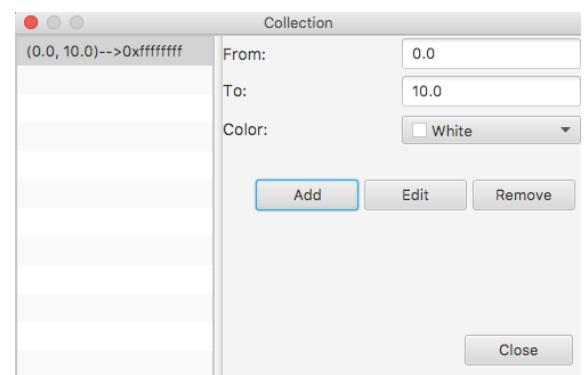
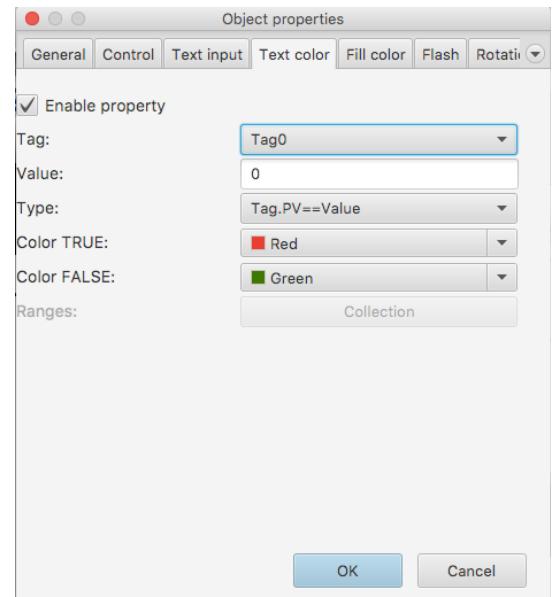
Text color

Text color property lets object to change color of text when conditions is TRUE or FALSE. To edit text color property click **Text color** tab on the object property window.

1. Select the **Tag** value of which will be compared.
2. Enter the comparison **Value**.
3. Select **Type** of comparison.
4. Choose a color that will result if the comparison is true in **Color TRUE**.
5. Choose a color that will result if the comparison is false in **Color FALSE**.
6. If you select *Tag.PV in the range* in the **Type** combobox and click **Collection** button. You'll see the window:

1. Enter the value from which the object will change color in the **From** field.
2. Enter the value to which the object will change color in the **To** field.
3. Choose **Color**.

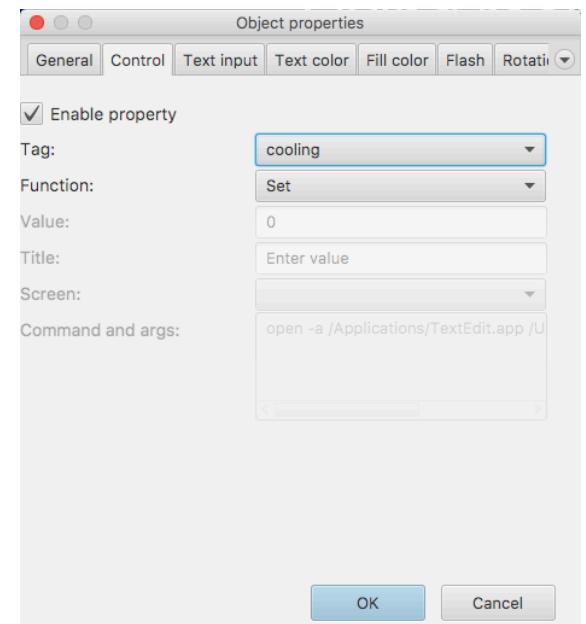
You can *Add*, *Edit* or *Remove* collection element of text color conditions.



Control (for buttons)

Control property lets to write value to the tag. To edit control property click **Control** tab on the object property window.

1. Select **Tag** which will be recorded value.
2. Select **Function** of writing value: *Set* - will write true to the tag; *Reset* - will write false to the tag; *Toggle* - if current tag's value true will write false, if currents tag's value false will write true; *Push* - during pressing button will write true; *Set value* - will write Value to the tag; *Enter value* - will call dialog that lets you enter value; *Call screen* - will call selected screen; *Call popup* - will call selected popup screen, *Close popup* - will close popup screen, *Call external software* - lets call external software by using command and arguments of OS.
3. When you select *Set value* **Function** write **Value** that will be written to the tag.
4. When you select *Enter value* **Function** write **Title** of the called dialog that lets you enter value.



5. When you select *Call screen* or *Call popup Function* choose **Screen** that will be called after clicking button.

6. Enter **Command and args** of the OS for calling external software. Examples:

for MacOS: *open /Applications/TextEdit.app*

for Windows: *C:/Progra~1/somesoftware.exe*

for Android: *ops.tesla.scada* (*name of the Android application package*)

for iOS: http://www.youtube.com/watch?v=VIDEO_IDENTIFIER (*youtube scheme for calling in iOS*).

