

# Movicon NExT

## 5.0 Visualization Client

Ver.3.4.268



# Table of Contents

1.1. MOVICON.NEXT .....	1
<b>2. SCREENS.....</b>	<b>3</b>
2.1. PROJECT SCREENS.....	3
2.2. SCREEN TEMPLATES .....	4
2.3. SCREEN PROPERTIES .....	5
2.4. SCREEN NAVIGATION .....	13
2.5. EMBEDDED SCREENS .....	14
2.6. OBJECT BROWSER.....	15
2.7. SCREEN LAYOUT.....	16
2.8. AUTO-SHOW AND SWIPE HIDDEN SCREENS .....	17
2.9. AUTOLOADING SCREENS IN MEMORY .....	18
2.10. SCREEN PARAMETERIZATION.....	18
2.11. SCREEN LEVELS .....	21
<b>3. GRAPHIC .....</b>	<b>23</b>
3.1. GRAPHICS EDITOR .....	23
3.2. EDITING OPERATIONS .....	25
3.3. COLOR SELECTION.....	27
3.4. STATIC GRAPHICS EFFECTS .....	31
3.5. OBJECT IMAGES.....	32
3.6. COMMON PROPERTY EDITOR.....	33
3.7. OBJECT CONTEXTUAL MENU.....	34
3.8. OBJECT COMMAND MENU.....	35
3.9. COMMAND AND ANIMATION EXPLORER WINDOW.....	37
3.10. USING XAML IN MOVICON.NEXT .....	38
3.11. ALIASES IN OBJECTS .....	41
<b>4. SYMBOLS .....</b>	<b>47</b>
4.1. CREATING COMPOSED SYMBOLS .....	47
4.2. SYMBOL GALLERY .....	47
4.2.1. Inserting symbols from the Symbol Gallery .....	49
4.3. SYMBOL LIBRARY CONCEPTS .....	51
4.4. SELECTING COMPOSED SYMBOL COMPONENTS .....	52
4.5. POWER TEMPLATES.....	53
<b>5. TOOLBOX OBJECT AND PROPERTIES.....</b>	<b>57</b>
5.1. TOOLBOX.....	57
5.1.1. Inserting objects from the Toolbox.....	57
5.2. GENERAL DRAWS AND OBJECT PROPERTIES .....	58
5.3. DRAWS AND OBJECT BEHAVIOUR PROPERTIES.....	59
5.4. PRESERVE PROPERTIES OF LOCAL DRAWS AND OBJECTS .....	59
5.5. DRAWS AND OBJECTS USER ACCESS PROPERTIES .....	60
5.6. ANIMATION AND COMMAND PROPERTIES .....	61
5.7. MANAGING EXPRESSIONS IN DRAWS AND OBJECTS .....	61
5.8. USING ENGINEERING UNITS .....	63
5.9. ALARMS VIEWERS .....	64
5.10. ALARM WINDOWS .....	64
5.11. ALARM BANNERS .....	69
5.12. HISTORICAL LOG VIEWER .....	71
5.13. PROGRESS BARS .....	74
5.14. BUTTONS.....	79

