mikroprojekt

R&D, design and production of electronic and computing systems
Aleja Blaža Jurišića 9, HR_10040 Zagreb, Croatia
T/**F** +385 1 2455 659 | contact@mikroprojekt.hr |www.mikroprojekt.hr

IQ Editor – User Manual

HoC HMI Solution

13/7/2015

MPP150101

Rev. 1.1

MB:03552624 | **OIB**:10067934951 | Zagrebačka banka **žiro račun** 2360000-1101431443 | **IBAN**: HR0823600001101431443
Subjekt upisan na Trgovačkom sudu u Zagrebu pod brojem MBS 080037797 | Temeljni kapital 21.000,00 kn uplaćen u cijelosti
Osnivači/članovi društva: **Damir Ježić**, jedini član d.o.o.
Osoba ovlaštena za zastupanje: Damir Ježić, direktor, zastupa društvo pojedinačno i samostalno

Table of contents

Revision History

Revision	Date	Author	Modification
1.0	31.03.2015.	MR	Initial
1.1	07.07.2015.	IP	Upgrade

Related Documents

ID	Code	Description

1. Overview

In this document we will show, in steps:

- How to set up the IQ-Editor
- Outline the basic tool concepts and usage flow
- Demonstrate basic concepts of HMI design

1.1. Prerequisites

- IQ-Editor 1.51 Installer
- Project files accompanying this presentation
- Microsoft Windows OS environment

1.2. What is the IQ-Editor ?

IQ Editor is a MS Windows application with a **G**raphical **U**ser **I**nterface (GUI) design tool. The main purpose of the **IQ-Editor** is designing of **IQ-Projects** that run on **IQ-Engine**.

IQ-Editor enables:

- **W**hat-**Y**ou-**S**ee-**I**s-**W**hat-**Y**ou-**G**et (WYSIWYG) UI design using graphical objects
- Simple building of HMI (**H**uman **M**achine **I**nterface) pages using bitmap graphics
- Create an HMI-on-Chip graphical user interface projects
- Add and modify various graphical objects, pages and buttons
- Preview and test your GUI in live interaction
- Generate deployment files for your HMI-on-Chip embedded hardware

- Unlimited[1] number of
  - pages in IQ project
  - objects on a page
  - tags
  - actions and action triggers
- Plenty of different
  - object types, including logical objects
  - tag types, including string types for 90 code pages
  - action types, including flow control actions
- Very comfortable and intuitive what-you-see-is-what-you-get editing
- Familiar text formatting and graphics positioning options

Some further interesting features are:

- Transparent page background; useful for example when IQ-Engine is used to display an on-screen display menu over video contents
- **Alpha blending** and **transparencies** for all graphic objects
- Image dithering, enabling **24-bit graphics quality** even on displays with lesser capabilities
- **Object overlapping**; IQ-Engine optimally draws any number of overlapped objects, including transparent ones
- **Styles**, setting consistent look to objects
- **Tags** and actions model, setting consistent behavior template to all objects types
- Automatic IQ project conversion to different target device; e.g. with different screen size
- Indexed page, object and tag naming provides quick and immediate renaming in all IQ-Editor windows
- **Smart font recognition** provides the best possible font match on targets with limited font file resources
- Automated navigation with keys through control objects
- Controllers for handling local, resident and memory tags
- Embedded and modifiable screen saver
- Pop-up pages - dialogs and windows that appear above the normal page

[1] *"**unlimited**" means that number is limited only with the amount of storage memory available on your target device for IQ project and with the amount of working memory for processing currently opened page.*

- Global page - page that is active all the time during project execution below the normal page
- Merging projects
- **Dynamic editing** with editable fields (including right-to-left editing)
- Keyboard emulation for all supported code pages
- Smart input control
- Detailed warning reporting on project compilation
- Dynamic tag viewing and editing (*currently on PC Simulator*)

1.2.1. IQ-Project

An **IQ Project** is most similar to a presentation or Visio project: it is a single file, which contains a list of pages. Each page can contain various usable objects like text fields that indicate state of some manufacturing machine or illustrative static images.

1.2.2. IQ-Engine

IQ-Engine is small-footprint fast-performance embedded **application** that runs on you target device. Its task is to provide supervision and control of external devices connected to your target device, using graphical user friendly interface designed by ***IQ-Editor***. Its PC version is also delivered in order to quickly test your IQ project during designing process.

2. Installing the IQ Editor

IQ-Editor v1.50 (or newer) installer can be downloaded from the Mikroprojekt website (link).

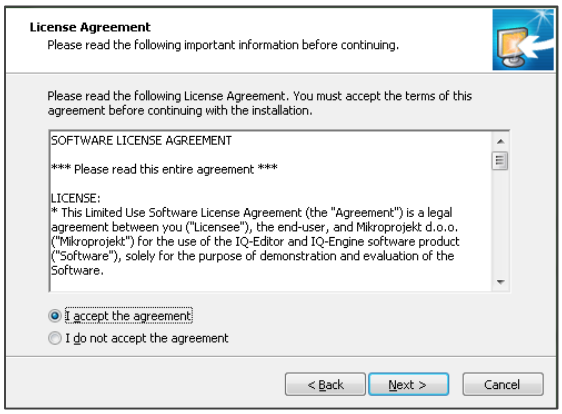
After downloading file from the website, you may start to install the software by running the executable.

The installer will also install the Microsoft Visual C++ redistributable files required for running the IQ-Editor.

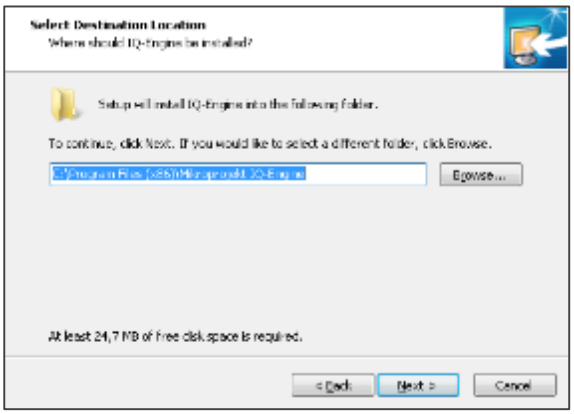
Installation procedure is pretty straightforward and standard for Microsoft Windows environment, and there is no surprises here.

Let the installation wizard guides you through the installation process. (Picture 1.)

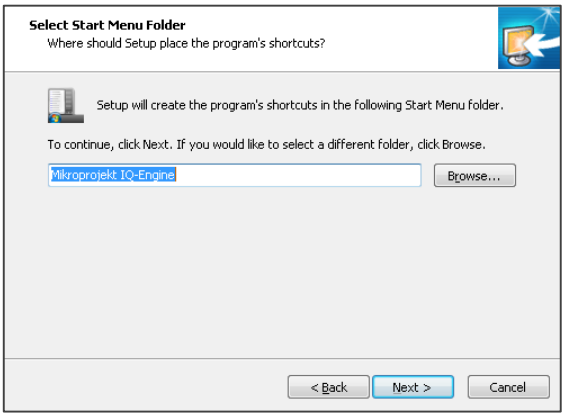
Picture 1: Installation Wizard



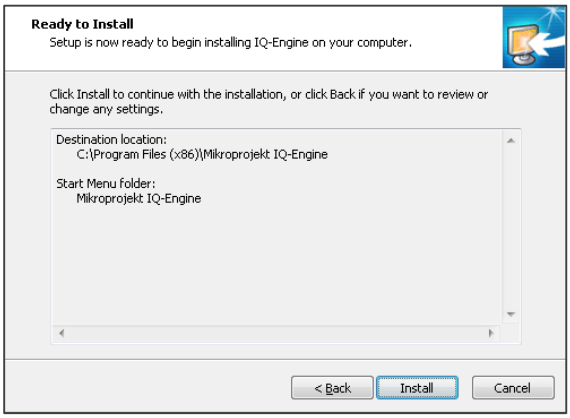
Picture 2: License Agreement



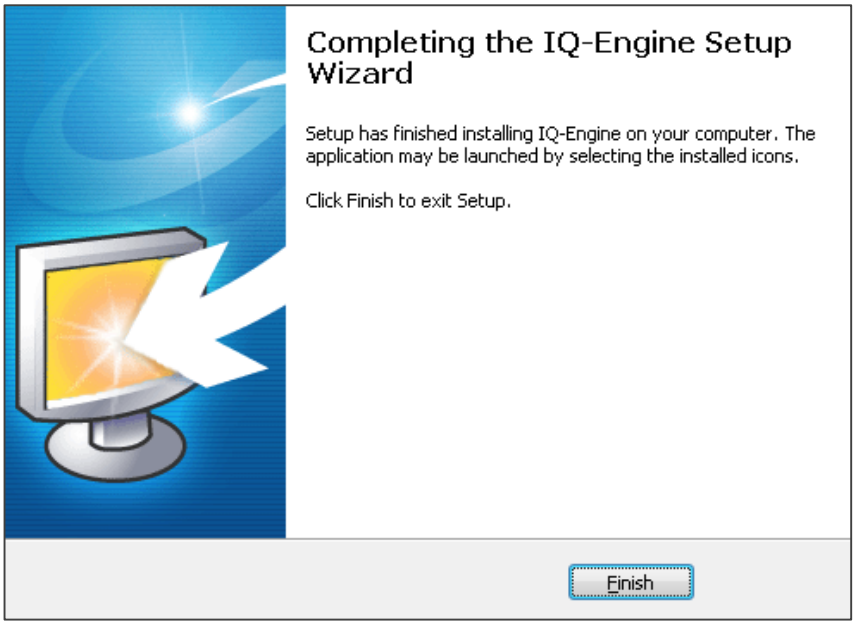
Picture 3: Destination Location



Picture 4: Start Menu Folder



Picture 5: Ready to Install



Picture 6: Installation is done

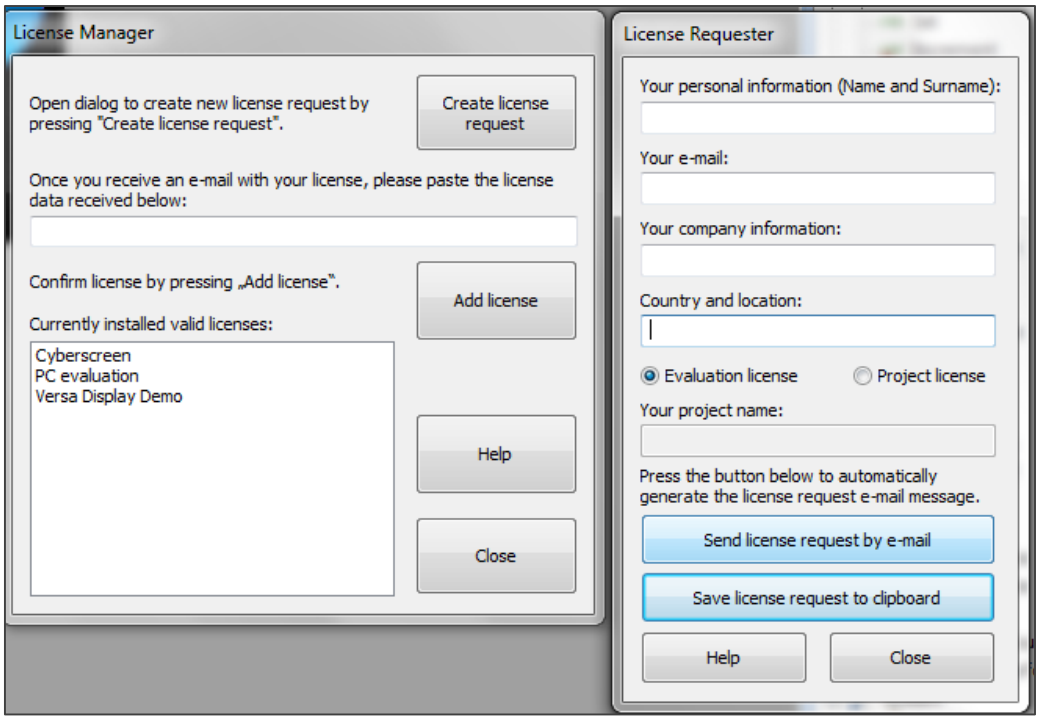
Once the tool is installed, the licensing manager dialog will appear (Picture 7)

Just follow this procedure (***Next >***) and then:

- Click on „***Create license request***"
- Enter your data
- Select „***Evaluation license***"
- Send the license request by e-mail

This will generate the license request e-mail for „PC Evaluation" which can be sent out to Mikroprojekt.

Note:
If an e-mail client or handler is not available on your machine, please press the „***Save license request to clipboard***" button. This will copy the license request, which can then be pasted in a text file or a webmail interface.



Picture 7: License Dialog

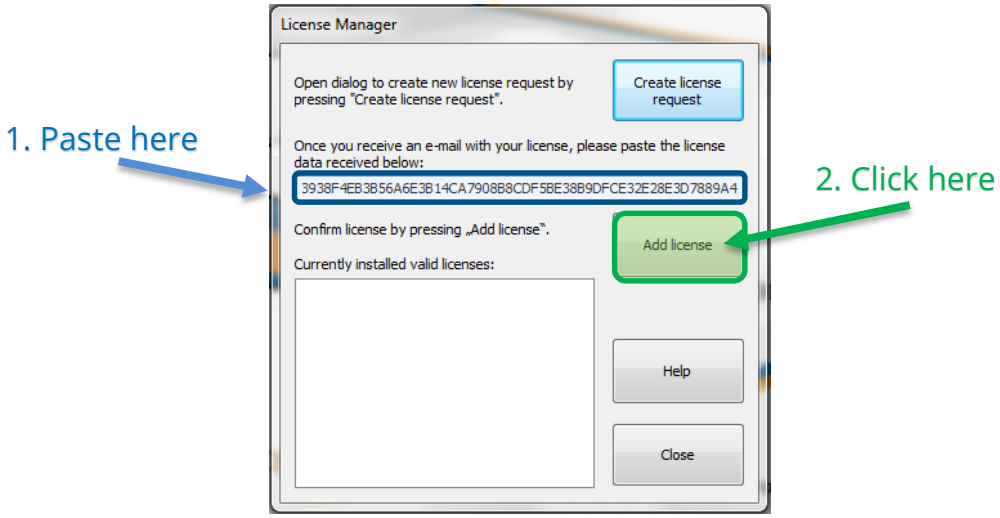
Please send out the request to licensing@mikroprojekt.hr in order to obtain a license.

After you have received an e-mail with your license key(s), enter them, one by one, in the key field and click the „**Add license**" to add your license. (Picture 8.)

The „**PC Evaluation**" license should be loaded first and it is mandatory for live interface preview on your computer. Additional licenses cover additional hardware platforms

Note
If you have a certain hardware supported by the HMI-on-Chip solution which you plan to use, please repeat this procedure to generate a new request for this hardware as well. This will provide you immediately with a license for further lessons.

- Instead of Evaluation license, click on „***Project license***"
- Enter project name

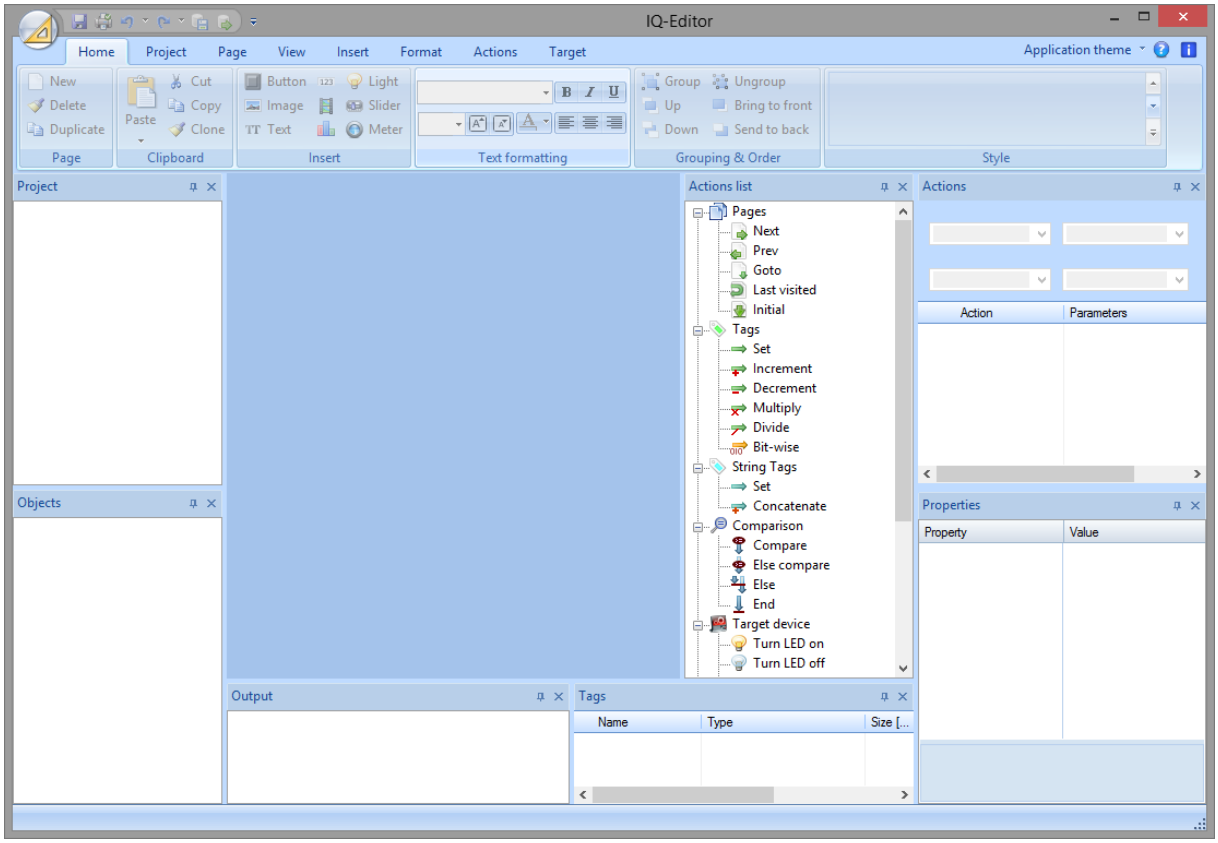


Picture 8: Entering license(s)

After you have loaded your licenses, press „**Close**" and the main editor window will open (Picture 9.).