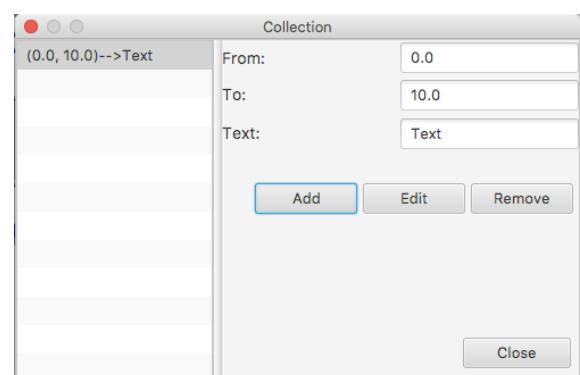
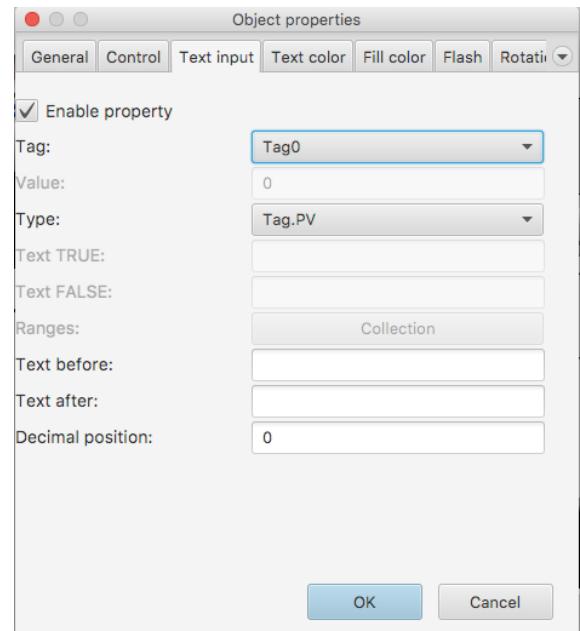
Text input

Text input property lets object to control display tag's value or text when conditions is TRUE or FALSE. To edit text input property click **Text input** tab on the object property window.

1. Select the **Tag** value of which will be compared.
2. Enter the comparison **Value**.
3. Select **Type** of comparison or display tag's value.
4. Enter text that will result if the comparison is true in **Text TRUE**.
5. Enter text that will result if the comparison is false in **Text FALSE**.
6. Write the text will be shown before displayed text in the **Text before**.
7. Write the text will be shown after displayed text in the **Text after**.
8. Enter **Decimal position** of displayed text in the field.
9. If you select *Tag.PV in the range* in the **Type** combobox and click **Collection** button. You'll see the window:

1. Enter the value from which the object will change text in the **From** field.
2. Enter the value to which the object will change text in the **To** field.
3. Write displayed **Text**.

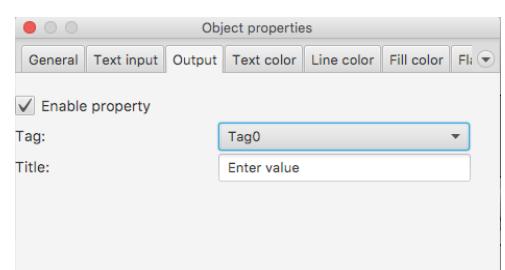
You can *Add*, *Edit* or *Remove* collection element of displayed text conditions.



Output

Output property lets to write value to the tag. To edit output property click **Output** tab on the object property window.

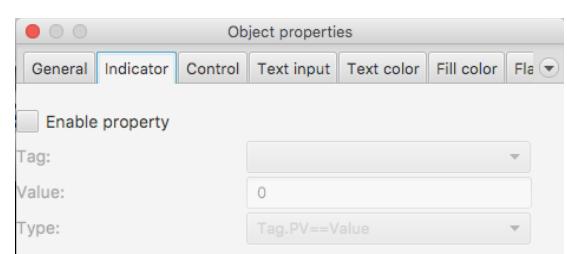
1. Select the **Tag** where value will be written.
2. Enter **Title** of the dialog that will be used to write value to the tag.



Indicator

Indicator property lets to indicate object depending on value of the tag. To edit indicator property click **Indicator** tab on the object property window.

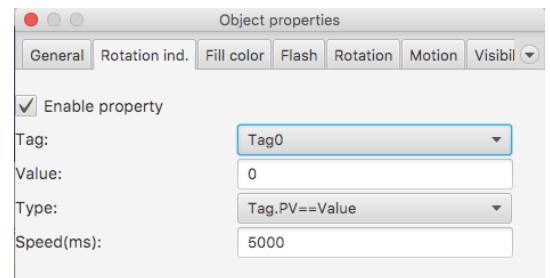
1. Select the **Tag** value of which will be compared.
2. Enter the comparison **Value**.
3. Select **Type** of comparison.



Rotation Indicator

Rotation Indicator property lets to rotate object around its center depending on value of the tag. To edit indicator property click **Rotation ind.** tab on the object property window.

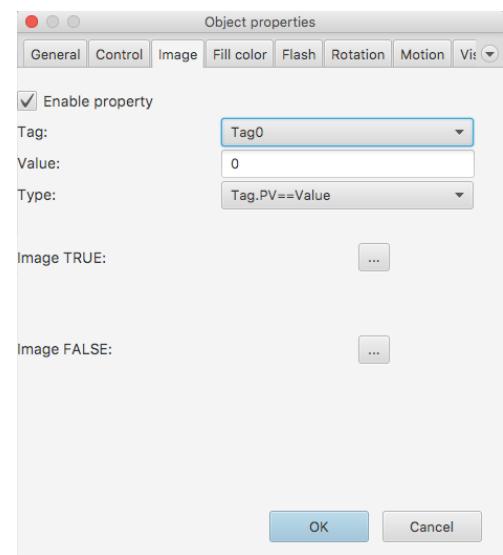
1. Select the **Tag** value of which will be compared.
2. Enter the comparison **Value**.
3. Select **Type** of comparison.
4. Enter rotation speed in **Speed(ms)**.



Image

Image property lets object to change image when conditions is TRUE or FALSE. To edit image property click **Image** tab on the object property window.