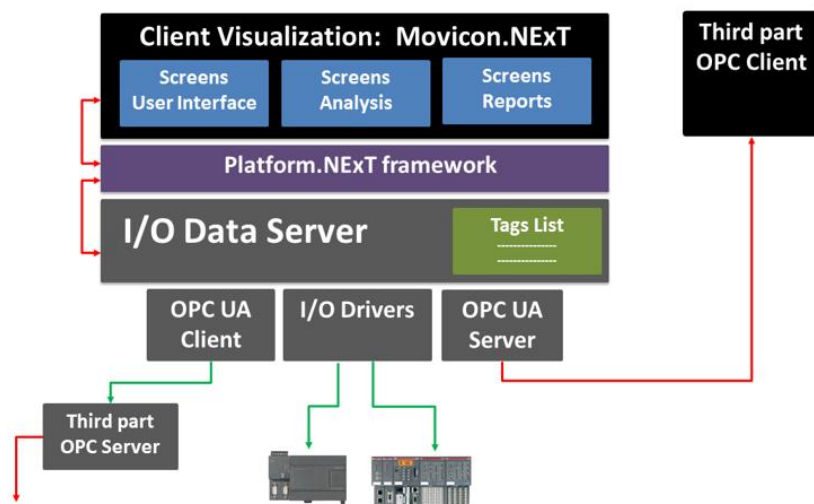
5.15. CLOCKS .....	81
5.16. DB CONNECTORS .....	84
5.17. COMBO BOX .....	84
5.18. DB LIST BOX .....	86
5.19. DB PIVOT GRID .....	88
5.20. DIALOG BOX.....	90
5.21. SWITCH BUTTONS, LIGHTS AND CHECKBOX .....	90
5.22. COMBO BOX .....	94
5.23. GROUP BOX .....	95
5.24. DB LIST BOX .....	97
5.25. OPTION BUTTON .....	98
5.26. SPIN BUTTON .....	100
5.27. DISPLAYS .....	102
5.28. GAUGES AND ARC BARGRAPHS .....	107
5.29. HOT REGION .....	112
5.30. DYNAMIC IMAGES.....	113
5.31. ANIMATED IMAGES.....	113
5.32. MEDIA ELEMENT CONTROL .....	115
5.33. IP CAMERA VIEWER.....	117
5.34. DRAWS.....	118
5.35. POTENTIOMETERS .....	122
5.35.1.....	122
5.36. RECIPE VIEWER.....	124
5.37. SCHEDULER VIEWER.....	127
5.38. SELECTORS.....	128
5.39. VERTICAL AND HORIZONTAL SLIDERS.....	131
5.40. SWITCHES .....	133
5.41. SWITCH BUTTONS, LIGHTS AND CHECKBOX .....	133
5.42. TOGGLE SWITCHES.....	137
5.43. TAB SCREENS .....	140
5.44. TEXT .....	142
5.45. TEXTS .....	142
5.46. ANIMATED TEXT.....	144
5.47. TRENDS .....	146
5.48. TREND AND DATA ANALYSIS VIEWERS.....	146
5.49. CHART.....	147
5.50. CHART XY .....	151
5.51. DATA ANALYSIS RT .....	156
5.52. HISTORICAL TREND.....	163
5.53. SPARKLINE CHART .....	167
5.54. STATES CHART .....	169
5.55. REALTIME TREND.....	172
5.56. VIEWERS.....	176
5.57. DB GRID CONTROL .....	176
5.58. AUDITING TRAIL VALIDATOR .....	179
5.59. CROSS REFERENCE VIEWER.....	181
5.60. PDF VIEWER .....	182
5.61. REPORT VIEWER .....	184
5.62. WEB BROWSER VIEWER .....	186
5.63. HISTORICAL DATA VIEWERS .....	188
5.64. AUDITING TRAIL VALIDATOR .....	188
5.65. DATA LOGGER VIEWER.....	190
5.66. HISTORICAL VIEWER .....	192
5.67. SYSTEM LOG WINDOW .....	193
5.68. HISTORICAL GRID .....	195
5.69. MISCELLANEOUS VIEWERS .....	195
5.70. CALCULATOR .....	195
5.71. MEDIA ELEMENT CONTROL .....	197
5.72. MEDIA ELEMENT CONTROL .....	199
5.73. OPC CLIENT STATUS VIEWER .....	201