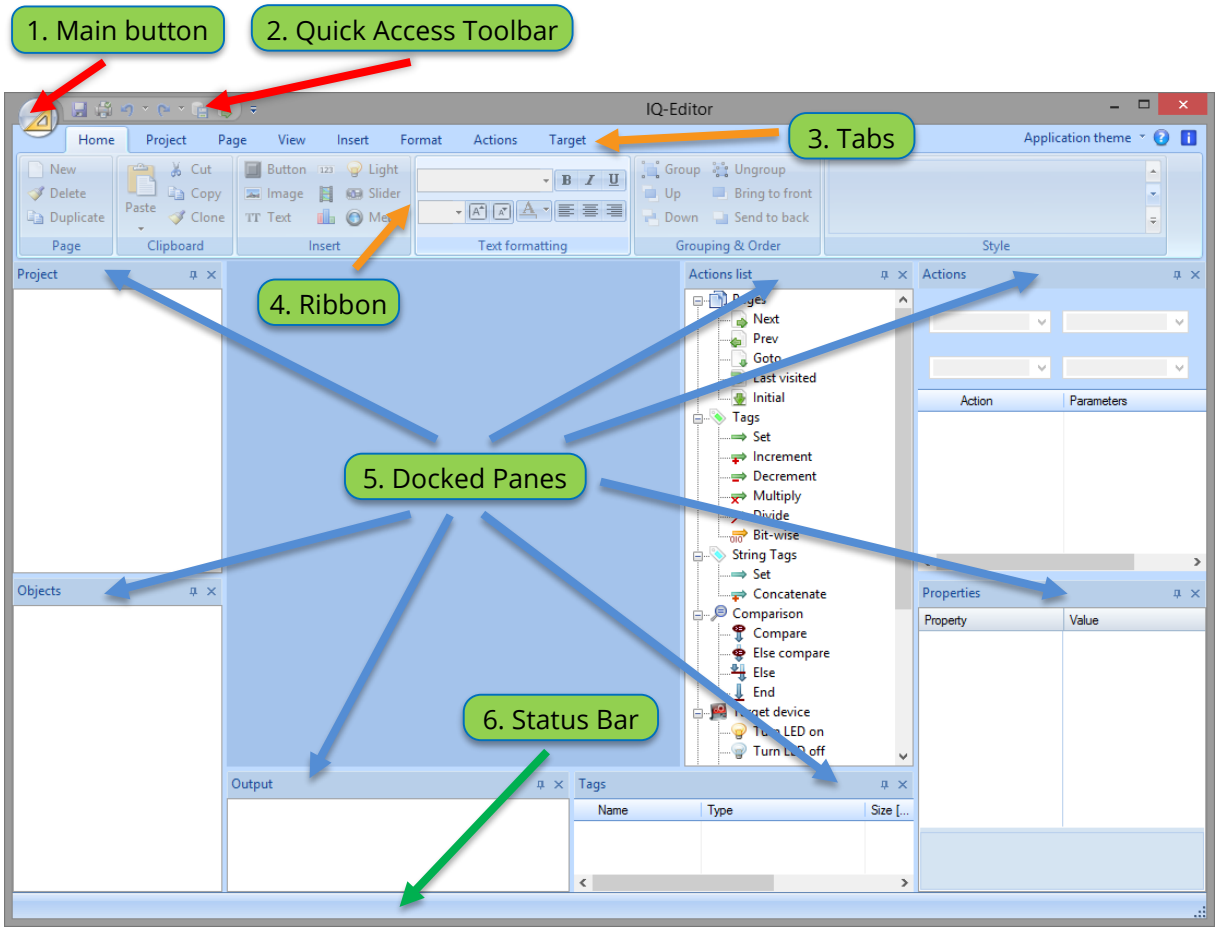
3. Working Environment

The IQ Editor's working window is quite similar to other applications in Microsoft Windows environment, with the standard set of window controls and panes which we are going to explain in details further in this document.



Picture 9: IQ Editor | Main Window

The **IQ-Editor** main window is divided into multiple parts as we can see on the Picture 10.



Picture 10: The Main Window Parts (Docked)

Main window parts are:

1. Main Button
2. Quick Access Toolbar
3. Tabs
4. Ribbons (Tabbed Ribbons)
5. Docked Panes
    - Project | Objects | Action List | Actions | Properties | Tags | Outputs

The picture above shows the standard working window with all panes docked to main window.

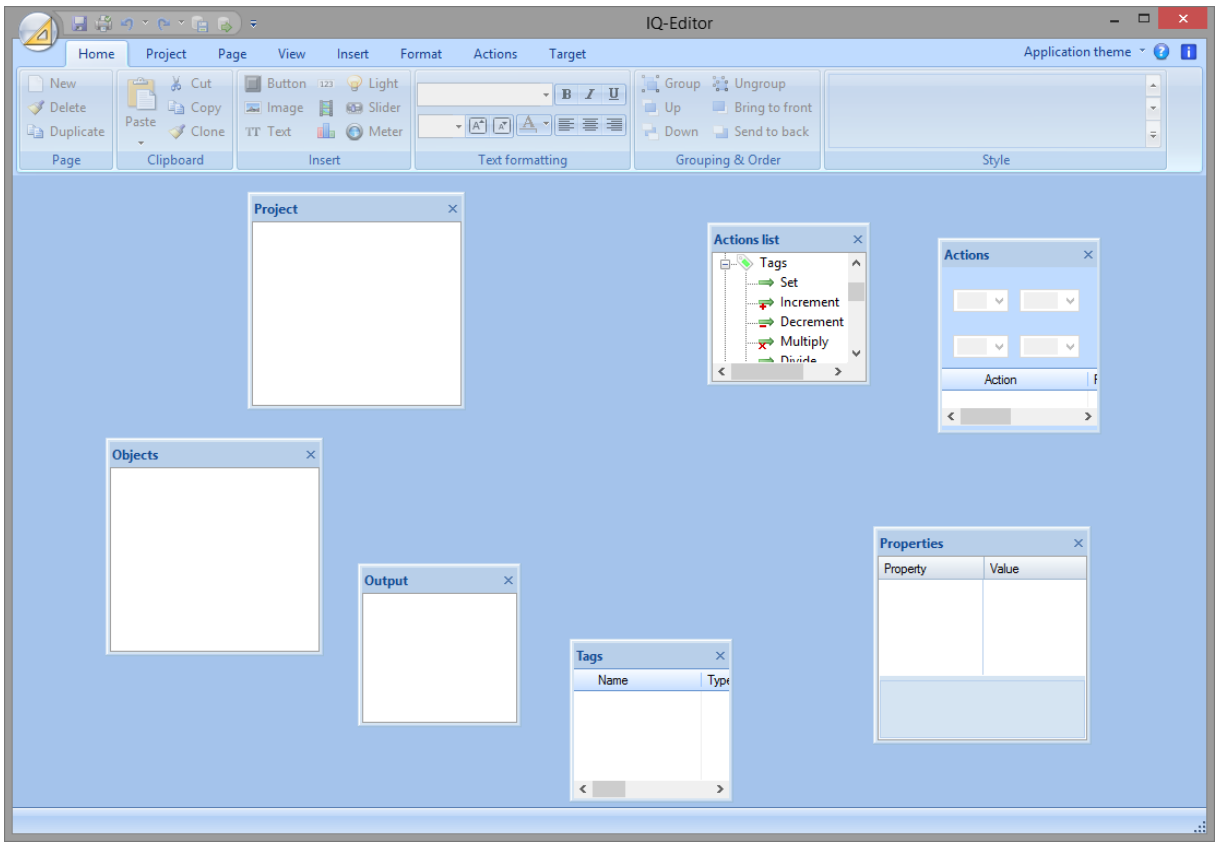
Of course, all these panes can be undocked and rearranged, so let see how we can do it.

3.1. Dock / Undock panes

Picture 11: Dragging the pane

Place your mouse pointer over the pane's title bar and using drag & drop technique, un-dock the pane. (Picture 11.)

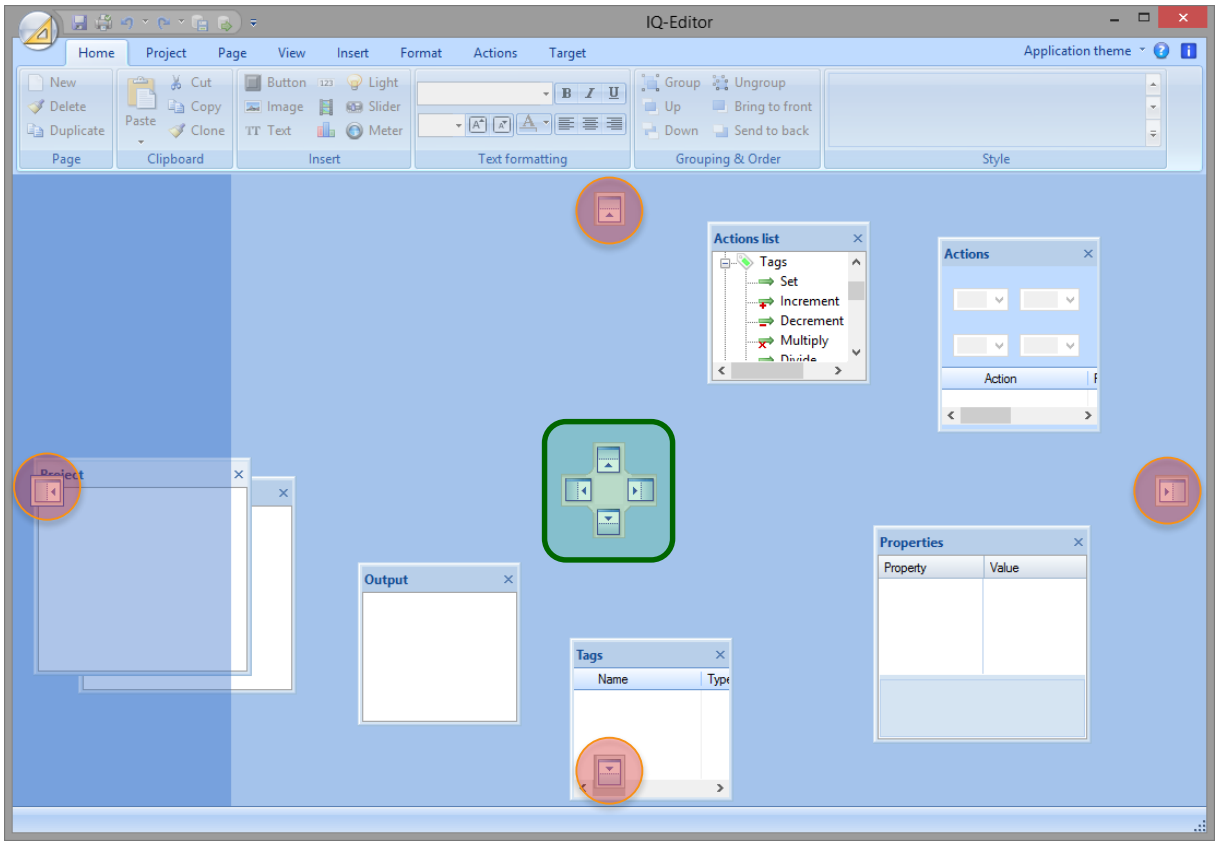
We can apply this on all panes in our working window, as you can see on Picture 12.



Picture 12: Undocked panes

Now, how can we get panes back docked to the main window ?

Let say, for example, we want to dock the “Project” pane. If we start to drag the undocked pane, we will notice the docking control buttons in the main window, as it is shown on Picture 13.



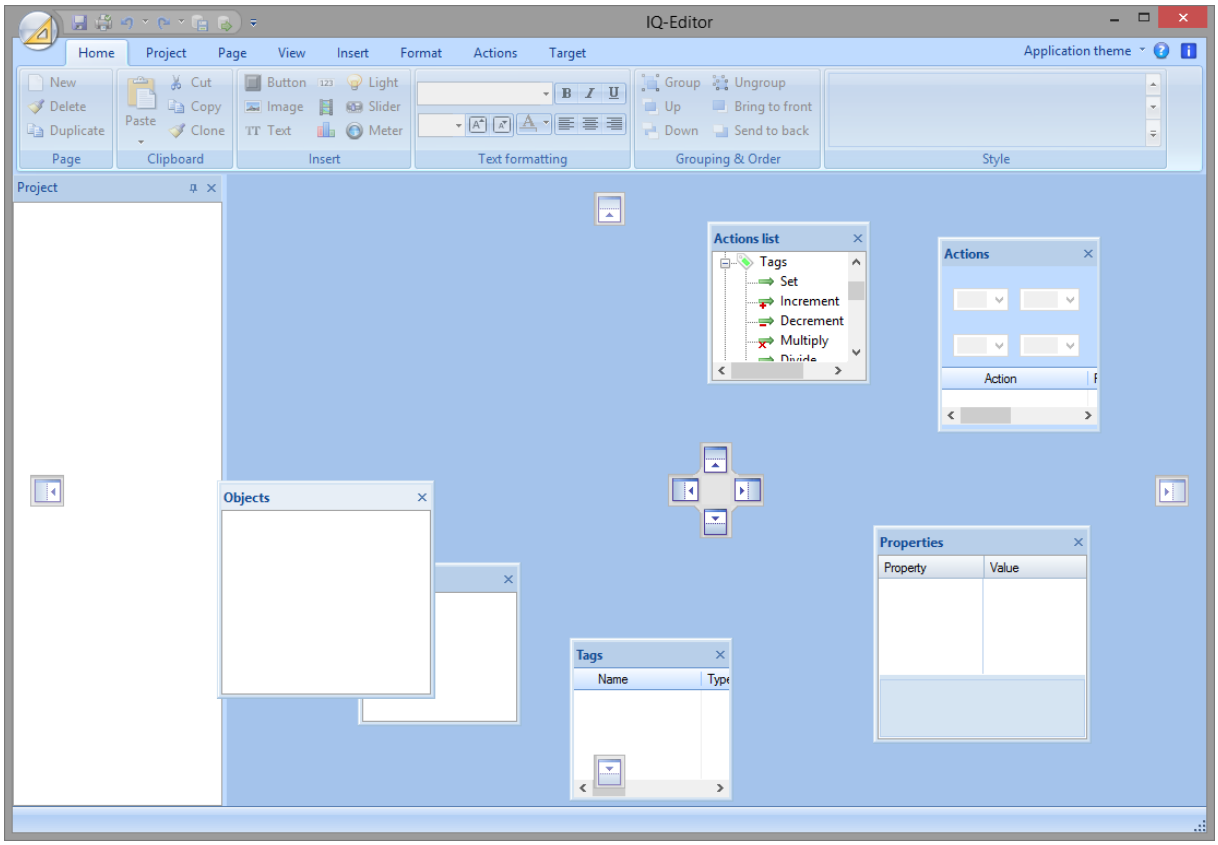
Picture 13: Docking the pane

There are 3 sets of control buttons for docking panes, as we can see on Picture 13 and Picture 16:

1. **Side-edge** controls (*red circles, Picture 13*)
2. **Next-to** controls (*green square, Picture 13)*
3. **Relative-in-pane** controls (*blue oval, Picture 16*)

When we drop a pane on the “**Side-edge**” control, the pane will be docked on the full width/height edge, depending on the chosen position, and will move all other panes from this edge position (if they exists) right after the dropped pane.

If we try to dock multiple panes (one-by-one) on the same side-edge control position, the last dropped pane will be the closest to the edge.



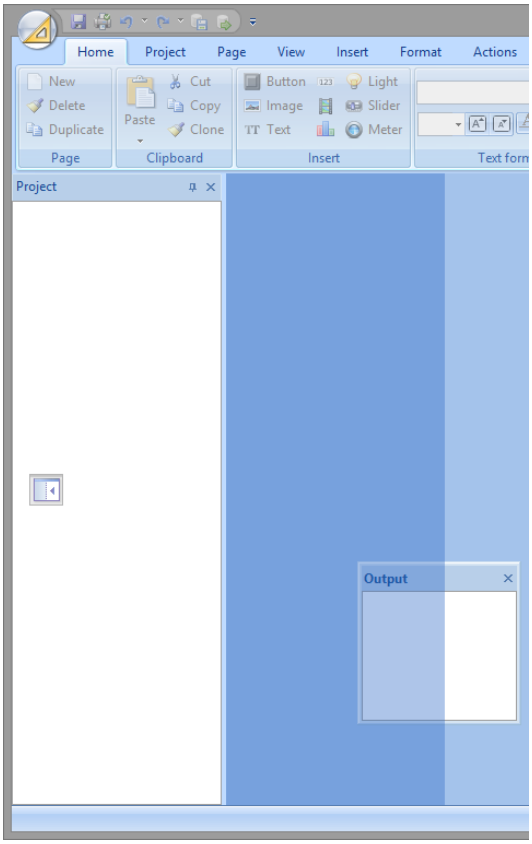
Picture 14: The Project pane is docked

If we are going to drop a pane on the "**Next-to**" control, dropped pane will be placed next to the existing panes, without making any changes on them (Picture 15.)