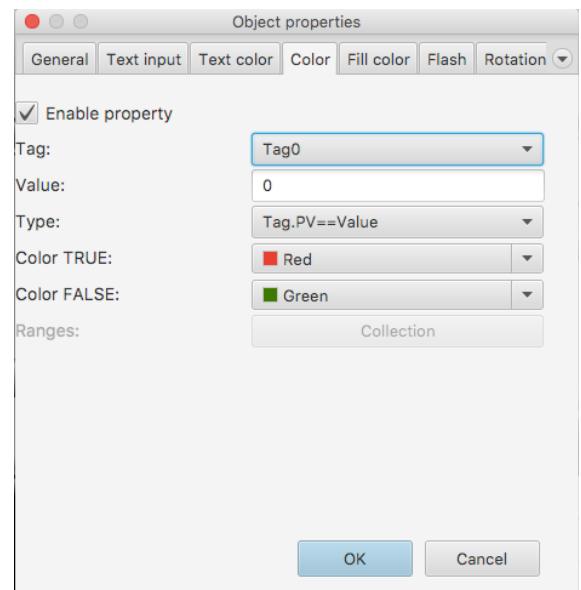
1. Select the **Tag** value of which will be compared.
2. Enter the comparison **Value**.
3. Select **Type** of comparison.
4. Choose image that will result if the comparison is true by clicking **Image TRUE** button.
5. Choose image that will result if the comparison is false by clicking **Image FALSE** button.



Color

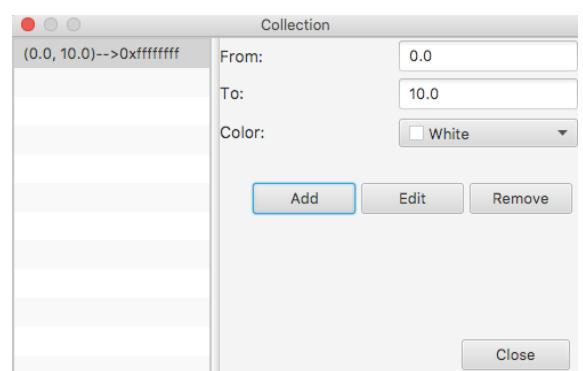
Color property lets object to change color of its when conditions is TRUE or FALSE. To edit color property click **Color** tab on the object property window.

1. Select the **Tag** value of which will be compared.
2. Enter the comparison **Value**.
3. Select **Type** of comparison.
4. Choose a color that will result if the comparison is true in **Color TRUE**.
5. Choose a color that will result if the comparison is false in **Color FALSE**.
6. If you select *Tag.PV in the range* in the **Type** combobox and click **Collection** button. You'll see the window:



1. Enter the value from which the object will change color in the **From** field.
2. Enter the value to which the object will change color in the **To** field.
3. Choose **Color**.

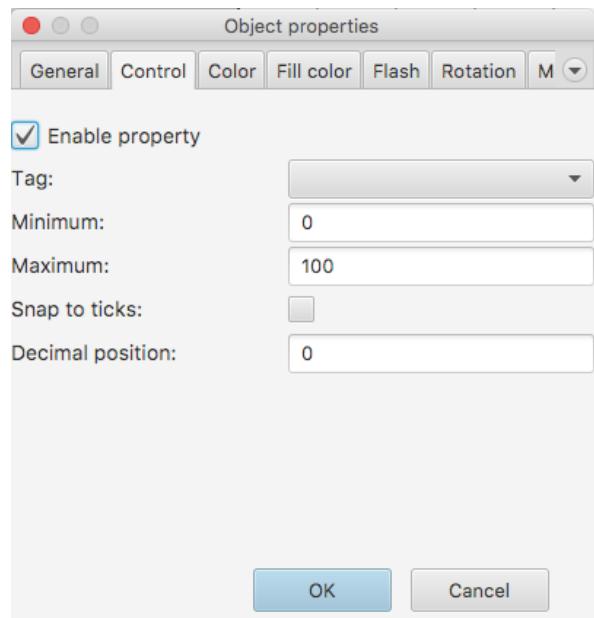
You can *Add*, *Edit* or *Remove* collection element of color conditions.



Control (slider)

Control property lets object to write value to the tag. To edit control property click **Control** tab on the object property window.

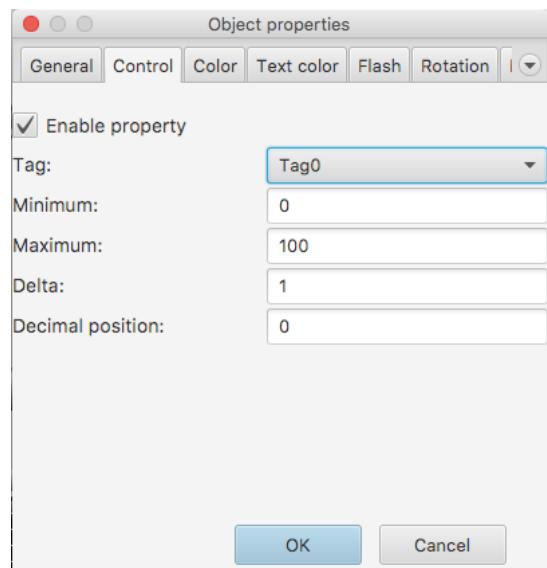
1. Select the **Tag** value of which will be changed.
2. Enter **Minimum** value of the control.
3. Enter **Maximum** value of the control.
4. Check **Snap to ticks** if you want to bind control's value.
5. Enter **Decimal position** of displayed text in the field.



Control (counter)

Control property lets object to write value to the tag. To edit control property click **Control** tab on the object property window.