5.74. SPREADSHEET VIEWER .....	203
5.75. STREAM MEDIA ELEMENT .....	205
<b>6. TAG ASSIGNATION .....</b>	<b>207</b>
6.1. ASSIGNING TAGS TO OBJECTS.....	207
6.1.1. <i>The Reference Tag</i> .....	207
6.2. POINTING TO VARIABLE BITS AND ARRAY VARIABLE ELEMENTS .....	209
6.3. EXPRESSIONS IN OBJECTS .....	210
6.4. FUNCTION_REFERENCE_SECTION.....	213
6.5. LOCAL CLIENT TAGS .....	233
6.6. CLIENT SYSTEM TAGS .....	233
<b>7. STYLES .....</b>	<b>237</b>
7.1. USING STYLES .....	237
7.2. MANAGING STYLES IN MOVICON NEXT .....	238
7.2. MANAGING STYLES IN MOVICON.NEXT FROM EXTERNAL FILES .....	244
<b>8. DYNAMICS ANIMATION .....</b>	<b>249</b>
8.1. DYNAMIC ANIMATION SETTINGS .....	249
8.2. DYNAMIC ANIMATION PROPERTIES.....	251
8.3. DYNAMIC ANIMATION FUCTIONS LIST .....	253
8.4. BACK COLOR ANIMATION .....	253
8.5. BLINK ANIMATION.....	254
8.6. BORDER COLOR ANIMATION .....	255
8.7. CLIPPING ANIMATION .....	256
8.8. COMPOSED MOVEMENT ANIMATION .....	257
8.9. DASHED LINE .....	259
8.10. DYNAMIC TEXT ANIMATION .....	260
8.11. ENABLE ANIMATION .....	262
8.12. FILLING ANIMATION.....	263
8.13. HEIGHT ANIMATION .....	264
8.14. HORIZONTAL X MOVEMENT ANIMATION .....	265
8.15. ROTATION ANIMATION .....	266
8.16. SCALE ANIMATION.....	267
8.17. TRANSPARENCY ANIMATION .....	269
8.18. STORYBOARD ANIMATION .....	270
8.19. VERTICLE Y MOVEMENT ANIMATION .....	271
8.20. VISIBILITY ANIMATION .....	272
8.21. WIDTH ANIMATION .....	274
8.22. VISUAL STATE ANIMATION.....	275
<b>9. COMMANDS FOR OBJECTS .....</b>	<b>277</b>
9.1. COMMAND SETTINGS.....	277
9.2. SCRIPT EXECUTION COMMANDS.....	280
9.3. SYSTEM COMMANDS .....	281
9.4. ALARM COMMANDS .....	282
9.5. CHANGE LANGUAGE COMMAND .....	284
9.6. RECIPE COMMANDS .....	284
9.7. COMMANDS IN METHODS .....	286
9.8. COMMANDS FOR REPORTS.....	286
9.9. USER COMMANDS.....	287
9.10. COMMANDS FOR SCREENS .....	288
9.11. COMMANDS FOR TAG VARIABLES .....	290
9.12. 3D CAMERA VIEWER COMMANDS .....	293
9.13. CHANGE CONVERTER COMMANDS .....	294
9.14. LOGIC COMMANDS.....	294
9.15. OPEN MAP COMMANDS.....	295
9.16. TEXT TO SPEECH COMMANDS .....	296

<b>10. EVENTS MANAGER.....</b>	<b>297</b>
10.1. COMMANDS ON EVENT .....	297
<b>11. 3D GRAPHICS .....</b>	<b>301</b>
11.1. 3D ANIMATION GRAPHICS.....	303
11.2. 3D SYMBOL PROPERTIES .....	308
11.3. INNER SCREENS FOR 3D ANIMATION GRAPHICS .....	310
11.4. USING THE CAMERA IN 3D GRAPHICS .....	312
<b>12. GEOLOCATION .....</b>	<b>315</b>
12.1. GEO PAGE.....	315
12.2. DYNAMIC DATA IN GEOGRAPHIC MAPS .....	317
<b>13. MULTITOUCH .....</b>	<b>319</b>
13.1. USING MULTITOUCH .....	319
13.1.1. <i>Manipulating objects</i> .....	319
13.2. MULTI TOUCH GESTUALITY .....	321
<b>14. STRING AND MULTILANGUAGES .....</b>	<b>323</b>
14.1. TEXT STRINGS .....	323
14.2. TEXT MANAGEMENT.....	325
14.3. AUTOMATIC TEXT TRANSLATION .....	328
14.4. STRING EDITOR COMMANDS .....	329
14.5. SPECIAL STRING ID .....	330
<b>15. COMMAND MENU.....</b>	<b>339</b>
15.1. PROJECT MENUS.....	339
15.2. MENU PROPERTIES .....	341
15.3. MENU ACTIVATION IN RUNTIME.....	342
<b>16. SHORTCUT .....</b>	<b>343</b>
16.1. PROJECT SHORTCUTS .....	343
16.2. SHORTCUT PROPERTIES .....	344
16.3. SHORTCUT ACTIVATION IN RUNTIME .....	345
<b>17. DIAGNOSTIC AND LOG.....</b>	<b>347</b>
17.1. SYSTEM LOG WINDOW .....	347
17.2. WATCH WINDOW(TAGS) .....	348
17.3. SERVER (OPC UA BROWSER) DIAGNOSTICS WINDOW .....	349
17.4. CLIENT STATUS (OPC UA CLIENT STATUS) WINDOW .....	349
17.5. GRAPHICAL OBJECT DIAGNOSTICS .....	349
<b>18. GENERIC INFORMATION .....</b>	<b>351</b>
18.1. TIPS AND TRICKS.....	351
18.2. HOW TO CREATE WPF OBJECTS IN PLATFORM.NEXT .....	353