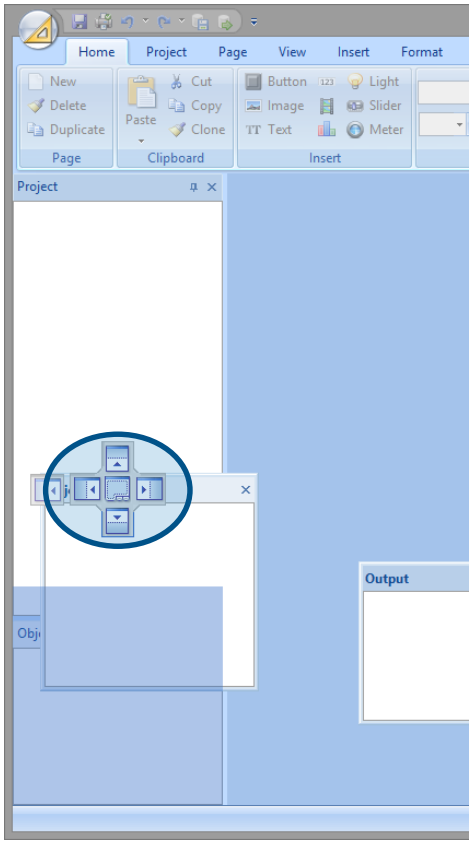
The last control, "**Relative-in-pane**" will be shown if you drag a pane on some already positioned pane and offers 5 possible positions for your floating pane(s) (Picture 16.) :

Table 1: Relative-in-pane combinations

Position of the new, dropped pane	Split docked space	Position of the old, docked pane
Top	*Vertically*	*Bottom*
Bottom		*Top*
Left	*Horizontally*	*Right*
Right		*Left*
Inside (tab on the right)	*No split*	*No change, tab on the left*

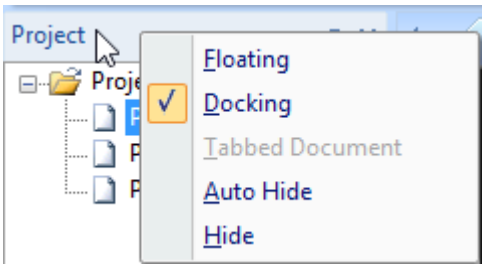


Picture 15:
Docking the pane Next-to existing



Picture 16:
Relative-in-pane control

Another way to dock/un-dock pane is by clicking with the right mouse click on the pane's header, which will open this menu:



Picture 17: Right click menu for panes

Floating	*Un-dock the pane*
Docking	*Dock the pane*
Tabbed Document	*Not in use…*
Auto Hide	*Turn auto hiding on for selected pane*
Hide	*Turn off the pane from working environment*

3.2. Toolbars and menus

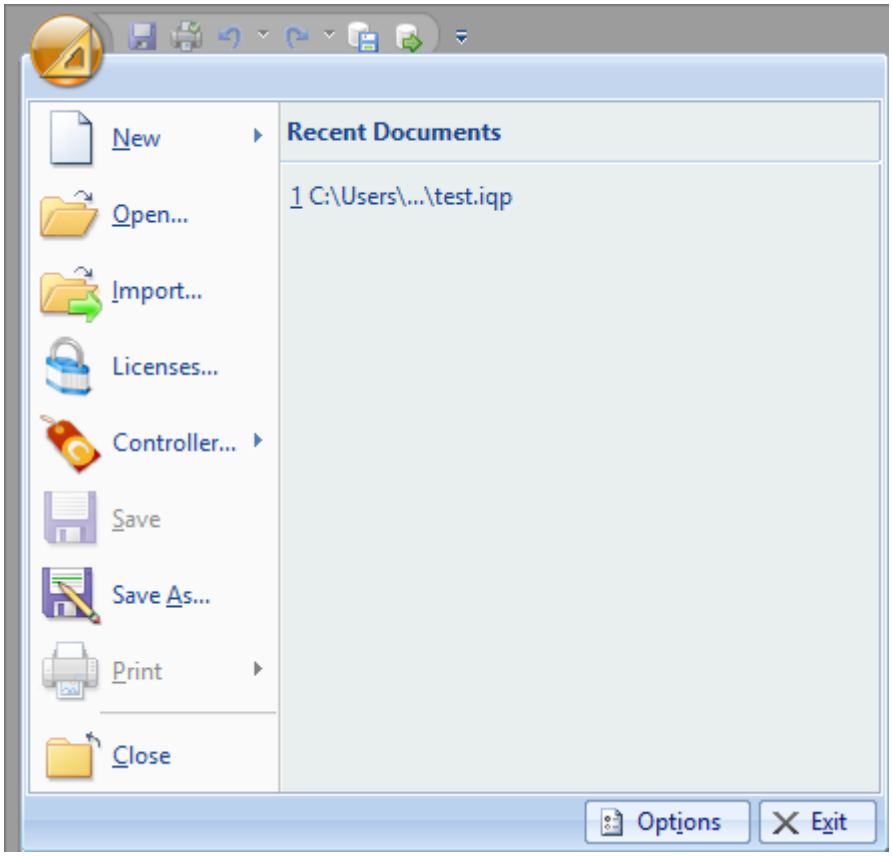
In this chapter we will describe available tools within ribbons.

IQ-Editor is based on the design principles of Microsoft's „Fluent UI" used in its Office 2007, 2010 and 2013 productivity suites and widely adopted in other applications.

As such, it uses the ribbon as the primary interface element.

The overall feel and use will be familiar to people used to the Microsoft Office suite.

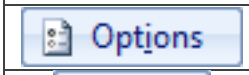
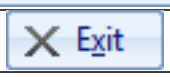
3.2.1. Main menu button



Picture 18: Main menu

The starting point for the activity is the main menu, which is reached by pressing on the Application button with the triangle logo icon in the corner

Table 2: Main menu tools

	**New project**. Here you can start and create a blank project with predefined presets.
	**Open** existing projects. Standard “Open files” dialog for Windows application.
	**Import** an existing IQ project/template into currently opened project.
	**License** Manager - provides adding of new IQ licenses for that HoC-Editor seat
	**Controller** allows loading of new memory controller. *This feature is currently concept presentation that just loads memory tags specific for that controller. It will be fully functional when HoC-Editor will be able to load any memory controller found in Controllers folder (in DLL form).*
	**Save** changes in a new or existing project(s)
	Save new or existing project(s) with another name
	**Print** project
	**Close** current project
	Customize Quick Access Toolbar
	**Exit** the application

3.2.2. Quick Access Toolbar

For frequently used tools as well as for all users who like to use a classic toolbar over the ribbons, ***Quick Access Toolbar*** will be much appreciated. (Picture 19.)

Picture 19: Quick Access Toolbar

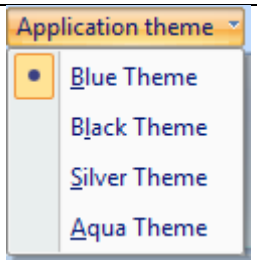
Table 3: Quick Access Toolbar Commands

	Save	*Saves project, including all modifications made on all pages.*
	Quick print	*Prints current page using default printing settings.*
	Undo	*Un-does last or several last actions.*
	Redo	*Re-does last or several last undone actions.*
	Save to file	*Compiles project, if needed, and saves it to a file.*
	Play	*Compiles project, if needed, and runs it on PC version of IQ-Engine.*

You can add various tools through the ***Main Menu > Options.***

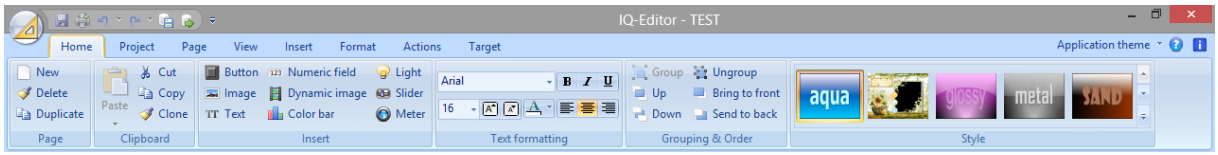
3.2.3. Application Theme

Table 4: Application Theme, Help & About

 **Picture 20 : Application theme**	**Application theme**	*You can choose one of the 4 predefined themes which will be affecting entire working environment of the IQ Editor. Default one is "**Blue Theme**"*
	Help	*Displays help documentation.*
	About	*Displays application version number.*

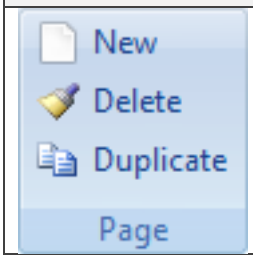
3.3. Ribbons

3.3.1. Home

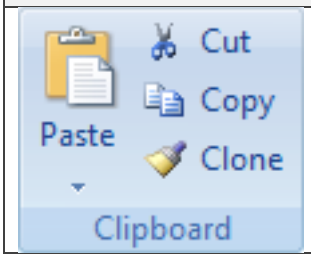


Picture 21: Home Ribbon

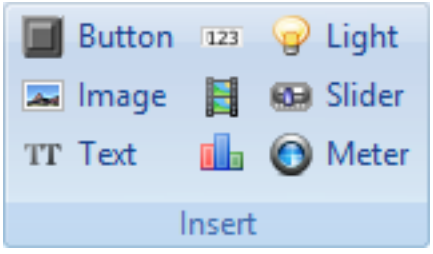
There are several tool groups in Home ribbon. : *Page, Clipboard, Insert, Text Formatting, Grouping & Order and Style*

Group	Description
	**New** (normal) page at the end of the project
	Delete current page
	Duplicate current page

Ribbon group 1: Home | Page

Group	Description	Shortcuts
	**Cut** object	**CTRL + X**
	Copy object	**CTRL + C** | *CTRL + Insert*
	Clone object	**CTRL + D**
	Paste object	**CTRL + V** | *SHIFT + Insert*

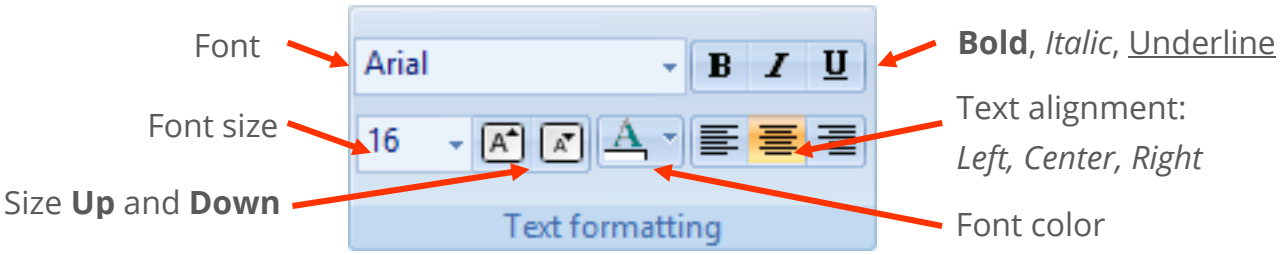
Ribbon group 2: Home | Clipboard



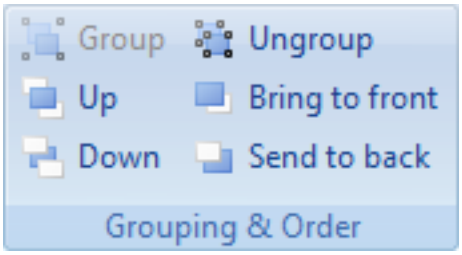
Ribbon group 3: Home | Insert

You may insert various predefined graphical objects on your page(s) by simply clicking on appropriate object, and then “click & expand” object rectangle to desired size.

Also, you can always change current object’s shape by clicking on it and drag & drop control points of your object.



Ribbon group 4: Home | Text Formatting



Ribbon group 5:
Home | Grouping & Order

To group multiple objects in a new single object, follow these steps:

1. Select every single object which you want to group by pressing CTRL key and click on objects, one by one.
2. Click on "***Group***" button on the Home ribbon, or through the context menu (right mouse click and choose "***Group***")

To ungroup previously grouped object to original objects, follow these steps:

1. Select previously grouped object
2. Click on the "***Ungroup***" button

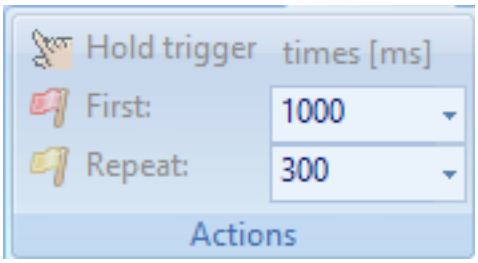
	Selected object will be	**lifted** one layer up
		pushed one layer down
		on the top of all existing objects
		at the bottom of all existing objects

Ribbon group 6: Home | Style

We can choose various ***predefined styles*** from the *Style* group: *Aqua, Glossy, Glossy, Metal, Sand, Vista, Wood,* and one ***custom style***.

3.3.2. Project

Picture 22: Project Ribbon



Ribbon group 7: Project | Actions

This first group defines Hold trigger time in milliseconds.

In this particular setting HMI will wait 1 sec until first touch, and then will repeat the reading every 300 milliseconds.

Ribbon group 8: Project | Screen Saver

With this one, we can create a **Screen Saver page**, and also define how many seconds we need to wait for it to start.

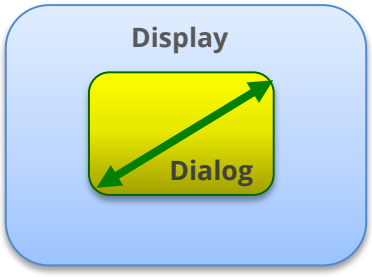
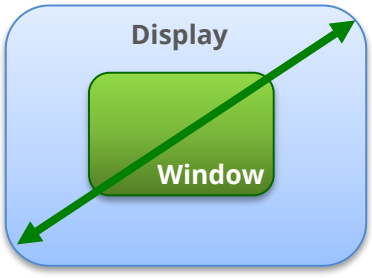
Ribbon group 9: Project | Position

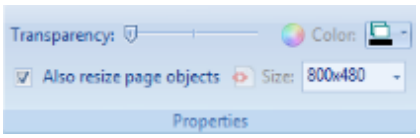
This one is for setting the display offset when we have video stream on the screen. With this settings, we can tune horizontal position, starting from the left side, and vertical from the top of the display.

3.3.3. Page

Picture 23: Page Ribbon

Table 5: Page Ribbon | Insert Group

Icon	Definition	
	Normal page - default page to work with	
	Initial page – normal page that is initially opened on loading project on IQ-Engine	
	Global page – This one will be constantly active all the time during project execution on IQ-Engine along with other active pages. It comes handy in cases when: • *continuous running of logical objects (like timers that continuously work even on page change)* • *fixed appearance and functionality of visual objects (like "Next page" buttons)* • *faster page change (because visual object on global page are not redrawn)* • *project initialization (because "on entering page" events for global page are triggered only on project loading)*	
	Screensaver page	
	Dialog page When we need some dialog to pop-up on the display and restricts the touch control of display area to only this particular dialog window, we will use this option.	
	Window is the same as **Dialog** with one significant difference; you may touch and use the whole display real-estate, with all pages below, not only window's as in Dialog case.	



Ribbon group 10:
Page | Properties

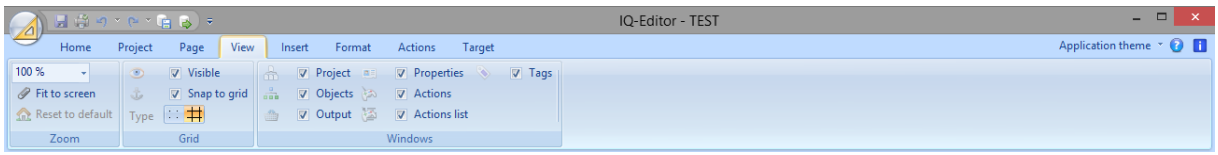
On the **Properties** group, we can change:

- ***Transparency*** of page,
- ***Background color*** and
- ***Page Size***
  - The Page Size is the same for the entire project and usually corresponds to the size of the LCD screen used.

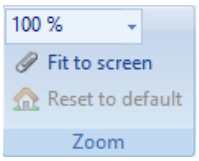
Note:
The project size can be changed/scaled later on, but it is most efficient to set it proper at start.

The checkbox „***Also resize page objects***" controls whether the resize will affect the objects in the page. This way, you can easily retarget existing projects for new resolutions

3.3.4. View

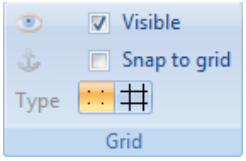


Picture 24: View Ribbon



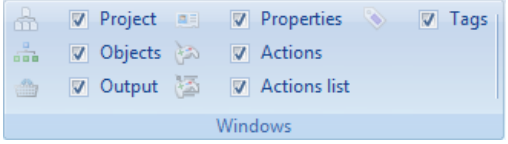
Ribbon group 11: View | Zoom

Standard zooming options, with predefined zoom levels and **Fit to screen** function.



Ribbon group 12: View | Grid

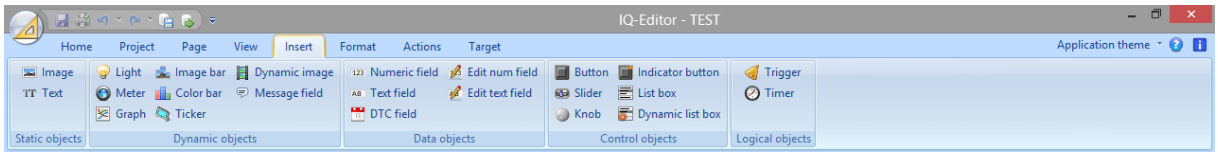
From here we may turn on & off grid (***Visible***) and ***Snap to grid*** option, as well as grid type (dots or lines).



Ribbon group 13: View | Windows

Yet another way to turn on & off main window panes.

3.3.5. Insert



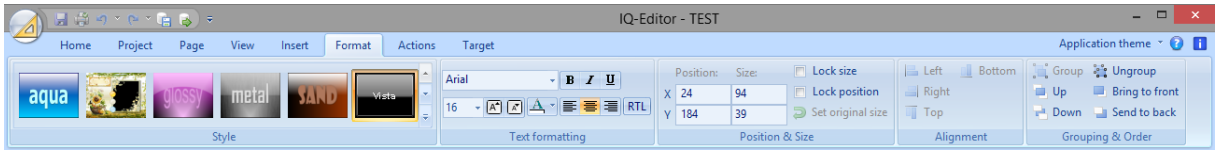
Picture 25: Insert Ribbon

Table 6: Insert ribbon groups

Static objects		
	Image	*Inserts static image.*
	Text	*Inserts static text.*
Dynamic objects		
	Light	*Inserts light. Indicates value of associated tag with specified colors.*
	Meter	*Inserts meter. Angle of the meter needle represents value of associated tag.*
	Graph	*Inserts graph. Displays curve showing values of associated tag through time.*
	Image bar	*Inserts image bar. Associated tag defines bar length; an image is cut to bar length.*
	Color bar	*Inserts color bar. Associated tag defines bar length; a color gradient rectangle is cut to bar length.*
	Ticker	*Inserts ticker. Associated image scrolls from left to right.*
	Dynamic image	*Inserts dynamic image. Value of associated tag is index into the list of images.*
	Message field	*Inserts message field. Value of associated tag is index into the list of messages.*
Data objects		
	Numeric field	*Inserts numeric field. Displayed value is numerical value of associated tag.*
	Text field	*Inserts text field. Displayed value is string contents of associated tag.*
	DTC field	*Inserts date/time/clock field. Displayed date/time/clock value is taken from associated tag.*
	Edit num field	*Inserts editable numeric field.*
	Edit text field	*Inserts editable text field.*
Control objects		
	Button	*Inserts a button.*
	Indicator button	*Inserts button with an indicator, behaving as light object.*
	Slider	*Inserts slider. Position of the slider thumb corresponds to value of associated tag.*
	Knob	*Inserts knob. Angle of the knob thumb corresponds to value of associated tag.*

	List box	*Inserts user-scrollable list box. Selected item corresponds to the value of associated tag.*
	Dynamic list box	*Inserts user-scrollable list box with dynamic text on items. Selected item corresponds to the value of associated tag.*
Logical objects		
	Trigger	*Inserts invisible trigger that provides actions execution on change of specified tag.*
	Timer	*Inserts invisible timer that provides tag value modification within specified range, with specified step and speed.*

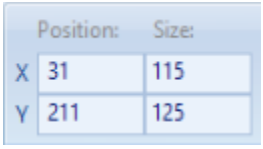
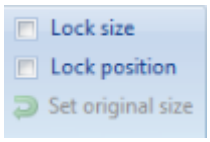
3.3.6. Format



Picture 26: Format Ribbon

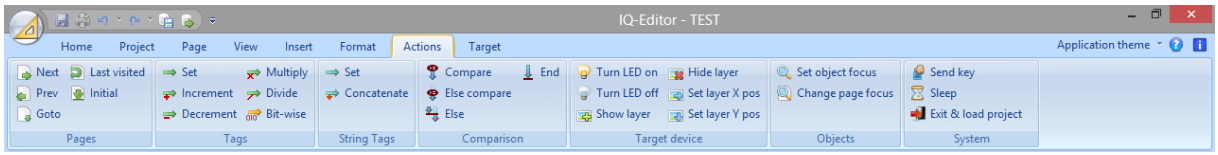
Style and ***Text Formatting*** groups were previously explained, since they are part of the "Home" ribbon too. Only exception here is **RTL** which stands for ***Right-To-Left*** text direction.