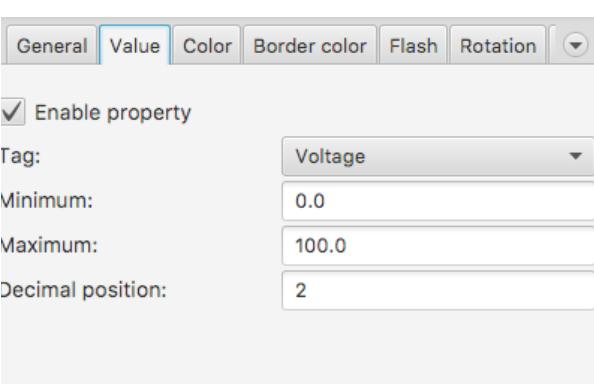
1. Select the **Tag** value of which will be changed.
2. Enter **Minimum** value of the control.
3. Enter **Maximum** value of the control.
4. Write **Delta** in the field.
5. Enter **Decimal position** of displayed text in the field.



Value (for meters)

Value property lets to control values of analog and digital meters depending on value of the tag. To edit value property click **Value** tab on the object property window.

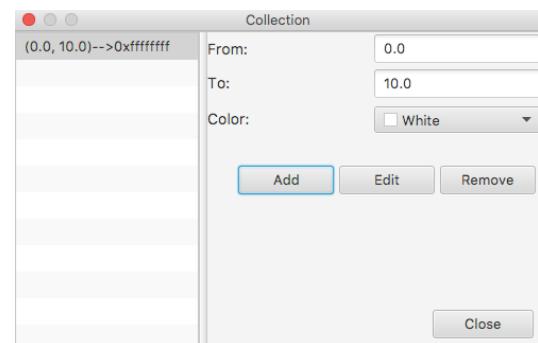
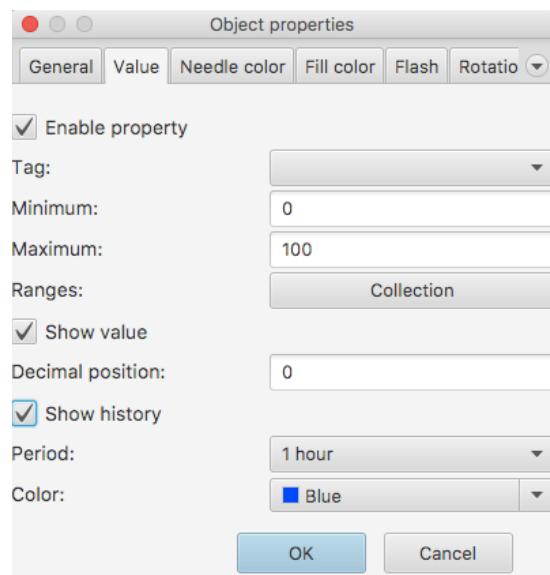
1. Select the **Tag** value of which will be used to control value of meter.
2. Enter minimum value of the meter in the **Minimum** field.
3. Enter maximum value of the meter in the **Maximum** field.
4. Enter **Decimal position** of displayed text in the field.



Value (for Range indicator and Gauges)

Value property lets to indicate depending on value of the tag. To edit value property click **Value** tab on the object property window.

1. Select the **Tag** value of which will be used to control value of the indicator.
2. Enter minimum value of the indicator in the **Minimum** field.
3. Enter maximum value of the indicator in the **Maximum** field.
4. Enter **Decimal position** of displayed text in the field.
5. Check **Show value** if you want to make visible number representation.
6. Check **Show history** if you want to make visible history information of the tag.
7. Choose **Period** of the history information.
8. Choose **Color** of the history information.
9. Click **Collection** of the **Ranges** if you want to add Color ranges of the indicator. You'll see the window:



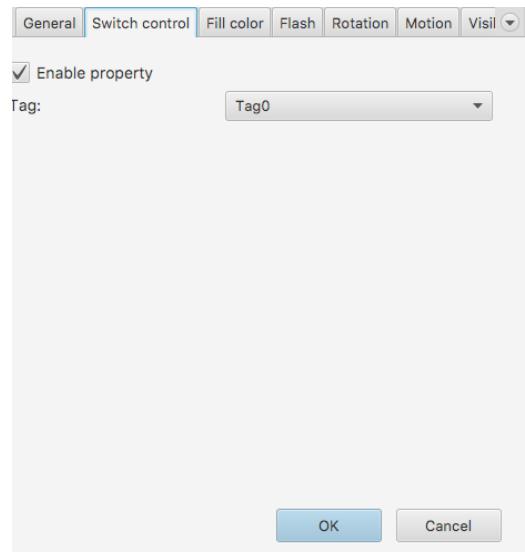
1. Enter the value from which the object will change color in the **From** field.
2. Enter the value to which the object will change color in the **To** field.
3. Choose **Color**.

You can *Add*, *Edit* or *Remove* collection element of line color conditions.

Switch control

Switch control property lets to switch boolean value of the tag. To edit switch control property click **Switch control** tab on the object property window.

1. Select the **Tag** value of which will be controlled by the switch.



Switch control (3 pos.)

Switch control property lets to switch boolean values of the tags. To edit switch control property click **Switch control** tab on the object property window.

1. Select the **Tag ON** value of which will be controlled by the switch.
2. Select the **Tag OFF** value of which will be controlled by the switch.

When the value of **Tag ON** is **true** and the value of **Tag OFF** is **false** the switch position will be ON.

When the value of **Tag ON** is **false** and the value of **Tag OFF** is **true** the switch position will be OFF.

In other situations the switch position will be **Neutral**.

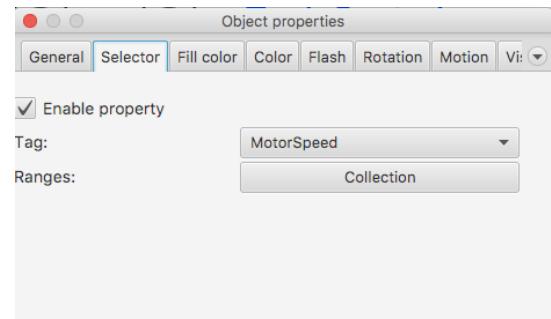
To switch click (or touch on mobile devices) on the position you want.