## 1.1. Movicon.NExT

Movicon.NExT is the HMI Client visualization module designed to manage and represent data graphically using the Automation Platform.NExT graphical screen pages. The Movicon.NExT Client module is the most important functional module along with the I/O Data Server module in the platform. This is because it allows the user to create graphical data representations (HMI, Human Machine Interface) of the entire system. The Client visualization model operates independently from the server. It is an application that is integrated in the platform and used for accessing the framework's Address Space to expose data and tags by representing them graphically in Screen containers that are project resources that manage the graphics. The graphics use drawings and symbols, symbol libraries and objects from the toolbox that, in association with reference Tags, provide user interface management (HMI) by means of displaying and setting values.



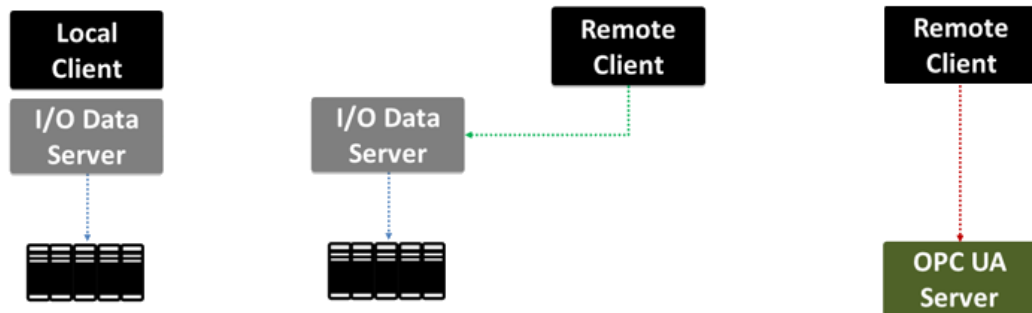
The graphical Movicon.NExT interface, as with the whole platform, is based on the WPF technology (Windows Presentation Foundation), the vectorial rendering engine of screen resolutions compiled for taking advantage of modern hardware component graphics. In addition to this main element, WPF offers a complete set of application development functions that include Extensible Application Markup Language (XAML), controls, data associations, layouts. 2D and 3D graphics, animations, styles, models, documents, multimedia elements, texts and Images. WPF is included in the Microsoft .NET Framework to compile applications that incorporate other elements from the .NET Framework class library.

### "Client Only" architecture

The Movicon.NExT Client module can be managed as:

- **Local Client**, meaning in the same machine where the I/O Data Server is running (i.e. Server-Client workstation)

- **Remote Client**, meaning in a different machine in respect to the one running the 'I/O Data Server'. In this case, only the client part can be installed using the ClickOnce setup as described in the topic on "ClickOnce Client Startup"
- **OPC UA Client** of a third party OPC UA server. The system's architecture, according to the client and server distribution, consents to maximum flexibility in managing and collocating data.



*This diagram shows architecture possibilities: distributing Client-Server, Client only or Server only applications.*

## ClickOnce Movicon.NExT Client

ClickOnce is a technology that allows the user to install and run a Windows application simply by clicking on a web page link.

The advantages in using this technology are:

- facilitated installation that is typical of web applications.
- simple to keep installed application updated.
- reduction in the impact that the installation may have on the system being used.
- no need to have administrator privileges

Applications distributed by means of using ClickOnce have a low impact on the entire system as they are only installed for each individual user.

The ClickOnce Movicon.NExT Client make remote Client architectures easier to handle especially when they include several Clients.