Table 7: Ribbon Format | Position & Size and Alignment

Position & Size		
	**X position**	*Position of left side of the object.*
	Y position	*Position of top side of the object.*
	X Size (Width)	*Width of the object.*
	Y Size (Height)	*Height of the object.*
	**Lock size**	*Object size remains constant while object is resized (even if resized as a part of multi-selection).*
	Lock position	*Object position remains constant while object is resized (even if resized as a part of multi-selection).*
	Set original size	*Resets object dimensions to original ones.*
Alignment		
	Left	*Aligns left sides of selected objects.*
	Right	*Aligns right sides of selected objects.*
	Top	*Aligns top sides of selected objects.*
	Bottom	*Aligns bottom sides of selected objects.*

"***Grouping & Order***" group was previously explained, since it is placed on the Home ribbon, as well.

3.3.7. Actions



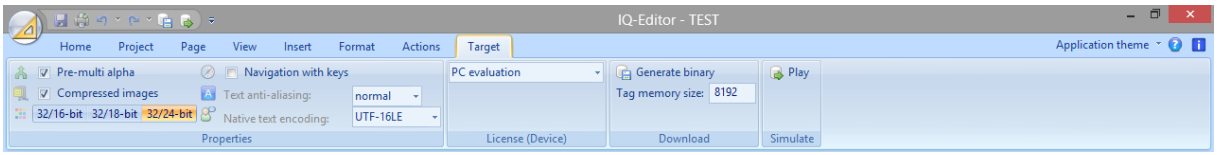
Picture 27: Actions Ribbon

Table 8: Actions ribbon groups

Pages		
	**Next**	*Inserts "**go to next page**" action.*
	**Previous**	*Inserts "**go to previous page**" action.*
	**Go to**	*Inserts "**go to page**" action. Parameter is name of page to go to.*
	**Last visited**	*Inserts "**go to last visited page**" action.*
	**Initial**	*Inserts "**go to initial page**" action.*
Tags		
	**Set**	*Inserts "**set tag**" action. Parameters are tag name and new value (or tag name).*
	**Increment**	*Inserts "**increment tag**" action. Parameters are tag name and modification value (or tag name).*
	**Decrement**	*Inserts "**decrement tag**" action. Parameters are tag name and modification value (or tag name).*
	**Multiply**	*Inserts "**multiply tag**" action. Parameters are tag name and modification value (or tag name).*
	**Divide**	*Inserts "**divide tag**" action. Parameters are tag name and modification value (or tag name).*
	**Bit-wise**	*Inserts "**bit-wise tag**" action. Parameters are tag name, modification value (or tag name) and bit-wise operator.*
String tags		
	**Set**	*Inserts "**set string tag**" action. Parameters are tag name and new value (or tag name).*
	**Concatenate**	*Inserts "**concatenate string tag**" action. Parameters are tag name and new value (or tag name).*

Comparison		
	**Compare**	*Inserts "**compare tag**" action. Parameters are tag name, comparing value (or tag name) and compare operator. Starts comparison block. Also starts the sequence of actions that are executed if comparison successes.*
	**Else compare**	*Inserts "**else compare tag**" action. Starts the sequence of actions that are executed if comparison successes and all previous comparisons fail. Ends the sequence of actions executed on last comparison.*
	**Else**	*Inserts "**else compare**" action. Starts the sequence of actions that are executed if all previous comparisons fail. Ends the sequence of actions executed on last comparison*
	**End**	*Inserts "**end compare**" action. Ends the sequence of actions executed on last comparison. Also ends comparison block.*
Target device		
	**Turn LED on**	*Inserts "**turn LED on**" action. Parameter is LED number.*
	**Turn LED off**	*Inserts "**turn LED off**" action. Parameter is LED number.*
	**Show layer**	*Inserts "**show layer**" action. Parameter is layer index.*
	**Hide layer**	*Inserts "**hide layer**" action. Parameter is layer index.*
	**Set layer horizontal position**	*Inserts "**set layer horizontal position**" action. Parameter is new horizontal/left position on target display.*
	**Set layer vertical position**	*Inserts "**set layer vertical position**" action. Parameter is new vertical/top position on target display.*
Objects		
	**Set object focus**	*Inserts "**set object focus**" action. Parameter is name of object to receive focus.*
	**Change page focus**	*Inserts "**change page focus**" action.*
System		
	**Set object focus**	*Inserts "**send key**" action. Parameter is key code to send.*
	**Sleep**	*Inserts "**sleep**" action. Parameter is number of milliseconds to remain inactive.*
	**Exit and load project**	*Inserts "**exit and load project**" action. Parameter is address of new project to load.*

3.3.8. Target



Picture 28: Target Ribbon

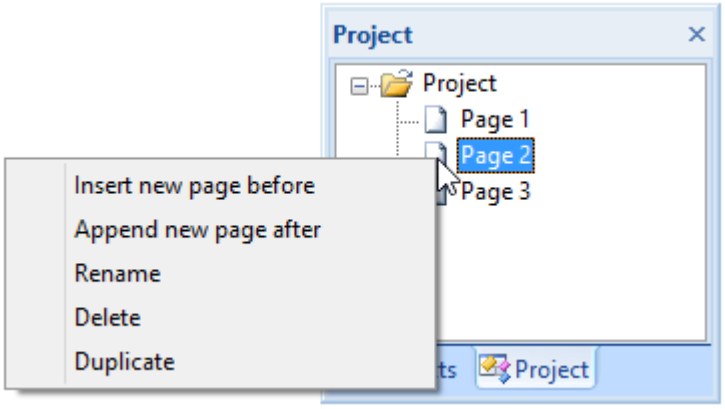
Table 9: Target ribbon groups

Properties		
	Pre-multiply alpha	*Affects displaying of images on target device, and should be set according to target capabilities (generally: do not set on PC, set for embedded devices).*
	Compressed images	*Makes compiled project smaller and faster to transfer, but also makes it longer to compile. On some targets, it may make project slower to load.*
	Navigation with keys	*Enable navigation among objects for target device that has a physical keyboard.*
	Bit depth	*Set to value that corresponds to capabilities of your target device. HoC-Editor dithers images for lower bit depth to avoid loss of quality.*
	Text anti-aliasing	*One of: none (fastest, but no anti-aliasing done), normal (smoothed edges), ClearType (further optimized for LCD screens, currently not supported).*
	Native text encoding	*Defines real encoding type for string tags with virtual tag type "native-encoding".*
Licenses		
	Licenses	*Specifies licensed target/device that project is compiled for.*
Download		
	Generate binary	*Compiles project, if needed, and saves it to a file.*
	Tag memory size	*Defines size of memory tag area on target device. Used just for "out of memory" warning indication in Tags window.*
Simulate		
	Play	*Compiles project, if needed, and runs it on PC version of IQ-Engine.*

3.4. Docking Windows/Panes

We have already explained the process of docking & un-docking windows/panes. In further text, we will describe the functionality for each of them.

3.4.1. Project



Picture 29: Project window / pane with context menu opened

Project window lists all pages in the project and provides various operations with pages.

Right click on ***Project item*** will open context menu where you can insert new page before or after that page, delete it, duplicate it or rename it. When renaming, page name change automatically updates all page references in object properties and action scripts.

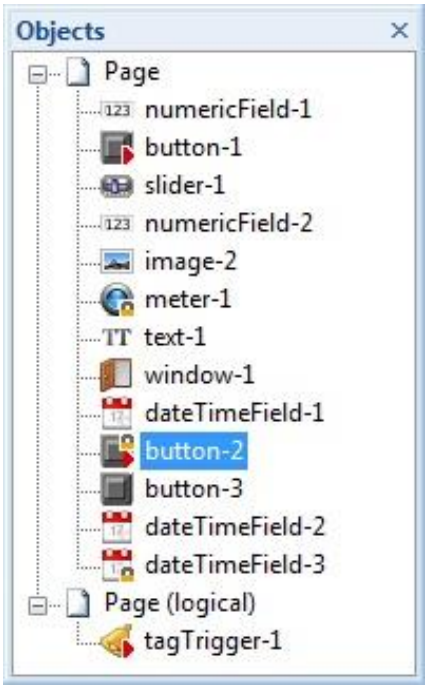
To change the ***page order*** inside a project, just press on any page item, drag & drop it on desired position. Page order is important for actions like ***'Go to next page***'.

Pages are distinguished by its type, which is shown by appropriate icon related to any page.

New page of any type can be inserted from ***Page ribbon,*** as well.

Page type can be changed in Properties window for Page properties (when none object is selected).

3.4.2. Object

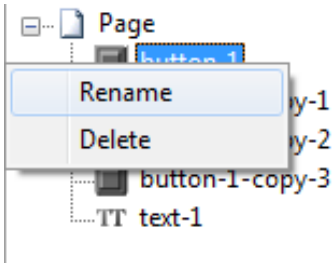


Objects window lists all objects on selected page and provides various operations with objects.

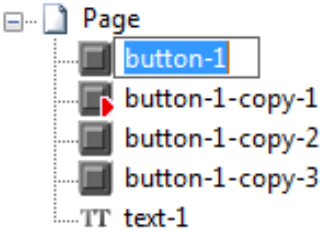
Table 10: Object actions

Action	How?
Select object	*Click on object item*
Multi-selected objects	*Click on object item while holding CTRL key to add object to or remove from the set of currently selected objects.*
Delete	*Right click on any object item to open popup menu*
Rename	
Change object order	*Drag & Drop in objects list*

Picture 30: Objects windows / pane



Picture 31: Renaming object



Picture 32: Renaming object

Object name change automatically updates all object references in object properties and action scripts.

All visual objects are placed within "Page" group. All logical objects, if any, are placed within "Page (logical)" group.

Object item selection change automatically reflects content of ***Properties window*** and ***Actions window***. When it is done in "Page" group additionally reflects selected objects in ***Working window***.

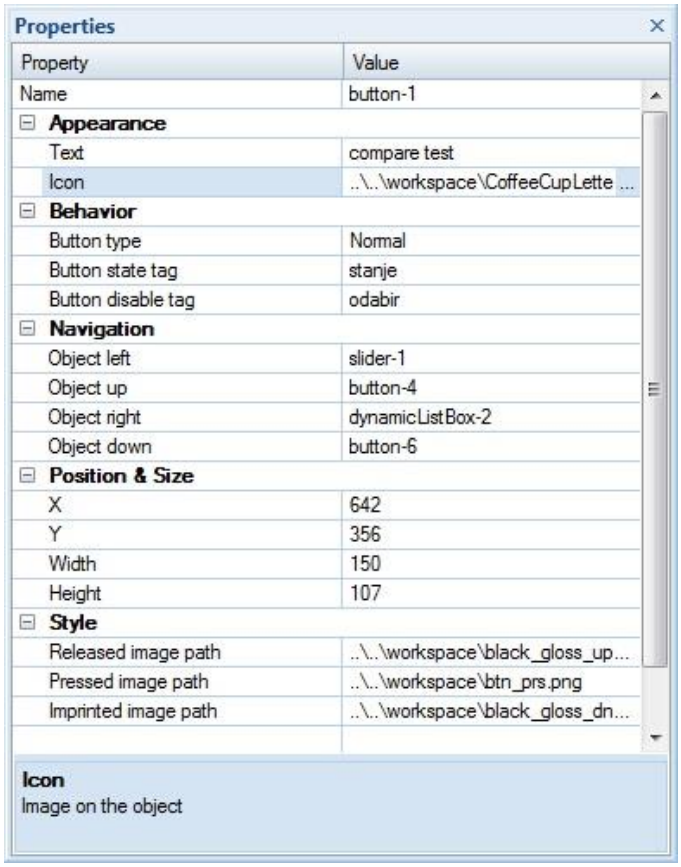
Object item icon is overlayed with "red arrow" mini-icon if object has actions and with "yellow padlock" mini-icon if object position or size is locked.

Note that "button-2" object on the Picture 30 (above) is both locked and has actions.

3.4.3. Properties

Properties window is the main window to change object appearance and behavior. It shows properties of currently selected object, or common properties of a group of selected objects. You can change property values of all visible properties.

All properties in ***Properties window*** are organized into meaningful groups indicated by bold names. These groups are:



Picture 33: Properties windows / pane

Table 11: Properties categories

Group	Description
Appearance	*Contains appearance properties specific for this object type.*
Behavior	*Contains behavior properties specific for this object type. Static objects don't have this group.*
Data	*Contains properties associated to tags.*
Scaling	*Contains 3 parameters for linear transformation of tag value, representing the value displayed or controlled with that object. See Modifying object chapter for more details.*

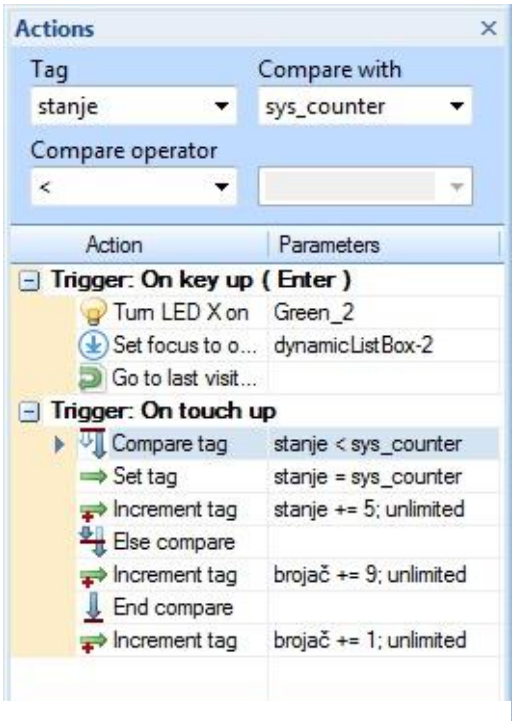
Navigation	*Is specific for control objects and contains directions for navigation among object using keyboard. This is especially useful for target devices without touch screen.*
Position & size	*Describes object position and size. You can move and resize object also here. This is not comfortable like in Working window, but can be useful in case of fine adjustment.*

Appearance of some object types is very complex and new groups are constituted in addition to Appearance group

Group	Description
Ticks & labels	*Is specific for objects with scales and describes appearance of scale ticks and labels.*
Grids	*Is specific for objects with grids and describes appearance of grid lines.*
Style	*Is an extension of Appearance group that is specific for some objects when Custom style is applied for that object. It allows detailed adjustment of object appearance.*

See Modifying objects chapter for more details about object styles.

3.4.4. Actions

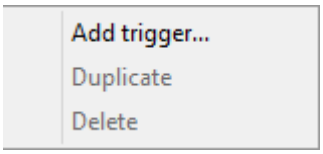


Picture 34: Actions window / pane

Actions window displays list of ***triggers*** on currently selected object and ***sequence of actions*** that are executed when a trigger is detected. If none object is selected, it displays list of global page triggers and associated actions.

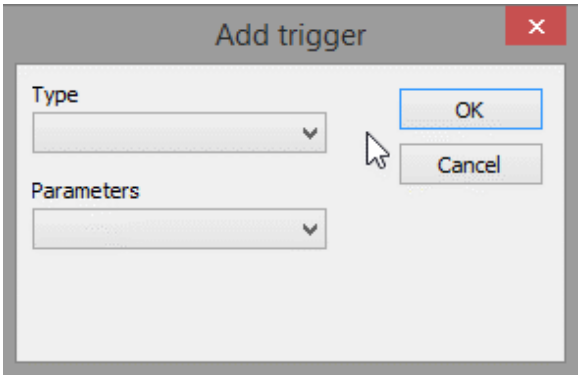
Click on desired action on Actions panel or double click on desired Actions list window item to insert new action into list of triggers and actions on selected object.

New action is added right after selected row. If none row is selected, it is added at the end of actions list.

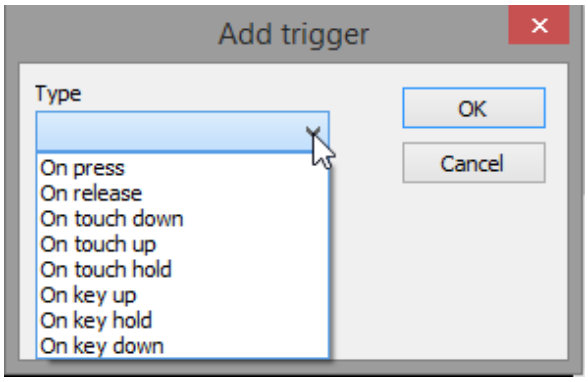


Right click on any action item to open popup menu. In popup menu you can add new trigger, but also delete or duplicate selected action.

Picture 35: Context menu



Picture 37: Add trigger



Picture 36: Add trigger | Types

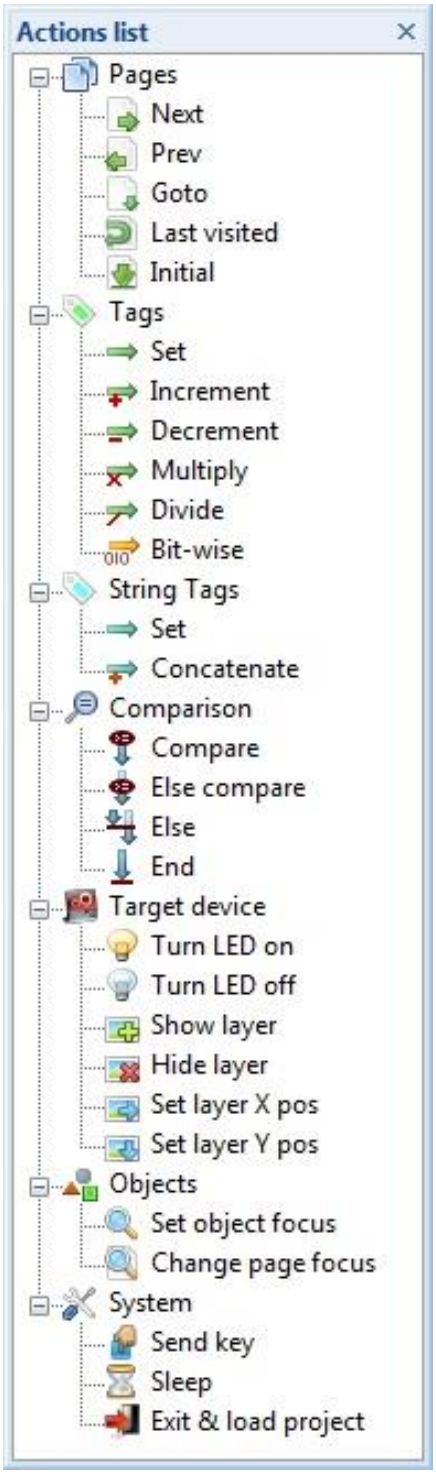
Press on any action item, drag it and drop it on desired position to change action position inside action sequence or even to move it to the action sequence of another trigger (if there is more than one trigger on selected object).