Selector

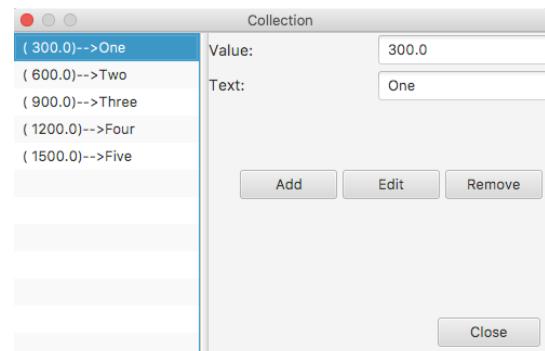
Selector property lets to enter values by clicking selector buttons. To edit selector property click **Selector** tab on the object property window.

1. Select the **Tag** which value will be changed.
2. Click Collection button to edit selector buttons.



After clicking you'll see window:

1. Enter the **Value** which will be written after clicking the button of the selector.
2. Enter **Text** of the selector's button.



Design script

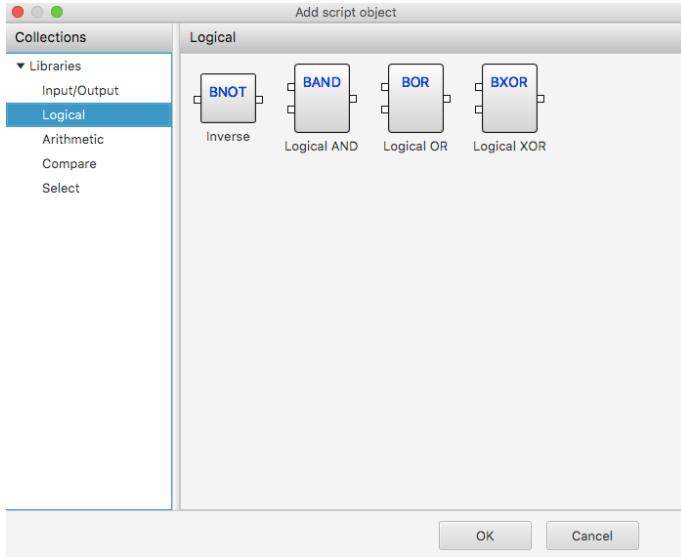
To start designing the script you want, you should double click on it or click right button on the **Project window->Scripts** and choose *Open script*. For creating scripts you should use FBD objects.

Create script object

Add new object on the screen you can in this way: click right button on the **Canvas** and choose *New object* item



You'll see the **Add script object** window:



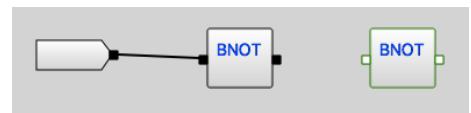
Select library which object you want to use in your project (all libraries and their objects described below). Select object you can in several ways:

- 1.By double clicking on the object.
2. By clicking on the object (select rectangle will appear) and then clicking OK button.
- 3.By clicking right button and choosing *Select* item.

Add script object window will disappear and you can select the location on the screen where you want to place an object.

Connect script objects

To connect two objects, click the end of the first (the end to paint over) and click start the second. This will bring up a line connection.



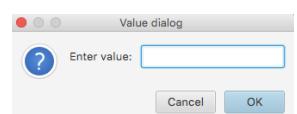
Bind script object to the tag

You can bind Input/Output script objects to the tag. To do this click on Input/Output script object, dialog will appear. Select tag you want to bind.



Enter value to the value script object

You can enter value to value script objects. To do this click on value script object, dialog will appear. Enter value you want to use with this object.



Duplicate script object

You can duplicate script object. Right click on the object you want to duplicate and select *Duplicate* menu item.

Erase script object

You can erase script object. Right click on the object you want to erase and select *Erase* menu item.

Erase connection line

You can erase connection line. Right click on the line you want to erase and select *Erase* menu item.

Script objects

Below description of script libraries and object.

Input/Output library

Input tag - this script object used to bind input tag to the script.

Output tag - this script object used to bind output tag to the script.

Value - this script object used to bind input constant value to the script.

Logical library

Inverse - this script object used to inverse input boolean value (Output = ! Input).

Logical AND - this script object used to logical operation AND for input boolean values (Output = Input & Input2).

Logical OR - this script object used to logical operation OR for input boolean values (Output = Input || Input2).

Logical XOR - this script object used to logical operation XOR for input boolean values (Output = Input XOR Input2).

Bitmap operations library

Inverse - this script object used to inverse input integer value (Output = ~ Input).

Bitmap AND - this script object used to logical operation AND for input integer values (Output = Input & Input2).

Bitmap OR - this script object used to logical operation OR for input integer values (Output = Input || Input2).

Bitmap XOR - this script object used to logical operation XOR for input integer values (Output = Input XOR Input2).

Left Shift - this script object used to left shift bits of input value (Output = Input << № of bits).

Right Shift - this script object used to right shift bits of input value (Output = Input >> № of bits).

Bytes to Short - this script object used to pack 2 bytes in the short (Output = Input<<8+Input2).

Short to Bytes - this script object used to unpack short value in 2 bytes (Output = Input[Input2]).

Shorts to Int - this script object used to pack 2 shorts in the int (Output = Input<<16+Input2).

Int to Shorts - this script object used to unpack int value in 2 shorts (Output = Input[Input2]).

Read bit - this script object used to read bit of the input value (Output = Input[Input2]).

Set bit - this script object used to set bit of the input value (Output = Input | 1<<Input2).

Reset bit - this script object used to reset bit of the input value (Output = Input & ~(1<<Input2)).

Arithmetic library

Addition - this script object used to arithmetic operation addition for input values (Output = Input + Input2).