Further information on using Movicon.NExT ClickOnce can be found in topic: "ClickOnce Client Startup"

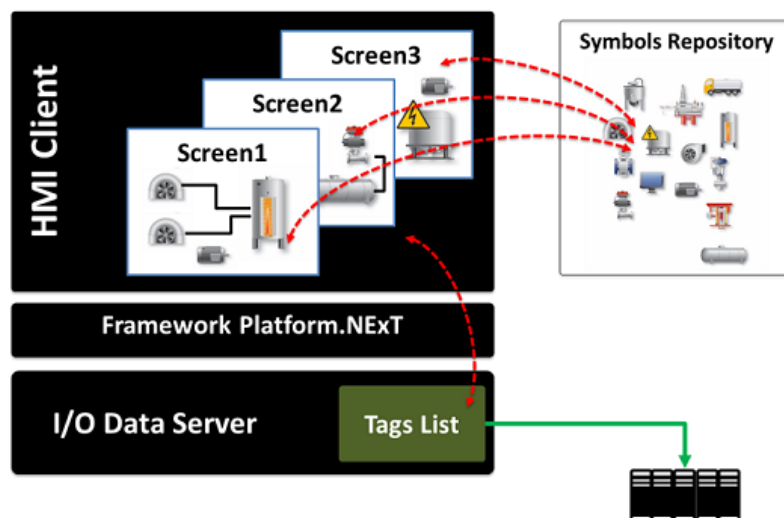


## 2. Screens

### 2.1. Project Screens

The screens are the fundamental resources for creating the project's graphical interface. In addition to graphical designs the screen windows are designed to contain controls being all the ready to use command and display graphics functions provided by Movicon. Screens are used to graphically represent all or part of the production process by using the commands and graphical animations associated with Tag variables connected to the field.

The screens are inserted in the projects using the "Screen" resource from the Project Explorer Window.



*A scheme showing the platform's Client graphical interface screen concept*

The Screens are based on the user interface created in the project by the design engineer. They are displayed in runtime to provide display functions and dynamic commands that are managed according to the Windows standards by means of using screen windows. These windows can be managed as whole screen pages with or without a frame (title bar and system menus) or as "pop-up" windows overlapping other windows in modal mode.

The modality of using these windows and changing pages can be defined by the user as pleased and as described later on in this guide.



The Screens are run as "Clients" of a project. Their graphical content is connected to Server variables, being both the platform's data Server and third party OPC UA data server.

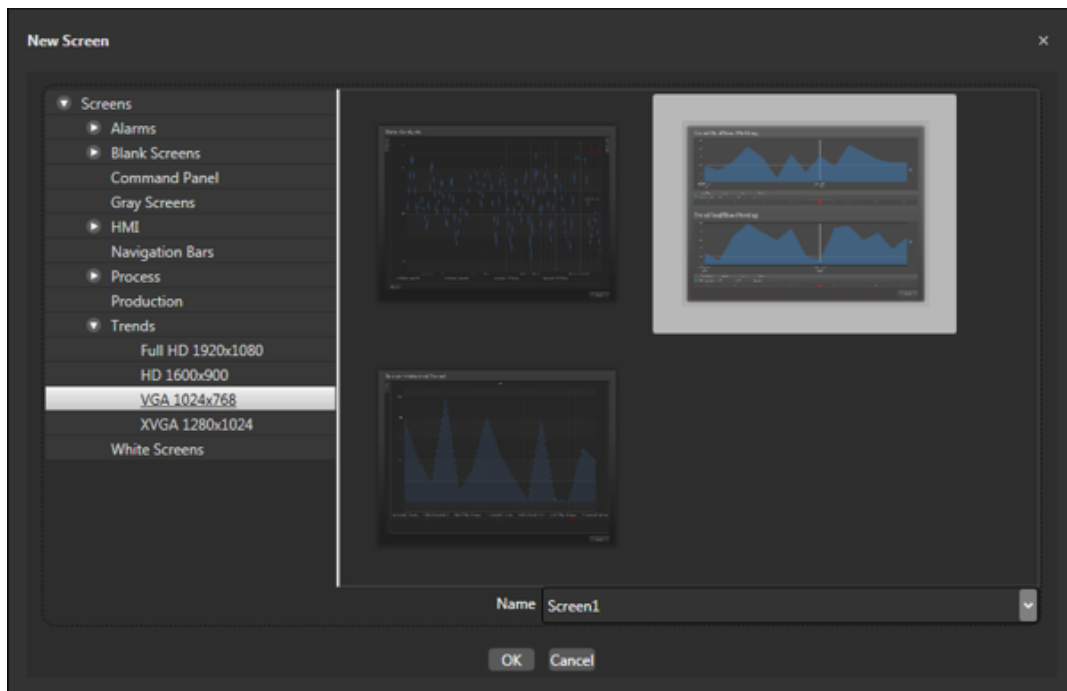
## 2.2. Screen Templates

Templates can be used for creating new screens.