Click action item to select it. If selected action requires additional parameters, suitable edit/list boxes on top of Actions windows are automatically enabled, allowing argument selection from list (for tags, LEDs, pages, objects, etc.) or editing (for values).

Text above edit/list boxes is changing on selection of new action type, describing meaning of associated action parameters (Picture 34).

Parameters column on action items contains short symbolic description of action parameters.

3.4.5. Action List



Picture 38: Action List

Actions list window shows all available actions.

Double click on any action item to add it to an object.

The same functionality is provided by ***Actions panel***, but this window can be docked next to Actions window, providing multiple actions adding without a lot of mouse movement.

3.4.6. Tags

Tags window lists all tags in project.

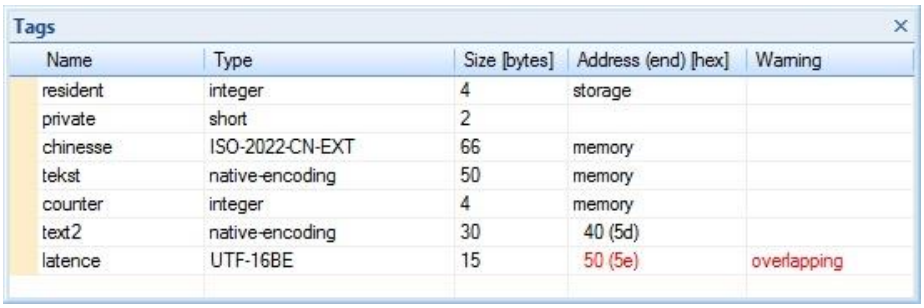
Action	Where	*Result*
Right click	**anywhere**	*Open popup menu, where you can add new tag into list of tags. After tag adding you are prompted to enter tag name into Name column*
	on any tag item	*Open popup menu. In popup menu you add new tag, but also delete, duplicate, merge or reorder selected tags.* ***Note** that you can also add tag in **Properties window** directly by editing suitable, tag-assigned object property.*
Click on tag	**Name column**	*Change name of existing tag. Tag name change automatically updates all tag references in object properties and action scripts.*
	Type column & select tag type from the list	*Change tag type.*
	Size column	*Change tag size. Tag size changing is allowed only for string tag types. Numeric tag types have predefined size.*
	Address column	*Change tag address.*
Click on	**Column header**	*For any of these columns to activate upward or downward sorting, or deactivate sorting for that column. **Name column** has the biggest priority. **Address column** has the lowest priority.*

If address is not stated, tag is managed as local tag. If address is 'store' (you can write just 's'), tag is managed as resident tag. If address is 'memory' (you can write just 'm'), tag is managed as auto-addressed memory tag.

If address is stated, tag is managed as fixed memory tag located on that address.

Warning column is not editable. It contains warning related to wrong tag address. Overlapping warning means that memory occupied by tag is already completely or partially used by another tag. Non-alignment warning means that tag address is not aligned. Numeric tags have to be aligned to their size. String tags have to be aligned to 2.

In case of warnings, compiling project may result in unexpected results!



Picture 39: Tags

Auto-addressing mechanism finds optimal addresses for auto-addressed memory tags. It fulfills gaps between fixed memory tags in the best way, respecting number and size of both gaps and auto-addressed tags.

It also generates memory tag report with fixed and calculated memory tag addresses:

```
   NAME  ADDRESS  ADDR  (DEC)  SIZE
   text        0             0    50
counter       32            50     4
  text2       40            64    30  fixed
latency       50            80    15  fixed  overlapping
Chinese       60            96    66
```

3.4.7. Output

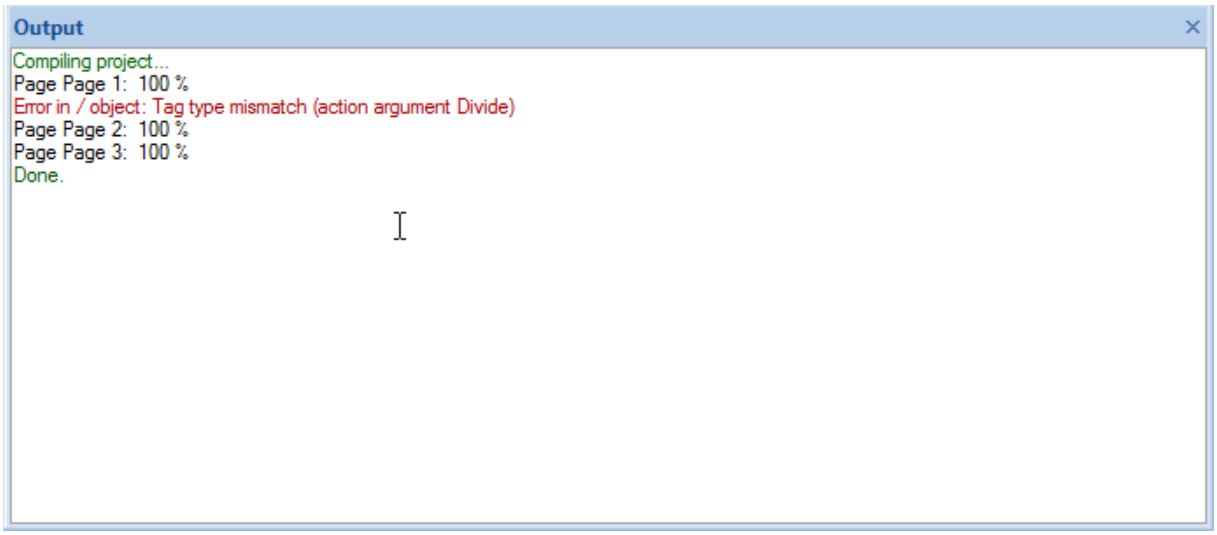
Output window displays progress and status messages from the application.

The most important progress to show is compilation progress, including compiler errors and warnings.

Double click on particular error or warning to select erroneous object.

If page containing that object is not selected, page is selected first. If page containing that object is not opened, page is opened and selected first.

Saving progress is also shown here.



Picture 40: Output

3.5. Objects

3.5.1. Button

Button executes actions when pressed.

Table 12: Button

Name	Name of the object.				
Appearance					
Text	*Text displayed on the object.*				
Icon	*Optional icon displayed on the object.*				
Icon size [%]	*Size of icon, in percent of button height.*				
Icon offset [%]	*Minimal icon distance from the left button border, in percent of button width.*				
Transparency	*Transparency of the object (255 is fully transparent)*				
Behavior					
Button type	*One of: normal, toggle.*				
Data (optional tag)					
Button state tag	*Reflecting button's pressed/released state.* [2]				
Button disable tag	*Reflecting whether button is enabled and can be pressed, or disabled.*				
Style	***Custom style only!***				
Released image path	*Relative path to image file used for background of*		*Released button.*		
Pressed image path			*Pressed button.*		
Imprinted image path			*Released button that remains imprinted.*		
Focused released image path			*Focused released button.*		
Focused imprinted image path			*Focused released button that remains imprinted.*		
Navigation					
Object left	*Defines object to receive focus when user wants to change focused object towards*		*left*	*side*	
Object up			*upper*		
Object right			*right*		
Object down			*bottom*		
Position and size					
X	*Position of the*	*left*	*side*	*of the object.*	
Y		*top*			
Width	*Width*				
Height	*Height*				

[2] ***Button*** *is control object that primarily writes value to its state tag, although reading is also allowed - external tag change will change button state.*

3.5.2. Color bar

Color bar shows tag values by cropping defined gradient color rectangle to corresponding length

Table 13: Color bar

Name	**Name of the object.**			
Appearance				
Gradient colors	*Defines color for minimum and maximum values; color can be equal (solid color).*			
Propagation from	*Defines which of four sides corresponds to minimum value.*			
Transparency	*Transparency of the object (255 is fully transparent)*			
Behavior				
Minimum value	*Value when rectangle is not visible at all.*			
Maximum value	*Value when rectangle fits complete color bar area.*			
Low alarm value	*Lower values will activate low alarm trigger.*			
High alarm value	*Higher values will activate high alarm trigger.*			
Data				
Value tag	*Tag associated with bar length.*			
Ticks and labels				
Show ticks	*Defines*	*whether ticks are shown.*		
Tick color		*color of ticks.*		
Tick width		*width of ticks, in pixels.*		
Tick size [%]		*length of ticks, in percent of bar half-width.*		
Tick alignment		*tick alignment within bar - top/left, bottom/right, center or dual.*		
Tick placement		*tick placement within knob - inner or outer.*		
Tick distance		*distance between neighboring ticks, respecting value range.*		
Label distance [ticks]		*distance between labels, in ticks.*		
Scaling				
A1	*Multiplier*		*for associated tag value.*	
A2	*Divider*			
B	*Corrector*			
Position and size				
X	*Position of the*	*left*	*side*	*of the object.*
Y		*top*		
Width	*Width*			
Height	*Height*			

3.5.3. DTC Field

DTC field displays Date, Time or Counter.

Table 14: Date, Time or Counter Field

Name	Name of the object.			
Appearance				
DTC choice	*One of: date, time or clock (respectively interpreting value tag as date, time in seconds and time in milliseconds).*			
DTC format	*Defines which components like minutes, seconds etc. to display.*			
Separator	*Separator between components.*			
Transparency	*Transparency of the object (255 is fully transparent)*			
Data				
Value tag	*Tag associated with time & date [3], expressed in number of seconds (starting from 1.1.1970. for date and time; starting from project start for counter)*			
Position and size				
X	*Position of the*	*left*	*side*	*of the object.*
Y		*top*		
Width	*Width*			
Height	*Height*			

3.5.4. Dynamic Image

Dynamic image shows one of images from list using tag value as index.

Table 15: Dynamic Image

Name	Name of the object. 1/2
Appearance	
Transparency	*Transparency of the object (255 is fully transparent)*
Behavior	
No of images	*Number of images this object contains.*
Images	*List of relative paths to image files.*
Data	
Value tag	*Tag used as index to determine which image from the list to display.*

[3] *Associating to* `'sys_time'` *system tag displays local time. Associating to certain local tag displays counter showing specific relative time (like stop watch).*

Name	Name of the object.			2/2
Position and size				
X	*Position of the*	*left*	*side*	*of the object.*
Y		*top*		
Width	*Width*			
Height	*Height*			

3.5.5. Dynamic List-box

Dynamic list-box changes tag value when user scrolls/rotates list of items vertically (by press and drag) and selects one of visible items from the list (by touch). Touched item is usually shown with different background. In difference to (static) list-box, text on items is dynamically updated from String tag. Therefore dynamic list-box may be considered also as a data object.

Table 16: Dynamic List-box

Name	Name of the object. 1/2
Appearance	
Maximum items	*Maximum allowed number of list items.*
Item size form	*The method of item size presentation - by visible items or by item height.*
Visible items	*Number of visible items (displayed when Item size form is by visible items).*
Item height	*Height of each single item (displayed when Item size form is by item height).*
Transparency	*Transparency of the object (255 is fully transparent)*
Behavior	
Scrolling type	*Scrolling or rotating. Rotating continues again with the first item after the last one. Both scrolling and rotating works only if number of list item is greater than number of visible items (items that fit into visible list-box area).*
Selection type	*On touch or fixed to center. On touch selects touched item. If selected from tag, takes care that selected item is visible. Fixed to center keeps selection in list-box center, so that is changed on scrolling instead on touch. If selected from tag, also takes care that selected item is always in list-box center.*
On release	*Slow down or stop immediately. Slow down continues with decelerated scrolling after you release touch. Initial scrolling speed depends on the sped of previous touch moving.*
Text list separator	*Separator used to separate items in list of list-box texts.*

Name	Name of the object. 2/2			
Data				
Value tag	*Tag associated with selected item in the list-box.* [4]			
Text list tag	*String tag associated with list of texts for each list item, separated by "text list separator" value. If this list is greater than maximum items, only maximum items are really taken in account.*			
Style	**Custom style only!**			
Item image	*Relative path to image file used for background of*		*non-selected item.*	
Selected image			*selected item.*	
Focused image			*selected item of focused list-box*	
Navigation				
Object left	*Defines object to receive focus when user wants to change focused object towards*		*left*	*side.*
Object up			*upper*	
Object right			*right*	
Object down			*bottom*	
Position and size				
X	*Position of the*	*left*	*side*	*of the object.*
Y		*top*		
Width	*Width*			
Height	*Height*			

3.5.6. Editable numeric field

123

Editable numeric field displays tag value and edits it when it is in editing mode.

Table 17: Editable numeric field

Name	Name of the object. . 1/2
Appearance	
No of digits	*Total number of digits to display.*
No of decimal digits	*Number of digits after decimal point.*
Leading zeros	*Activates displaying of leading zeros.*
Transparency	*Transparency of the object (255 is fully transparent)*
Data	
Value tag	*Tag associated with numeric value to display.*

[4] ***Dynamic list-box*** *is control object that primarily writes value to its selection tag, although reading is also allowed - external tag change will change selected item. If selected item is not visible, list-box is previously automatically scrolled to make selected item visible (to fit it within visible list-box area).*

Name	Name of the object. .				2/2
Scaling					
A1	*Multiplier for associated tag value.*				
A2	*Divider for associated tag value.*				
B	*Corrector for associated tag value.*				
Navigation					
Object left	*Defines object to receive focus when user wants to change focused object towards*			*left*	*side.*
Object up				*upper*	
Object right				*right*	
Object down				*bottom*	
Position and size					
X	*Position of the*	*left*	*side*	*of the object.*	
Y		*top*			
Width	*Width*				
Height	*Height*				

3.5.7. Editable text field

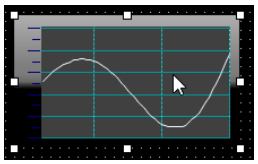
Dynamic text

Editable text field displays value of textual (string) tag. It also edits string tag value when field is in editing mode.

Table 18: Editable text field

Name	Name of the object.				
Appearance					
Transparency	*Transparency of the object (255 is fully transparent)*				
Data					
Value tag	*Tag associated with textual value to display.*				
Navigation					
Object left	*Defines object to receive focus when user wants to change focused object towards*			*left*	*Side.*
Object up				*upper*	
Object right				*right*	
Object down				*bottom*	
Position and size					
X	*Position of the*	*left*	*side*	*of the object.*	
Y		*right*			
Width	*Width*				
Height	*Height*				

3.5.8. Graph



Graph shows tag value through time in the 2D curve form.

Table 19: Graph

Name	Name of the object. .		1/2
Appearance			
Border width	*Width of the border.*		
Signal color	*Color used to draw signal curve.*		
Static signal type	*How signal curve is drawn; one of: line, dot.*		
Dynamic signal type	*How new and old values are handled when signal curve reaches right side of graph area; one of: overlapping (its drawing continues from left side), sliding (entire curve scrolls).*		
Transparency	*Transparency*	*of the object, excluding signal curve (100 is fully transparent)*	
Signal transparency		*of signal curve (255 is fully transparent)*	
Behavior			
No of samples in second	*Number of samples taken per second. Actual value may change slower or faster than that.*		
Sample distance [pixels]	*Distance between sample points, in pixels.*		
Minimum value	*Lowest allowed value. Curve point with lowest value is shown at the bottom of graph area.*		
Maximum value	*Highest allowed value. Curve point with highest value is shown at the top of graph area.*		
Low alarm value	*Lower values will activate low alarm trigger for actions.*		
High alarm value	*Higher values will activate high alarm trigger for actions.*		
Data			
Value tag	*Tag associated with graph value.*		
Ticks and labels			
Show ticks	*Defines*	*if ticks are shown.*	
Tick color		*color of ticks*	
Tick width		*tick width, in pixels.*	
Tick size [%]		*tick length, in percent of graph half-width.*	
Tick alignment		*tick alignment within graph - top/left, bottom/right or center.*	
Tick distance		*distance between neighboring ticks, respecting value range.*	
Label distance [ticks]		*distance between labels, in ticks.*	
Grids			
Show horizontal grids	*Defines*	*if horizontal*	*grid lines are shown.*
Show vertical grids		*if vertical*	
Grid style		*style of grid lines; one of: dotted line, full line.*	
Grid color		*color of grid lines.*	
Grid distance [seconds]		*distance between vertical grid lines, in seconds.*	

Name	Name of the object. 2/2			
Style	**Custom style only!**			
Background path	*Relative path to image file used for graph background.*			
Signal area color	*Color of graph signal area (where curve is drawn).*			
Coloring type	*Type of signal area coloring; fill, frame or none.*			
Scaling				
A1	*Multiplier*			*for associated tag value.*
A2	*Divider*			
B	*Corrector*			
Position and size				
X	*Position of the*	*left*	*side*	*of the object.*
Y		*top*		
Width	*Width*			
Height	*Height*			

3.5.9. Image

Image displays a static image.

Table 20: Image

Name	Name of the object.			
Appearance				
Image	*Relative path to image file (or absolute if image is on different disk drive than project file).*			
Transparency	*Transparency of the object (255 is fully transparent)*			
Position and size				
X	*Position of the*	*left*	*side*	*of the object.*
Y		*right*		
Width	*Width*			
Height	*Height*			

3.5.10. Image bar

Image bar shows tag values by cropping associated image to corresponding length.