Subtraction - this script object used to arithmetic operation subtraction for input values (Output = Input - Input2).

Multiplication - this script object used to arithmetic operation multiplication for input values (Output = Input * Input2).

Division - this script object used to arithmetic operation division for input values (Output = Input / Input2).

Modulo - this script object used to arithmetic operation modulo for input values (Output = Input % Input2).

Power - this script object used to arithmetic operation power for input values (Output = Input^{Input2}).

ABS - this script object used to arithmetic operation absolute for input value (Output = |Input|).

Sign - this script object used to arithmetic operation sign for input value (Output = -Input).

Int - this script object used to arithmetic operation for getting integer part of the input value (Output = int(Input)).

Sqrt - this script object used to arithmetic operation sqrt of the input value (Output = Math.Sqrt(Input)).

Compare library

Equal - this script object used to comparison operation equal for input values (Output = Input == Input2).

Not Equal - this script object used to comparison operation not equal for input values (Output = Input != Input2).

Greater - this script object used to compare operation greater for input values (Output = Input > Input2).

Less - this script object used to compare operation less for input values (Output = Input < Input2).

Equal or Greater - this script object used to compare operation equal or greater for input values (Output = Input >= Input2).

Equal or Less - this script object used to compare operation equal or less for input values (Output = Input <= Input2).

Select library

Selectable enable - this script object used to select value form Input2 if Input1 is true (IF Input==true THEN Output=Input2).

Selectable negate - this script object used to select value form Input2 if Input1 is false (IF Input==false THEN Output=Input2).

Minimum - this script object used to select minimum value of Input2 and Input1 (Output=Min(Input, Input2)).

Maximum - this script object used to select maximum value of Input2 and Input1 (Output=Max(Input, Input2)).

Arrays library

Index read - this script object used to select array's element. Input1 is an array. Input2 is index of element (Output = Input1[Input2]).

Index write - this script object used to change array's element. Input1 is an element. Input2 is index of element (Output[Input2] = Input1).

Triggers/Counters library

Rising edge trigger- this script object used to generate rising impulse duration PV ms when Input1 get TRUE from FALSE.

Falling edge trigger- this script object used to generate falling impulse duration PV ms when Input1 get FALSE from TRUE.

RS trigger- this script object used to imitate RS trigger.

Timer ON- this script object used for delay timer for the duration PV when Input1 get TRUE from FALSE.

Timer OFF- this script object used for delay timer for the duration PV when Input1 get FALSE from TRUE.

Counter- this script object used to count impulses of boolean value in Input1. Counter resets when Output become equal PV.

Counter Down- this script object used to count impulses of boolean value in Input1. Counter starts from value PV. Counter resets when Output become equal 0.

Multivibrator - this script imitates impulse generator with PV period. It starts when IN1 changed from *false* to *true*.

Trigonometric library

Degrees to radians - this script object used to convert degrees to radians.

Radians to degrees - this script object used to convert radians to degrees.

Sine - this script object used to calculate sin of Input value. (Output = $\sin(\text{Input})$).

Cosine - this script object used to calculate cos of Input value. (Output = $\cos(\text{Input})$).

Tangent - this script object used to calculate tag of Input value. (Output = $\text{tag}(\text{Input})$).

Arc Sine - this script object used to calculate arc sin of Input value. (Output = $\text{arc sin}(\text{Input})$).

Arc Cosine - this script object used to calculate arc cos of Input value. (Output = $\text{arc cos}(\text{Input})$).

Arc Tangent - this script object used to calculate arc tag of Input value. (Output = $\text{arc tag}(\text{Input})$).

Hex operations library

Hex to Integer - this script object converts hex value into integer.

Integer to Hex - this script object converts integer value into hex.

Call screen library

Call screen - this script object used to call screen when Input's value turns from *false* to *true*.

Call popup - this script object used to call popup screen when Input's value turns from *false* to *true*.

Date and time library

Current date and time - this script object used to get date and time components depending on Input value:

0 - get seconds.

1 - get minutes.

2 - get hour of the day considering AM/PM.

3 - get hour of the day.

4 - get day of the week (1-Sunday, 2-Monday...).

5 - get day of month.

6 - get month (0 - January, 1 - February...).

7 - get year.

8 - get minutes of the day (hour*60 + minutes).

Servers library

IP or URI address - this script object used to change server's IP or URI address when Input's value changed.

Reconnect - this script object used to reconnect server when Input's value turns from *false* to *true*.

Recipes library

Select recipe - this script object used to choose recipe row. Input2 is an input that contains name of the recipe. Input1 is number of the row (starting from 1). Output = true if recipe row is chosen.

Simulation

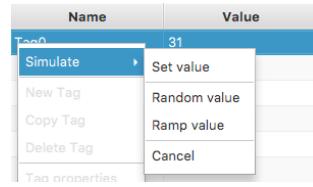
You can simulate behavior of your project. To start simulation select the menu item **Project** and **Run simulation** or click button on the **Toolbar**.

If you use users in your project Login dialog will appear. Select user and enter password in the field. Now you can simulate your project.



You can change value of the tag by double clicking on it in the **Project window ->Tags**. Or you can click by right button on the tag and select *Simulate* and *Set value* menu item. Also you can simulate behavior of the tag:

1. *Random value* - periodically change the value of the tag randomly.
2. *Ramp value* - periodically change the tag value from 1 to 100 by adding 1.



By selecting *Simulate* and *Cancel* you annul the task.

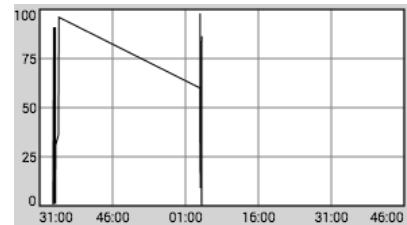
Also it's possible to change value of the tag using control graphical objects of your project like text, buttons, slider, counter and etc. For example if you use Text object enable output property and bind to the tag you want to use.

During simulation click on it and enter value you want.

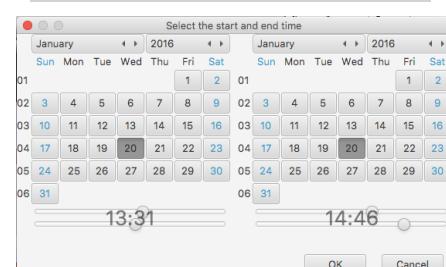


Also you can simulate behavior of **Trend** and **Events log** objects. Place these objects on the **Canvas**. Set properties of the object as described in previous chapters.

During simulation trend will look like this:



To select start and end time click on it. You'll see the following dialog. Select times and click OK.



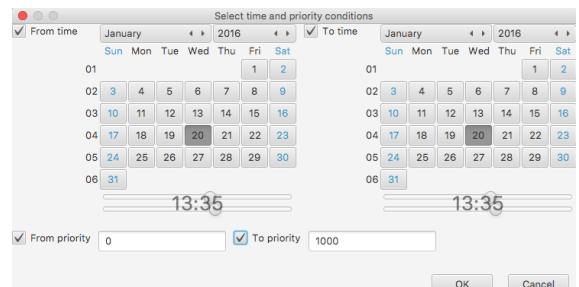
During simulation Events log will be look like this:

1. To **View** message in the separate dialog double click on it or click right button on it and select **View** menu item.
2. To acknowledge record click by right button on it and select **Acknowledge** menu item.
3. To acknowledge all records on the table click by right button on the table and select **Acknowledge All** menu item.
4. To delete record click by right button on it and select **Delete** menu item.
5. To delete all records on the table click by right button on the table and select **Delete All** menu item.

Events (All)						
Name	Time	Type	Pri...	Message	Value	
Tag0	20/01/02:05:04	HiHi	50	Lever too high	86	
Tag0	20/01/02:05:03	LoLo	50	Level too low	77	<input type="button" value="View"/>
Tag0	20/01/02:05:02	HiHi	50	Lever too high	87	<input type="button" value="Acknowledge"/>
Tag0	20/01/02:05:02	LoLo	50	Level too low	83	<input type="button" value="Acknowledge All"/>
Tag0	20/01/02:05:01	HiHi	50	Lever too high	89	<input type="button" value="Delete"/>
						<input type="button" value="Delete All"/>

You can select records that you want to see in the table.

Click on the table's title. You'll see **Select time and priority conditions** dialog. Select start and end times of records displayed in the table. You can also set records with what priorities will be displayed.



Load on device

When project is created (screens, servers, tags, scripts and users), the project can be loaded on the mobile device or other PC. For this purpose, first the corresponding TeslaSCADA Runtime mobile app on the Android device or PC apps on the Windows, Linux or MAC OS must be installed and started.

If the app has now been installed on the mobile device or PC, there are 2 ways to load the project to the device.

1. Network method.
2. Manual method.

Network method

This method must, the PC on which the TeslaSCADA IDE is started, and the mobile device or PC on which TeslaSCADA Runtime started and the project will be stored in a Wi - Fi network (note IP addresses) are.

Procedure:

1. Enable WiFi on your mobile device or PC where installed

TeslaSCADA Runtime.

2. Start the TeslaSCADA2 Runtime app.

3. Open it in the editor TeslaSCADA2 IDE the desired project to be transferred.

4. Select the menu item *File* and *Load on Device*.

5. It now opens the dialog **Load on Device** and it

will now search for mobile devices with the active TeslaSCADA2 Runtime. You can start a broadcast search and browse the entire network. However, since some routers do not forward broadcasts, there is also the possibility of a specific device search on the IP address.

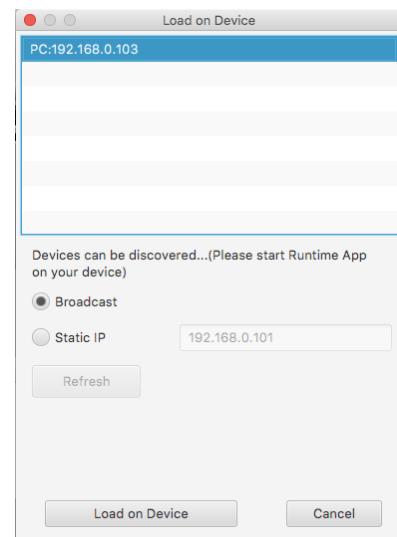
This search takes a normally 5-10s. In individual cases it may happen that this search can take up to 3 minutes.

If you can't find a device you can try to restart **Load on Device** dialog and TeslaSCADA2 Runtime application.

6. After a successful search in this dialog box all found mobile devices with active TeslaSCADA Runtime app will be shown.

7. Now select the desired target device and press the **Load on Device** button.

8. After a successful transfer, the target device with TeslaSCADA2 Runtime load new project.



Manual method

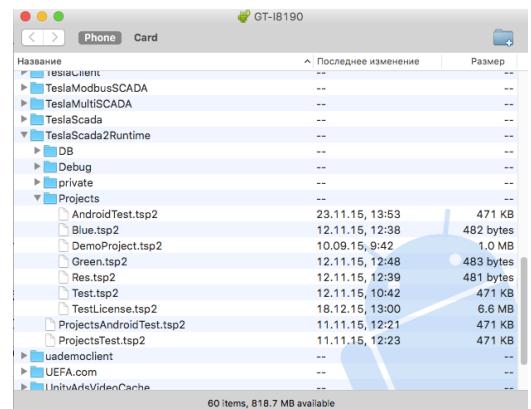
Another way to load a project on the mobile device is a file explorer such as: the **Android File Transfer** for Mac OS.