Table 21: Image bar

Name	Name of the object.			
Appearance				
Bar fill image	*Relative path to image file used to draw the bar.*			
Propagation from	*Defines which of the possible four sides corresponds to minimum value.*			
Transparency	*Transparency of the object (255 is fully transparent)*			
Behavior				
Minimum value	*Value when associated image is not visible at all (completely cropped).*			
Maximum value	*Value when associated image is completely visible (fits complete image bar area).*			
Low alarm value	*Lower values will activate low alarm trigger.*			
High alarm value	*Higher values will activate high alarm trigger.*			
Data				
Value tag	*Tag associated with graph value.*			
Ticks and labels				
Show ticks	*Defines*	*if ticks are shown.*		
Tick color		*color of ticks*		
Tick width		*tick width, in pixels.*		
Tick size [%]		*tick length, in percent of graph half-width.*		
Tick alignment		*tick alignment within graph - top/left, bottom/right or center.*		
Tick placement		*tick placement within knob - inner or outer.*		
Tick distance	*Defines*	*distance between neighboring ticks, respecting value range.*		
Label distance [ticks]		*distance between labels, in ticks.*		
Scaling				
A1	*Multiplier*			*for associated tag value.*
A2	*Divider*			
B	*Corrector*			
Position and size				
X	*Position of the*	*left*	*side*	*of the object.*
Y		*top*		
Width	*Width*			
Height	*Height*			

3.5.11. Indicator button

Indicator button is "combination" of Button and Light. It executes actions when pressed, but can also indicate tag value.

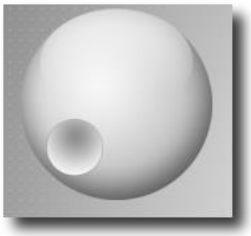
Table 22: Indicator button

Name	**Name of the object. .**		**1/2**
Appearance			
Text	*Text displayed on the object.*		
Icon	*Optional icon displayed on the object.*		
Icon size [%]	*Size of icon, in percent of button height.*		
Icon offset [%]	*Minimal icon distance from the left button border, in percent of button width.*		
Indicator size form	*The method of indicator size presentation - by percentage or by constant value.*		
Indicator size	*Indicator size (in form defined by Indicator size form).*		
Transparency	*Transparency of the object (255 is fully transparent)*		
Behavior			
Button type	*One of: normal, toggle.*		
No of color states	*Number of colors/states this object can display.*		
Indicator colors	*List of colors.*		
Data (optional tag)			
Value tag	*Tag used as index to determine color to display.*		
Button state tag	*Reflecting button's pressed/released state.* [5]		
Button disable tag	*Reflecting whether button is enabled and can be pressed, or disabled.*		
Style	***Custom style only!***		
Released image path	*Relative path to image file used for background of*	*released button.*	
Pressed image path		*pressed button.*	
Imprinted image path		*released button that remains imprinted.*	
Focused released image path		*focused released button.*	
Focused imprinted image path		*focused released button that remains imprinted.*	
Navigation			
Object left	*Defines object to receive focus when user wants to change focused object towards*	*left*	*side*
Object up		*upper*	
Object right		*right*	
Object down		*bottom*	

[5] *Indicator button is control object that primarily writes value to its state tag, although reading is also allowed - external tag change will change button state.*

Name	Name of the object. .			2/2
Position and size				
X	*Position of the*	*left*	*side*	*of the object.*
Y		*top*		
Width	*Width*			
Height	*Height*			

3.5.12. Knob



Knob changes tag value when user rotates its thumb. Thumb angle corresponds to tag value.

Table 23: Knob

Name	Name of the object. .		1/2
Appearance			
Start angle [degrees]	*Angle associated with minimum value.*		
End angle [degrees]	*Angle associated with maximum value.*		
Direction	*One of: clockwise, counterclockwise.*		
Thumb offset [%]	*Thumb offset from the knob rim, in percent of knob radius.*		
Thumb size [%]	*Thumb size, in percent of knob radius.*		
Transparency	*Transparency*	*of the object, excluding thumb (100 is fully transparent)*	
Thumb transparency		*of the thumb (255 is fully transparent)*	
Behavior			
Minimum value	*Value when thumb is*	*at start position.*	
Maximum value		*at end position.*	
Discrete value step	*Minimum difference to previous value for thumb to move; 0 to redraw thumb in for any change of value (analog knob).*		
Data			
Value tag	*Tag associated with thumb position.* [6]		
Ticks and labels			
Show ticks	*Defines*	*whether ticks are shown.*	
Tick color		*color of ticks*	
Tick width		*tick width, in pixels.*	
Tick size [%]		*tick length, in percent of knob radius.*	
Tick offset		*distance of ticks from the object rim, in percent of knob radius.*	
Tick placement		*tick placement within knob - inner or outer.*	
Tick distance		*distance between neighboring ticks, respecting value range.*	
Label distance [ticks]		*distance between labels, in ticks.*	

[6] *Knob is control object that primarily writes value to its value tag, although reading is also allowed - external tag change will change thumb position.*

Name	Name of the object. .				2/2
Style	**custom style only!**				
Background path	*Relative path to image file used for*			*knob background.*	
Thumb image path				*knob thumb background.*	
Focused background path				*focused knob background.*	
Navigation					
Object left	*Defines object to receive focus when user wants to change focused object towards*			*left*	*side*
Object up				*upper*	
Object right				*right*	
Object down				*bottom*	
Scaling					
A1	*Multiplier*			*for associated tag value.*	
A2	*Divider*				
B	*Corrector*				
Position and size					
X	*Position of the*	*left*	*side*	*of the object.*	
Y		*top*			
Width	*Width*				
Height	*Height*				

3.5.13. Light

Light indicates tag value.

Table 24: Light

Name	Name of the object.			
Appearance				
Text	*Text displayed on the object.*			
Icon	*Optional icon displayed on the object.*			
Transparency	*Transparency of the object (255 is fully transparent)*			
Behavior				
Nr of color states	*Number of colors/states this object can display.*			
Light colors	*List of colors.*			
Data (optional tag)				
Value tag	*Tag used as index to determine color to display.*			
Style	***Custom style only!***			
Background path	*Relative path to image file used for light background.*			
Position and size				
X	*Position of the*	*left*	*side*	*of the object.*
Y		*top*		
Width	*Width*			
Height	*Height*			

3.5.14. List-box

List-box changes tag value when user scrolls/rotates list of items vertically (by press and drag) and selects one of visible items from the list (by touch). Touched item is usually shown with different background.

Text on items are predefined (static) for each list item.

Table 25: List-box

Name	Name of the object. 1/2
Appearance	
Maximum items	*Maximum allowed number of list items.*
Item size form	*The method of item size presentation - by visible items or by item height.*
Visible items	*Number of visible items (displayed when Item size form is by visible items).*
Item height	*Height of each single item (displayed when Item size form is by item height).*
Transparency	*Transparency of the object (255 is fully transparent)*
Behavior	
Scrolling type	*Scrolling or rotating. Rotating continues again with the first item after the last one. Both scrolling and rotating works only if number of list item is greater than number of visible items (items that fit into visible list-box area).*
Selection type	*On touch or fixed to center. On touch selects touched item. If selected from tag, takes care that selected item is visible. Fixed to center keeps selection in list-box center, so that is changed on scrolling instead on touch. If selected from tag, also takes care that selected item is always in list-box center.*
On release	*Slow down or stop immediately. Slow down continues with decelerated scrolling after you release touch. Initial scrolling speed depends on the sped of previous touch moving.*
Item titles	*List of texts for each list item. If this list is greater than maximum items, only maximum items is really taken in account.*
Data (optional tag)	
Value tag	*Tag associated with selected item in the list-box.* [7]

[7] ***List-box*** *is control object that primarily writes value to its selection tag, although reading is also allowed - external tag change will change selected item. If selected item is not visible, list-box is previously automatically scrolled to make selected item visible (to fit it within visible list-box area).*

Name	Name of the object. 2/2				
Style	***Custom style only!***				
Item image	*Relative path to image file used for background of*			*non-selected item.*	
Selected image				*selected item.*	
Focused image				*selected item of focused list-box.*	
Navigation					
Object left	*Defines object to receive focus when user wants to change focused object towards*			*left*	*side*
Object up				*upper*	
Object right				*right*	
Object down				*bottom*	
Position and size					
X	*Position of the*	*left*	*side*	*of the object.*	
Y		*top*			
Width	*Width*				
Height	*Height*				

3.5.15. Message field

Message field

Message field displays one of predefined texts using tag value as index.

Table 26: Message filed

Name	Name of the object.			
Appearance				
Line alignment	*Vertical text alignment.* [8]			
Transparency	*Transparency of the object (255 is fully transparent)*			
Behavior				
No of text messages states	*Number of text messages this object contains.*			
Text messages	*List of text messages.*			
Data (optional tag)				
Value tag	*Tag used as index to determine text to display.*			
Position and size				
X	*Position of the*	*left*	*side*	*of the object.*
Y		*top*		
Width	*Width*			
Height	*Height*			

[8] *Note that horizontal text alignment and other standard text formatting properties are rather adjusted on **Text formatting** group of Home panel and Format panel than using standard object properties.*

3.5.16. Meter

Meter indicates value of a tag by rotating the needle to corresponding position.

Table 27: Meter

Name	Name of the object. .		1/2
Appearance			
Start angle [degrees]	*Angle associated with*		*minimum value.*
End angle [degrees]			*maximum value.*
Direction	*One of:*		*clockwise or counterclockwise.*
Needle type			*image or shape (triangle).*
Needle size [%]	*Length of the needle in percent of meter radius*		
Transparency	*Transparency of*	*the object, excluding needle (100 is fully transparent)*	
Needle transparency		*the needle (255 is fully transparent)*	
Behavior			
Minimum value	*Value when rectangle*		*is not visible at all.*
Maximum value			*fits complete color bar area.*
Low alarm value	*Lower values will activate low alarm trigger.*		
High alarm value	*Higher values will activate high alarm trigger.*		
Discrete value step	*Minimum difference to previous value for needle to move; 0 to redraw needle for any change of value (analog meter).*		
Data			
Value tag	*Tag associated with needle position.*		
Ticks and labels			
Show ticks	*Defines*	*whether ticks are shown.*	
Tick color		*color of ticks.*	
Tick width		*width of ticks, in pixels.*	
Tick size [%]		*length of ticks, in percent of meter radius.*	
Tick offset [%]		*distance of ticks from the object rim, in percent of meter radius.*	
Tick alignment		*tick alignment within meter - top/left, bottom/right or center.*	
Tick distance		*distance between neighboring ticks, respecting value range.*	
Label distance [ticks]		*distance between labels, in ticks.*	

Name	Name of the object. 2/2			
Style	**Custom style only!**			
Background path	*Relative path to image file used for meter*			*background.*
Needle image path				*needle background. (for image needle type only!)*
Needle rotation X offset	*X*	*offset from meter center to meter needle rotation center.*		
Needle rotation Y offset	*Y*			
Needle color	*Color of meter needle. (for shape needle type only!)*			
Needle base size [%]	*Base*	*size of meter needle in percent. (for image needle type only!)*		
Needle rotation offset [%]	*Tail*			
Scaling				
A1	*Multiplier*	*for associated tag value.*		
A2	*Divider*			
B	*Corrector*			
Position and size				
X	*Position of the*	*left*	*side*	*of the object.*
Y		*top*		
Width	*Width*			
Height	*Height*			

3.5.17. Numeric field

123

Numeric field displays tag value.

Table 28: Numeric field

Name	Name of the object.			
Appearance				
No of digits	*Total number of digits to display.*			
No of decimal digits	*Number of digits after decimal point.*			
Leading zeros	*Activates displaying of leading zeros.*			
Transparency	*Transparency of the object (255 is fully transparent)*			
Data (optional tag)				
Value tag	*Tag associated with numeric value to display.*			
Scaling				
A1	*Multiplier*	*for associated tag value.*		
A2	*Divider*			
B	*Corrector*			
Position and size				
X	*Position of the*	*left*	*side*	*of the object.*
Y		*top*		
Width	*Width*			
Height	*Height*			

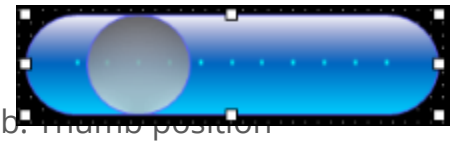
3.5.18. Page

Object representation of page (selected when neither one object on page is selected).

Table 29: Page

Name	Name of the object.	
Page		
Background color	*Page background*	*page color. Valid if page background image is not defined. Also adjustable from Page panel (Properties group).*
Background transparency		*transparency. Use maximum value (255) to make page background invisible. Also adjustable from Page panel.*
Background image		*image*
Page type	*Page type. If new page type is singular one, previous singular page of that type is automatically changed to normal page.*	
Initial focus	*Initial object focus for navigation among objects, set every time when this page is entered.*	

3.5.19. Slider



Slider changes tag value when user moves its thumb. Thumb position corresponds to tag value.

Table 30: Slider

Names	Name of the object. . 1/2	
Appearance		
Propagation from	*Specifies which of four sides of the object corresponds to minimum value.*	
Channel width	*Width of "channel" at the middle of the slider that "directs" thumb's movement.*	
Channel offset	*Offset of the "slider movement channel" from borders of the slider.*	
Thumb size [%]	*Size of sliding thumb, in percent of slider body width.* Can be greater than 100. In that case thumb size is equal to slider object width and slider body width is decreased accordingly.	
Transparency	*Transparency of*	*the object, excluding sliding thumb (100 is fully transparent)*
Thumb transparency		*sliding thumb (255 is fully transparent)*

Names	Name of the object. 2/2			
Behavior				
Minimum value	*Value when thumb is at*	*leftmost slider position.*		
Maximum value		*rightmost slider position.*		
Discrete value step	*Minimum difference to previous value for thumb to move; 0 to redraw thumb for any change of value (analog slider).*			
Data				
Value tag	*Tag associated with thumb position.* [9]			
Ticks and labels				
Show ticks	*Defines*	*whether ticks are shown.*		
Tick color		*color of ticks.*		
Tick width		*width of ticks, in pixels.*		
Tick size [%]		*length of ticks, in percent of slider half-width.*		
Tick offset [%]		*distance of ticks from center line of the object, in percent of slider half-width (not displayed for center alignment).*		
Tick alignment		*tick alignment within meter - top/left, bottom/right or center.*		
Tick placement		*tick placement within slider - inner or outer (not displayed for center alignment).*		
Tick distance		*distance between neighboring ticks, respecting value range.*		
Label distance [ticks]		*distance between labels, in ticks.*		
Style	**Custom style only!**			
Background path	*Relative path to image file used for slider*	*background.*		
Thumb image path		*Thumb background*		
Focused background path		*Focused slider background*		
Channel color	*Color of slider channel*			
Scaling				
A1	*Multiplier*	*for associated tag value.*		
A2	*Divider*			
B	*Corrector*			
Position and size				
X	*Position of the*	*left*	*side*	*of the object.*
Y		*top*		
Width	*Width*			
Height	*Height*			

[9] *Slider is control object that primarily writes value to its value tag, although reading is also allowed - external tag change will change thumb position.*

3.5.20. Text

Text

Text shows static text.

Name	Name of the object.			
Appearance				
Text	*Text displayed on the object.*			
Line alignment	*Vertical text alignment.* [10]			
Transparency	*Transparency of the object (255 is fully transparent)*			
Behavior				
No of color states	*Number of additional text color states. Base text color is defined within text formatting options.*			
Extra text colors	*List of additional text colors.*			
Data				
Value tag	*Tag used as index to determine text color to display.*			
Position and size				
X	*Position of the*	*left*	*side*	*of the object.*
Y		*top*		
Width	*Width*			
Height	*Height*			

3.5.21. Text field

Text field

Text field displays value of textual (string) tag.

Table 31: Text field

Name	Name of the object.			
Appearance				
Transparency	*Transparency of the object (255 is fully transparent)*			
Data (optional tag)				
Value tag	*String tag associated with textual value to display.*			
Position and size				
X	*Position of the*	*left*	*side*	*of the object.*
Y		*top*		
Width	*Width*			
Height	*Height*			

[10] *Note that horizontal text alignment and other standard text formatting properties are rather adjusted on **Text formatting** group of Home panel and Format panel than using standard object properties.*

3.5.22. Ticker

I am scrolling down the screen...

Ticker is a bitmap scrolling in one direction, usually used to display news, stock quotes and similar information.

Table 32: Ticker

Name	Name of the object.			
Appearance				
Image	*Relative path to image file used for ticker display content. Width is usually far greater than available space.*			
Transparency	*Transparency of the object (255 is fully transparent)*			
Behavior				
Scrolling type	*Scrolling or rotating. Scrolling re-starts image scrolling after image is completely displayed. Rotating glues the same image at the end (on the right side) of image after image is completely displayed. Rotating avoids image jumps in that way.*			
Speed pixels	*Speed of scrolling,*		*in pixels (each scroll is by that many pixels).*	
Speed interval [ms]			*in milliseconds (each scroll occurs that often).*	
Position and size				
X	*Position of the*	*left*	*side*	*of the object.*
Y		*top*		
Width	*Width*			
Height	*Height*			

3.5.23. Timer

Timer is logical (invisible) object that just that provides tag value modification within specified range, with specified step and speed.

Table 33: Timer

Name	Name of the object.
Timer	
Mode	*List of possible timer modes. Mode defines basic timer behavior.*
Start value	*Beginning timer value.*
Stop value	*Ending timer value. Timer stops when this value is reached in forward-stop and backward-stop modes.*
Step time [ms]	*Speed of changing current timer value in milliseconds.*
Data	
Value tag	*Tag containing current timer value.*

Timer disable tag	*Optional tag reflecting whether timer is enabled (and value tag is changing) or disabled.*

3.5.24. Trigger

Trigger is logical (invisible) object that provides triggering of action sequence of tag change.

Table 34: Trigger

Name	Name of the object.
Data	
Value tag	*Tag containing current timer value.*

4. Working with IQ-Editor

4.1. Projects

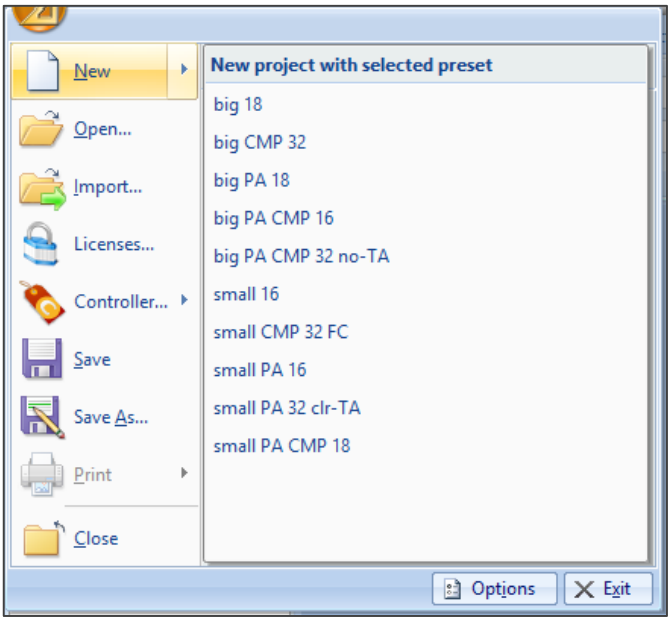
An **IQ Project** is most similar to a presentation or Visio project: it is a single file, which ***contains a list of pages***. Each page can contain various usable objects like text fields that indicate state of some manufacturing machine or illustrative static images.