Once the TeslaSCADA Runtime installed mobile app and once started on the sd card, a folder called

TeslaSCADA2Runtime->Projects is created.

Now, if the project which has been stored as .tsp2 file from the Windows, Linux or MacOS system TeslaSCADA IDE(The path was chosen when you first save of the project) will be manually copied to the folder of the sd card of the TeslaSCADA Runtime mobile device, the app can be started normally. Now loads the app, the file from this folder by clicking *Load* on the main menu of TeslaSCADA Runtime.

There is no problem to manually copy to the PC where TeslaSCADA Runtime is installed. You can use local network, flash driver or use any other storage device.



Import for iOS

When project is created, it can be imported for iOS mobile devices. To do import for iOS devices you should enter *File* and select *Import for iOS*. When you do it Import for iOS window will appear.

To do import project for iOS devices click *Import*, file dialog will appear, enter name of the file and click OK.

Import file has *.tsp2db extension.

This file based on SQL database

format and you can open and check it by using softwares for working with SQL databases. You can also open imported file by clicking *Open* button. Imported or opened file will be appeared in the text field. To activate project:

1. Choose license type.
2. Enter license number.
3. Click *Activate* button (it will change background colour to the green and «License available for activation» message will appear).
4. If you want to deactivate license click *Deactivate* button (it will change background colour to the green).
5. Load project on iOS device.
6. When loading of the project is completed on iOS device «Activation completed» message will appear (device should have an Internet access).

If TeslaSCADA2 Runtime has now been installed on the iOS mobile device (iPhone or iPad), there are 2 ways to load the imported project on the device.

1. Network method.
2. Manual method.

Click *Load on iOS device* to use Network method to load imported file on your iOS device.

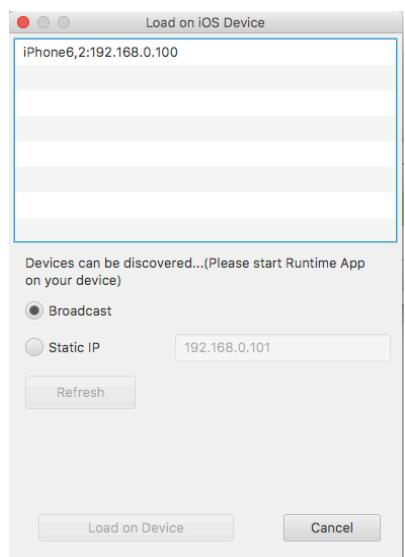


Network method

This method must, the PC on which the TeslaSCADA IDE is started, and the iOS mobile device on which TeslaSCADA2 Runtime started and the project will be stored in a Wi - Fi network (note IP addresses) are.

Procedure:

1. Enable WiFi on your mobile device where installed TeslaSCADA2 Runtime.
2. Start the TeslaSCADA2 Runtime app.
3. In TeslaSCADA2 IDE select the menu item *File* and *Import for iOS*.
4. Open the desired imported project. Click *Load on iOS device*.
5. It now opens the dialog **Load on iOS Device** and it will now search for mobile devices with the active TeslaSCADA2 Runtime. You can start a broadcast



search and browse the entire network. However, since some routers do not forward broadcasts, there is also the possibility of a specific device search on the IP address.

This search takes a normally 5-10s. In individual cases it may happen that this search can take up to 3 minutes.

If you can't find a device you can try to restart **Load on iOS Device** dialog and TeslaSCADA2 Runtime application on iOS device.

6. After a successful search in this dialog box all found mobile devices with active TeslaSCADA2 Runtime app will be shown.

7. Now select the desired target device and press the **Load on Device** button.

8. After a successful transfer, the target device with TeslaSCADA2 Runtime load new project.

Manual method

Another way to load an project on the iOS mobile device is iTunes ->File Sharing.

Procedure:

1. Open iTunes on your Mac or PC.
2. Connect your iPhone or iPad to your computer using the USB cable that came with your device.
3. Click your device in iTunes.



4. In the left sidebar, click Apps. Then, scroll to the File Sharing section at the bottom of the page.
5. Select TeslaSCADA2 Runtime see which files are available for sharing in that app on your device.
6. Now you can copy your imported project (*.tsp2db) to the documents folder of TeslaSCADA2 Runtime apps.

Use Telegram Bot

If you want to get events notification from your project in TeslaSCADA2 Runtime you can use Telegram messenger for this purpose. To do this you should create Telegram Bot:

1. You should have Telegram messenger installed on your device and have an account.
2. Open in browser <https://telegram.me/botfather>
3. Click button «Send message» or «Open in Telegram Web» (you should have login in web telegram client).
4. Open your Telegram client and choose BotFather.
5. Click button Start or type /start.
6. Enter /newbot.
7. Enter your bot's name. The name should be unique. This name you should enter in **Bot's name** field of project properties.
8. Then you should choose username for your bot.
9. After entering username you'll get the telegram bot's token. Enter it in **Bot's token** field of project properties.

Now you can use telegram bot in getting notification messages from TeslaSCADA2 Runtime. To do this you should find your created bot in your telegram messenger client and click button Start or enter /start. To stop getting notification messages enter /stop. Also you can get some information from your project:

1. Enter /tags to get current values of tags. You'll get information only from currently monitored tags (tags that enable history, events and tags of objects that displayed on currently opened screen).
2. Enter name of the tag used in your project. You'll get information about value of this tag and if tags supports history you'll get trend for last hour period. You can choose other period by clicking proper button.

Warning don't use underline in the name of the tags. Telegram have problems with working with this kind of names.

At this moment you can use Telegram bot only on desktop versions of TeslaSCADA2 Runtime.