4.1.1. Creating New Project

To start creating a new IQ Project, we need to click on Application button and ***New***, as we can see on Picture 41.

Picture 41: Creating a new project

Which will lead us to dialog window as follows on Picture 42.



Picture 42: New project presets

Command ***New*** creates a new empty project with default preset. This also adds one blank page to the project automatically.

In extension to command New, you can create a new empty project with selected preset.

Preset is a list of project properties specific for particular target device family. You have to choose preset during creating new project. Later you can change any preset property if necessary. Preset properties are placed on Target panel (Properties group).

Note that your supplier will provide you presets for all target devices that you are working with.

4.1.2. Opening Existing Project

To open an existing IQ-Project, we need to click on Application button and Open item in the main menu (Picture 43)

Picture 43: Opening an existing project

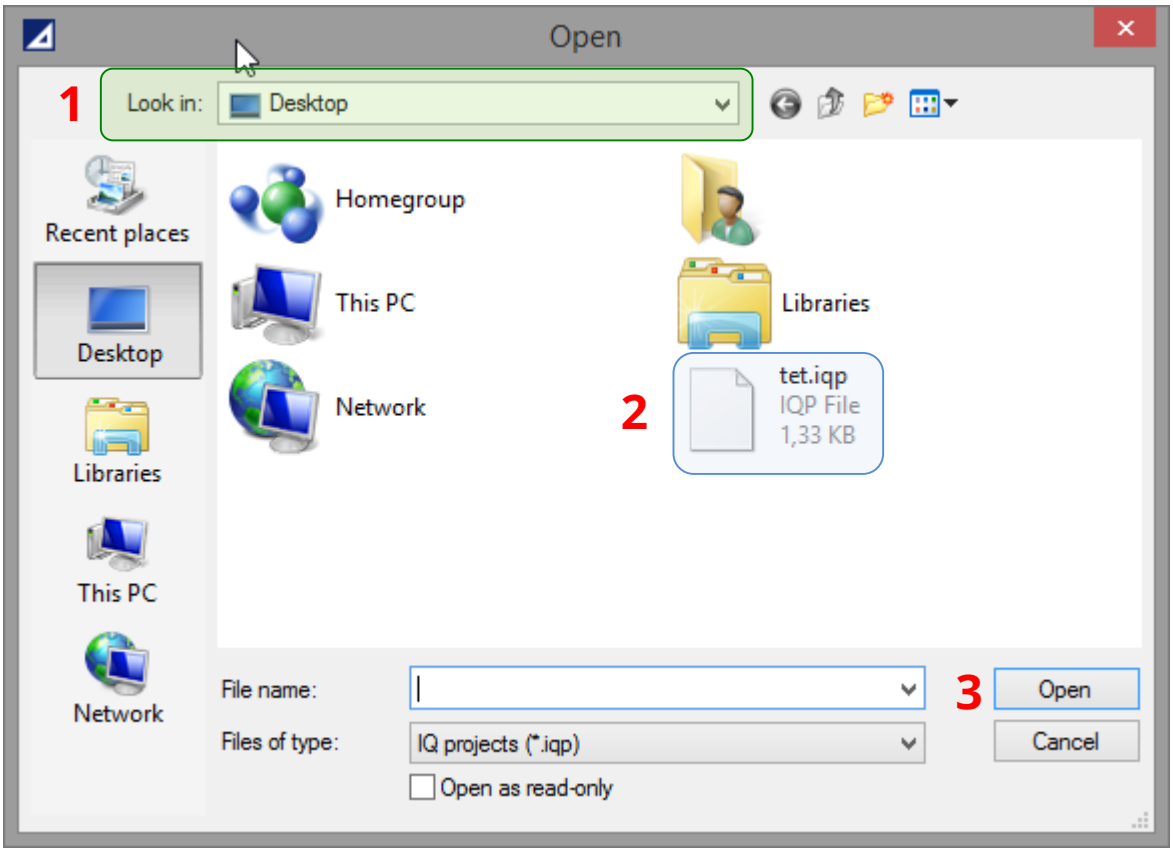
This will open a new, standard dialog window for opening files (Picture 44).

First, we need to choose the right folder (or drawer), where we keep our project(s) file(s) (**1**).

Then, we need to click on project file we want to open (**2**)

Finally, we confirm our choice by clicking on **Open** button (**3**)

That will open our project in the IQ-Editor.



Picture 44: Open file dialog window

4.1.3. Saving Project

When we want to save our project for later, we can use ***Save*** or ***Save As*** commands.

Picture 45: Saving the project

Save command saves changes or last version of an existing project, and ***Save As*** command allows us to save changes with a different name of project, so we can make a whole new project based on existing one.

If we are trying to save a newly created project, either ***Save*** or ***Save As*** command we choose, ***Save As*** will pops up because the project must have its name first.

Saving the project is similar to Opening a project; we need to choose a proper folder, then we need to specify the project name and instead of "***Open***", just click to "***Save***" or ***Save As***".

4.1.4. Merging Project

Often more designers have to work on the same large project. To provide that request, project merging has to be supported.

File panel command ***Import*** loads your existing project into currently opened one, providing project merging in that way.

Picture 46: Importing a new project | Merging

If importing project has different page size between imported and opened one, you are prompted to allow automatic page adjustment of imported project.

If both current and imported project have identical tags (the same name, type, size and address), we are prompted to choose between duplicating and merging of them.

- If we chose ***merging***, identical tags from both projects will be merged into one tag.
- If we chose ***duplication***, identical tags from imported project are automatically renamed by adding the nearest available ":{n}" suffix. *Tag name change automatically updates all tag references in object properties and action scripts.*

Note that we can merge identical tags afterwards. Just select group of tags in Tags window and all identical tags within that group will be merged. If ***tag_name*** and ***tag_name_2*** are merged, ***tag_name_2*** is automatically renamed to ***tag_name*** and name change updated in all tag references.

Handling described above is necessary due to tag conflict problem that often appears on project merging, especially if more designers develop projects starting from the same template that already includes set of predefined control tags. These predefined tags will appear as identical tags during project merging.

Besides tag conflict problem, page conflict problem also appears if pages from merged projects have the same name. In that case page names from imported project are automatically renamed by adding the nearest available " {n}" suffix. Page name change automatically updates all page references in object properties and action.

Page conflict problem also refers to page type for singular pages (like initial and global). If imported project has the singular page of the same type, it is automatically converted to normal page. New page type is assigned as "Normal ({original singular type})" and warning mark is added to the icon in Project window.

Merging projects provides also ability of using template pages. Particular wide-use page can be created as stand-alone project and saved as IQ template.

Opening IQ template automatically imports it into current IQ project.

If you want to open IQ template instead of importing it (in order to modify it), just close current IQ project first.

4.2. Pages

Project consists of pages. Pages are main IQ project presentation elements, consisting of visual and logical objects.

The supported types of pages are described previously in chapter 3.3.3 Pages.

When project is running on IQ-Engine, up to 3 pages are active at a time - global page, normal page on top of global page and window or dialog on top of normal page.

We also call them page layers.

When touch screen is used to handle control objects, situation with page layers is identical to the situation with just one layer.

When navigation with keys is used, however, situation is more complex because navigation is basically allowed within one page layer only. Therefore following rules are introduced in order to navigate efficiently among more page layers:

- Page layer is
  - ***navigational*** (focusable) only if initial focus is defined for that page
  - ***focused*** if key navigation currently navigates among objects on that page
- Only navigational page layer is allowed to be focused one
- Only one page layer is allowed to be focused at the time
- If both global and initial normal page are navigational, page focus is set to the global page on project loading
- Page focus can be changed from global to active normal page and vice-versa with "Change page focus" action
- When page focus is changed, object focus is set to the initial focus on the new focused page
- Page focus can't be changed when you are on window or dialog page; it is returned to the calling normal page after closing window/dialog
- If you use touch screen in parallel to key navigation, page focus can be changed to any page layer by touching control object belonging to that layer
- If you go to another normal page, page focus remains on current page layer (normal remain on normal and global remains on global); the exception is the case when page focus is on the normal page and another normal page is not navigational - focus is automatically switched to the global page in that case

Global page is opened under active ***normal page***. Visual objects on global page are visible only if active normal page background is disabled (by setting its transparency to the maximum value).

Note that visual objects on active normal page are always visible when dialog or window is opened because dialog and window are pages without background.

Project window lists all pages in the project and provides various operations with pages. You can create, delete, duplicate, rename and reorder pages here.

Home panel also contains operations with pages (group Page).

Page panel contains commands to insert page of any type and customize appearance of a single page.

When page is selected (neither one visual object on page is selected), ***Properties window*** shows page properties, including Page type property. Here you can change page type from the list of all available page types.

If you change page type into singular page type, previous singular page of that type is automatically changed to normal page.

Page tools from ribbon menus are already explained in chapter 3.3.3. Page.

4.3. Objects

Page consists of objects. Objects are the smallest presentation elements.

Object is defined on page with its attributes:

- *position,*
- *size,*
- *transparency and*
- *Z-order.*

Z-order is important for overlapping objects, telling which one is above another one.

Objects are functional units. Each type of object has its own unique function, as described in chapter ***3.5. Objects***

We distinguish 2 main groups of objects; ***visual*** and ***logical*** objects.

Logical objects provide specific functionality, but user can't see them on the page.

To add a new logical object to page, just click on desired type of object on ***Insert panel***. Logical objects are not visible and their position and size on the page don't need to be defined.

Visual objects are furthermore divided in 4 groups of objects:

- *Static, Dynamic, Data and Control objects.*

Data objects are objects with dynamic content, usually provided by tags.

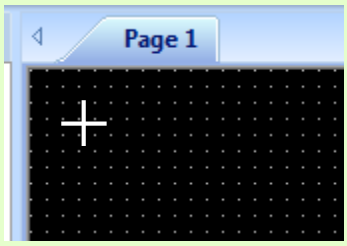
Control objects are changing primarily on user event like screen touch.

Objects window lists all objects on selected page and provides various operations with objects. You can select one or more objects, delete and rename them.

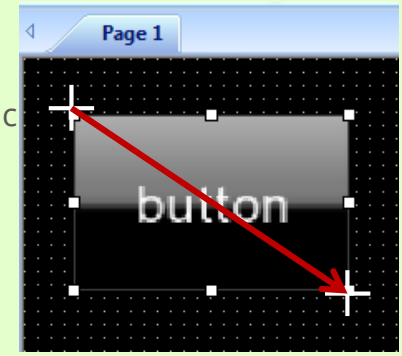
To add a new visual object to page, click on desired type of object on ***Insert panel*** and draw it on the page.

Example: Adding an object

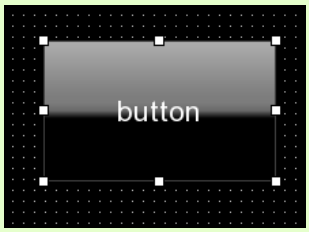
Drag the cursor into the working window. It will form a crosshair.



Click and drag diagonally to form a rectangle. The button will be drawn dynamically within the rectangle



Release the mouse button to finish the object



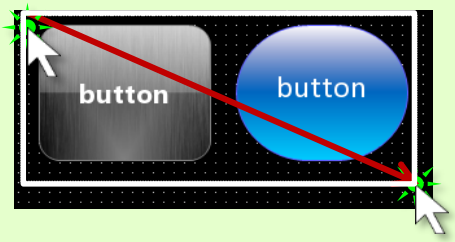
You can also select one or more objects, move, resize, reorder, copy, paste, delete, duplicate, lock, align, group and ungroup them.

Example: Multi-selection of objects

The Working window allows selection and manipulation of multiple objects.
Multi-selection can be performed by:

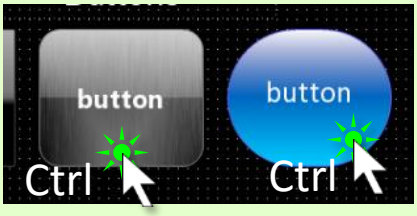
Select the right two buttons with any multi-selection method.

- *Dragging a selection rectangle over the objects in the working window*
  - *The selection rectangle must completely encircle the objects*

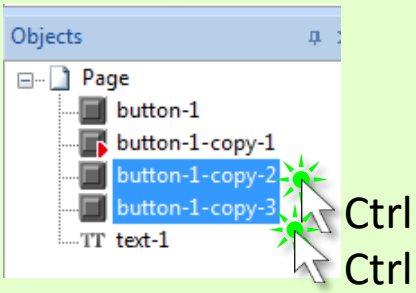


- *Clicking on the objects in the working window while holding the **Ctrl** key*

C



- *Clicking on the objects in the objects window while holding the **Ctrl** key*



- *Multi-selection objects form a group with a single bounding rectangle.*
- *The individual objects' size and position within the bounding rectangle scales accordingly.*

Note that any object selection change done in ***Working window*** automatically reflects selected object items in ***Objects window***; and vice-versa.

Objects in IQ projects have scalable object orientation. It means that they hold their general look on scaling.

In order to respect scalable orientation, all properties that represent fixed size of particular object components are changed to represent the percentage of complete object size.

IQ-Editor takes care about compatibility with old-style objects when old projects are loaded.

The only non-scalable object component is text, because design practice prefers manual font size adjustment. However, you can select any number of objects with text and decrease or increase their font size at once.

Scalable object orientation is especially useful when resizing page, which applies to all pages in project. In other words, page resizing scales all objects in project.

4.3.1. Modifying Objects

Properties window is the main window to change object appearance and behavior (Picture 33). It shows properties of currently selected object, or common properties of a group of selected objects. You can change property values of all visible properties.

Another important window to change behavior is ***Actions window*** (Picture 34). It displays list of triggers and actions of currently selected object. If nothing is selected, it displays page triggers and actions. Note that not all objects support actions.

All properties in ***Properties window*** are organized into meaningful groups indicated by bold names. One of groups is Scaling and appears for all objects that have numeric tag, which usually represents the value displayed or controlled with that object.

3 scaling parameter describes linear transformation from tag value to displayed/controlled value according to standard formula `y = A1/A2 * x + B`.

Text is specific part of many objects and its appearance is rather adjusted on Text formatting group of Home panel and Format panel than using standard object properties in Properties window.

Text formatting includes font family and size, style, text color and justification.

Styles provide consistent and professional look to objects within one page or the whole project.

If neither one style fits your requirement for particular object, you can use so-called ***Custom style*** that imports additional properties to ***Properties window***, allowing detailed adjustment of object appearance.

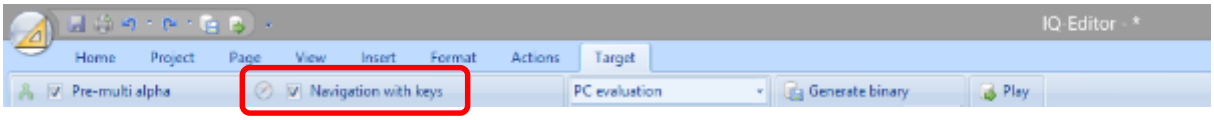
4.3.2. Navigation among the objects

Control objects are changing primarily on user event like screen touch.

Sometime is, however, more natural to use ***keys*** to change control objects. In many case target device even doesn't have touch screen and navigation with keys among object is the only option.

Navigation with keys is based on ***object focusing***. Focused object is the only object that receives user key events. Only control objects can be focused and only one control object on the page can be focused at the time.

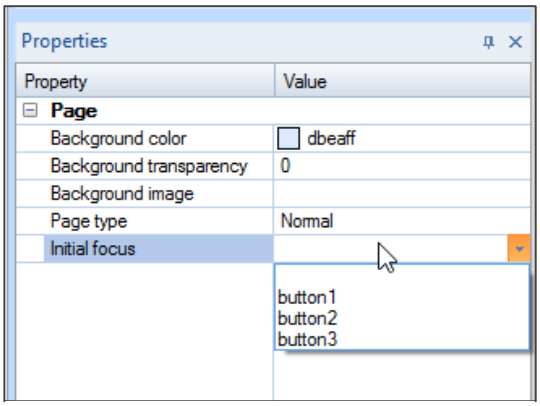
Navigation with keys can be enabled/disabled in Target panel.



Picture 47: Enabling| disabling navigation with keys

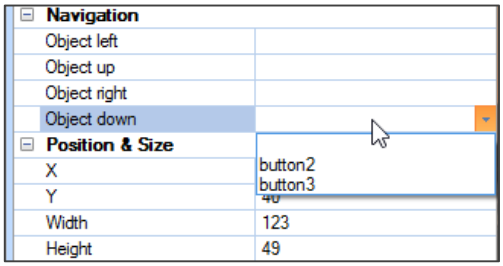
Initial focus defines first object that is focused on entering new page.

Initial focus is the property of page, which is settable in ***Properties window*** when none object is selected (Picture 48).



Picture 48: Initial focus

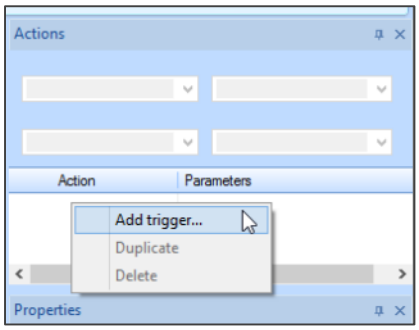
Navigation group of any control object consists of 4 properties that defines which control object will be focused when this object is currently focused and receives one of 4 possible arrow key events.



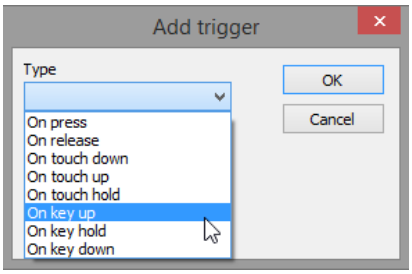
Picture 49: Navigation

Each control object has 3 additional specific triggers: ***on key up, on key hold*** and ***on key down***. (Picture 49). They start sequence of actions for focused object when key ***up***, ***hold*** and ***down*** events are received, respectively.

Key hold event behavior is additionally described with ***Actions group*** on ***Project ribbon***.



Picture 50:
Adding trigger | Right-click on Action area



Picture 51:
Add trigger window

If action trigger and navigation instruction within ***Navigation group*** are assigned to the same arrow key, action sequence execution has higher priority and overrides navigation.

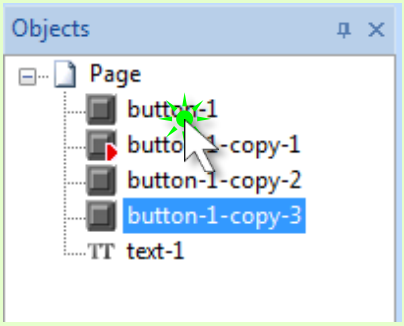
Action Set object focus can be used also for navigation. This brings additional flexibility in navigation among objects.

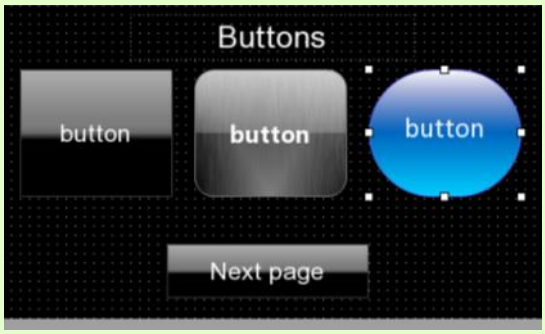
Focused object is visually assigned in different ways, depending on control object type:

- **button**: special image for release state
- **toggle button**: additionally to button, special image for imprinted state
- **slider & knob**: special image for background
- **list box**: special image for selected item
- **editable field types**: thin rectangle around field (see picture below!)

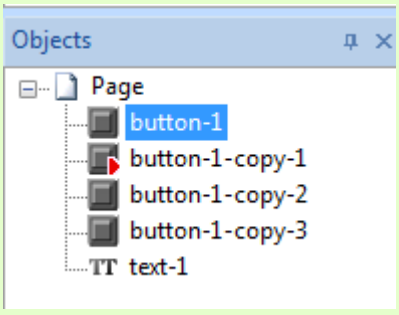
Special "focus" image has to be defined for custom object style. Predefined styles has this image already integrated, like for all other images.

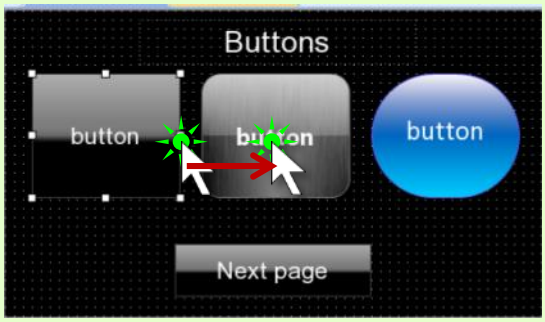
Example





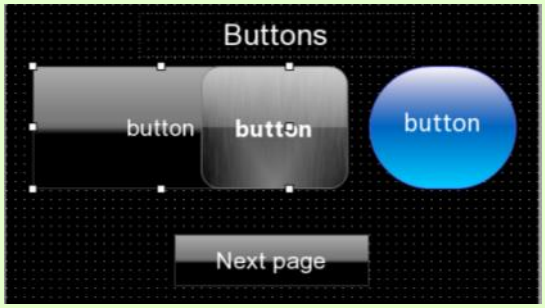
By clicking on the object name, the object gets selected in the working window
Same thing is accomplished by clicking the object in the working window

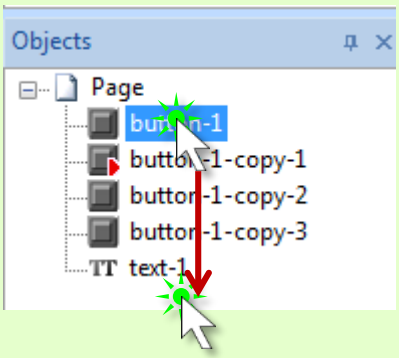


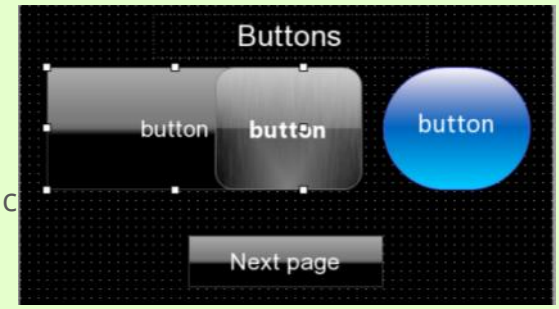


Now, select the right-middle reticle of the button and drag it to the right over the center button

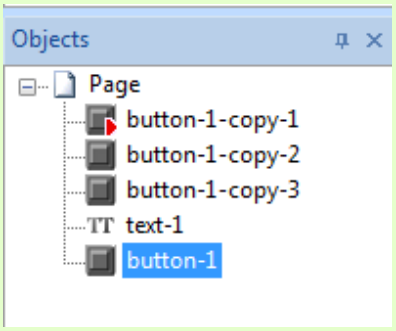
The button scales automatically, but is rendered behind the middle button

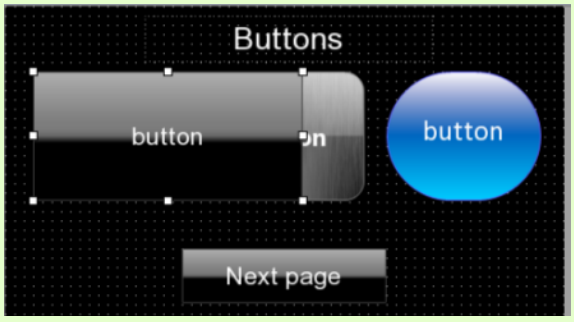






Now, click the Button-1 name in the objects window and drag it to the bottom
This will change the Z-order and move the Button-1 in the front





The object Button-1 is then rendered over the middle button

4.4. Tags

IQ-Engine task is to provide ***supervision*** and ***control*** of external devices connected to target device, using graphical user friendly interface ***designed by IQ-Editor.***

Tags are symbolic names associated to memory locations on target device, usually presenting states and values on external devices connected to target device.

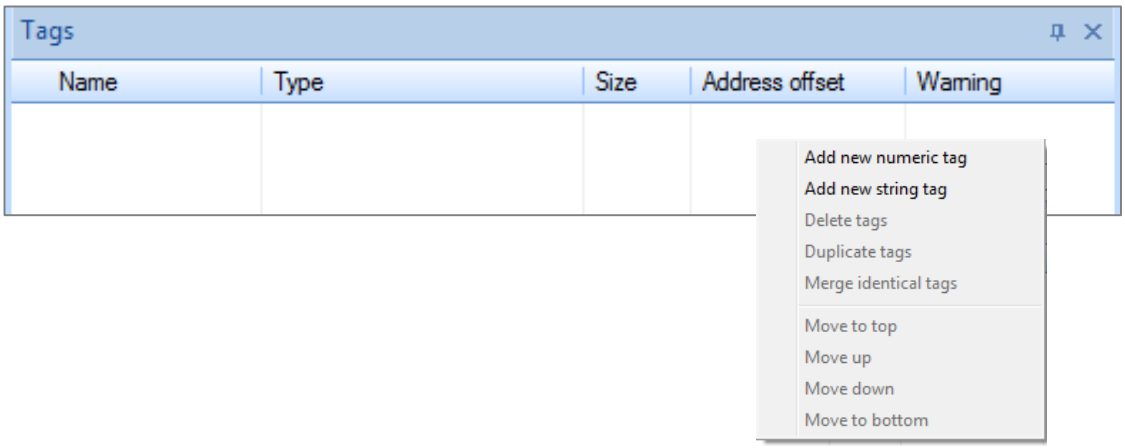
Therefore tags significantly simplify interaction between IQ-Engine running your IQ project and external devices through memory locations on target device.

For instance, it is much easier to create tag `'voltage_main_battery'` *that is associated to suitable memory location and format (e.g. 0x24005620, 2 bytes, unsigned, big endian order) than specifying it separately for each object or action that requires* it.

A usual procedure in IQ project designing is to ***create list of required tags first***. This part has to be done by someone who knows target device well and arrangement of external device data within its memory.

4.4.1. Adding Tags

Tags are primarily adding in ***Tags window***, but you can also add it directly by editing suitable, tag-assigned object property.



Picture 52: Adding tags

To add a tag – right click within the tag window to open the context menu.

IQ-Engine currently supports:

- **2 numeric tag** types with fixed size;
  - 2-byte short and
  - 4-byte integer and
- **90 string tag types** with variable size (90 different text encodings).

The added tag can be used as a variable in the UI, which can interconnect multiple objects and their behavior

It can also serve as a holder for system data to be visualized or controlled

The second step is designing pages and objects, and referencing created tags to tag-assigned object properties or action arguments.

Native-encoding is virtual string tag type that represents one of 90 real string tag types, as specified with Native text encoding on Target panel.

Tags are also distinguished by controllers that are driving them.

Currently supported controllers are local, resident and generic memory controller. Associated tags are respectively called local, resident and memory tags.

Tags architecture supposes that in the future each specific external device attached to target platform will be provided with its own memory controller. IQ-Editor will be able to load any memory controller found in Controllers folder (in DLL form).

Local and resident tags, in difference to memory tags, don't have associated memory location. IQ-Engine manages them in its local memory.

Resident tags holds their value even after target device shutdown and restart.

System tags are local tags with predefined functions. They present various counters, real time clock, current page, project statistics, etc.

Large group of actions is dedicated to operations with tags.

String tag actions take care about proper text conversion when tag is associated to tag with different text encoding.

Tag input control takes care about entering allowed tags only as object properties and action arguments.

System tags are read-only tags. Therefore, they are not allowed as tag value of control objects (*like Button, Slider, Knob, List-box and Timer*) and first action argument (representing tag where action result is stored).

String tags are not allowed in numeric object properties and numeric tag actions; and vice-versa.

Tag input control can't control use of proper tag types in the case when you change tag type afterwards. In that case, however, warnings about invalid tag type are reported during project compilation in Output window.

Table 35: System tags

System Tag	Description
`sys_counter`	*Incremental counter **+1/s***
`sys_EKG`	*Generator of EKG signal for a Medical Example*
`sys_time`	*Current system time*
`sys_page_current`	*Current page*
`sys_memory_current_bytes`	*Currently allocated memory. Useful for memory usage testing.*
`sys_memory_num_allocs`	*Number of memory allocations. As above, it is useful for memory usage testing.*

4.5. Actions & Triggers

Actions describe sequence of acts that ***IQ-Engine*** automatically performs on specified events or after detecting specified conditions.

***For example**, if water level in reservoir becomes too high, IQ-Engine automatically shows an alarm. Water level is presented by tag.*

Specified events or conditions that start actions are called ***triggers***.

***Example** of simple trigger is button press. A physical key press is little bit more complex because it requires an argument – key code.*

Triggers and actions are associated either to objects or to page - if none object is selected.

Some triggers are specific only for certain object types or page.

***For example**, triggers on key events are specific only for control objects (because they are focus-able) and page (when neither one object is focused).*

Triggers on alarms are specific only for objects with low/high alarm value, like Meter or Graph.

The sequence of actions started on trigger is executed sequentially, as actions are listed in ***Actions window*** - the main window for managing actions.

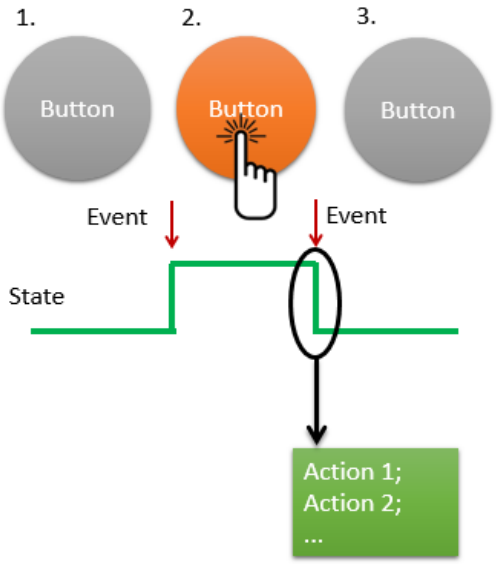
A button is a basic object able to react to the user's input

The user can trigger an action by pressing and releasing a button

The event which is the cause of the activity is called a ***Trigger***

Actions are activities which execute on the Trigger event

This is the basic model for handling interaction in the IQ-Engine



Picture 53: Button scheme

4.5.1. Adding Actions

To respond to user input, actions are added to the objects

Actions can be used to:

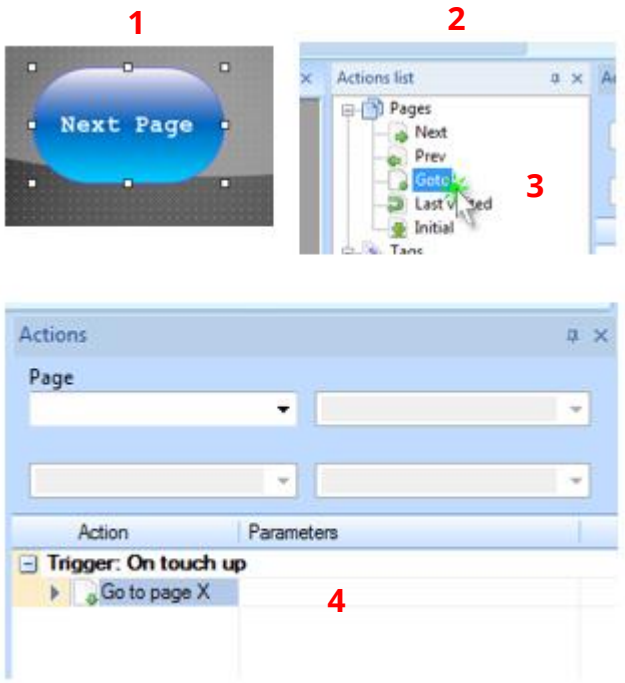
- Navigate the UI
- Manipulate data
- Control Hardware

Actions are added to objects through the ***Actions window*** and the ***Actions List*** window (which is the palette of available actions)

Tip*: In case there is no screen space to accommodate the Actions list window, the Actions ribbon can be used instead*

Let us now add an action to our button (Picture 54)

1. *Select the Button-1*
2. *Move the mouse to the Action List menu*
3. *Double-click the Pages/Go-to action*
4. A new trigger and action will appear in the actions window



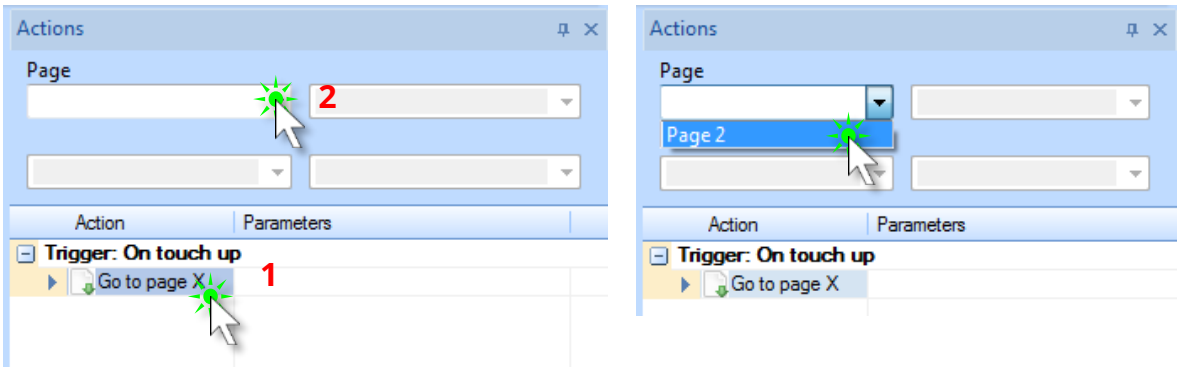
Picture 54: Adding actions

4.5.2. Editing Actions

The actions editor has two parts:

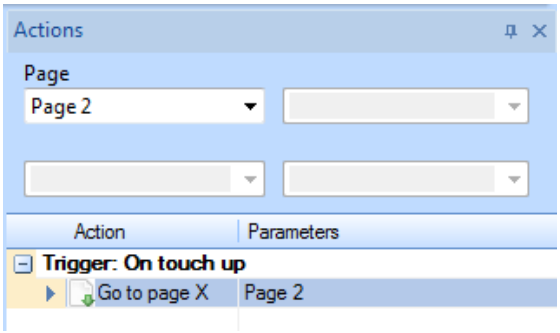
1. **The actions list**
   - Contains a list of triggers and actions
2. **The action editing pane** (top of the window)
   - The action editing pane contains 4 combo boxes which are enabled and change function based on the currently selected action

1. Click on the „**Go to page X**" action (Picture 55: Editing Actions)
   - For the „Go-to" action, the Top-left Combo box contains the list of available pages (not including the originating page)
2. Click on the combo box and select the page „**Page 2**", added earlier



Picture 55: Editing Actions

- The selected parameters are listed in the „Parameters" column
  - *If an action has no defined parameters, the IQ-Editor will report an error during the project compile*
- The action is listed under its default trigger – „On touch up"
  - *The action will execute after the button is released*
- Other triggers can be added for various events on the available objects
  - *All triggers on a button can hold actions simultaneously, and each trigger will execute its respective actions*
- A trigger's list of actions is executed sequentially in the order as they are listed in the editor



Picture 56: Edited actions

The majority of actions are executed instantly. Example is ***tag value*** setting.

Exceptions are so-called global actions. They are global because of their potential influence to the whole project. Their execution is suspended until all standard actions are executed and all other active processes finished. In other words, they are executed on IQ-Engine idle state.

***Example** of actions:*

Trigger = button release (object trigger on button 'button_1')

*1. Set tag **'abc'** value to 123*

2. Exit current and load new project 'second project'

3. Set object focus to button 'button_2'

Note that action 2 is global action that is executed after action 3 and other active processes finishes.

4.6. Compiling & Transferring Projects

When you want to see how your project works in practice, just go to ***Target panel*** and click on ***Play***.

IQ-Editor will start the following sequence:

1. ***compiles** your project into specially formatted file suitable for IQ-Engine,*
2. ***starts** PC version of **IQ-Engine** and*
3. ***load** compiled project file into IQ-Engine.*

Compilation progress is shown in Output window.

Once you are ready to run your IQ project on your target device, you have to download it. Just go to ***Target panel*** and click on ***Download***.

Don't forget to check first if your PC is properly connected to target device, and if IQ-Engine is already started on target device.

Also check if your preset properties (project properties specific for particular target device family) are correct.

Download option, however, is not supported yet. In the meantime you have to do following procedure:

- **Compile you project**.
  - Click on ***Save to file***,
  - Browse for the ***folder*** where you wish to ***keep*** your compiled projects, and
  - ***Enter project name.***
    (*IQ-Editor suggests you the same name as the name of IQ project.*)
- **Download compiled project** to your target device with a separate tool that usually comes together with you target device.
- **Start IQ-Editor** on target device and browse for downloaded compiled project.

Terms of use

Contact info

Mikroprojekt d.o.o.
Aleja Blaža Jurišića 9, HR-10040 Zagreb, Croatia
T/F: +385 1 2455 659
contact@mikroprojekt.hr
http://www.mikroprojekt.hr