### Screen Templates

when using the "New" command a window will appear through which a Screen Template can be selected and used as a "**new screen**". Movicon offers a selection of ready-to-use screen models in different formats for different types of use. These model types vary from an 'empty' screen to those set with predisposed graphical elements according to the most common usages.

Using "templates" for creating new screens will save the design engineer a great deal of time in the development stage considering that the "template" models can be customized as pleased.



To confirm the command of a new screen during the creation phrase you must first choose one of the Screen Templates from those available and then confirm with "OK". The new screen will then be created in the project based on the chosen model. The screen properties can then be modified using its Properties Window.

### Creating new customized Screen Templates

Screen Templates are screen models that are used for creating new project screens. Each time a new screen is inserted a Screen Template can be selected to use its characteristics for creating the new screen. The Screen Templates can also be created by the programmer to realize their own library of screen models to use as required. To create a Screen Template simply select one that has already been configured in the workspace or configure it as pleased by inserting the desired symbols and objects. Once this has been completed "save" this Screen as "Template" using the "**Create a Screen Template**" command from the "Screen Manager - Grouping - Symbol" Ribbon or use the equivalent command obtained by right clicking on the screen and assigning screen name and desired folder. Once this has been completed the screen should appear among those Templates available when creating the next new screen.

The Template Screens are saved in the following folder:

C:\ProgramData\Progea\Movicon.NExT\NewScreenTypes\



Attention: Screen Templates are deleted or removed by using the files in this folder.



When creating a new Screen Template it would be best to consider whether those variables, which were previously associated to symbols in such screens, can be used in different projects other than the one they were originally defined in.



The screen template will therefore be available as an object of the design environment that can be used within our or other projects.



*An example of a Movicon.NExT screen. This screen is one of those available from the system Screen Templates*

## 2.3. Screen Properties

The Screens are project graphical pages in which various objects, symbols and controls can be inserted for managing the HMI graphical interface. The Screens are managed by the project Client side.

Each screen has two property group settings. The first group is used for defining the screens aspect within the 'Tile Page' and its geo-localization co-ordinates. The second property group is used for customizing the screen's functional role when opened in runtime.



**Warning!** these two property groups can be accessed from two different points in the editor. The first property group is displayed in the Property Window by selecting the screen name in the project's Explorer Window. The second group is displayed after having opened the screen and focused on.

## General Screen Properties

The general screen properties are displayed in the Properties Window by selecting the name of the screen in the Project Explorer window. These properties are:

### Visible

When setting this property to True, the screen will be made visible in the Tile Page when the project is started up in Runtime. The screen will be represented with a square containing the name of the screen and a preview of its image. The screen will open by simply clicking this square. When this property is set to False, the screen will not be available in the Tile Page.

### Window Transparency

A screen's The WindowTransparency parameter is used to define the screen's transparency level. This parameter is taken into consideration by the screen only when opened in Synchro mode (Modal or Frame, the "Synchro Popup" option must be disabled), or when the screen is opened outside the runtime shell. This means when the screen is opened in a different monitor to the one set for default. Otherwise this parameter will have not effect.

The value of this parameter goes from 0.1 to 1, where 0.1 is the maximum transparency value and 1 is the maximum transparency value.



**Caution!** When an transparency value that is inferior to 1 is applied, the screen will open in the 'none' style ('Window Style' property). This means that the window will be without a title bar and with minimum sized borders.

### Window Transparency Only Inactive

When this property is set to True, the screen will become transparent only when without focus. When focused, the screen will return to its maximum Transparency level. This property can only be used when the screen has been opened in Synchro Frame mode as this the only open screen mode that permits the screen to be without focus in order keep it at the forefront.



Enabling the Window Transparency Only Inactive parameter for a screen that has been opened in "Modal" mode will display the screen as if set with the Transparency 1 value due to the fact that it will always have focus.

### Disable Zoom

When Enabled, the zoom function will be disabled for the screen.

### Scan Children Elements

This is used to disable the scanning of elements to determine whether there are any 3D elements to be associated for manipulation, ScrollViewer (again for touch screen manipulation), or images to be qualified and/or started up as media elements.

### Use Intellisense

Check or uncheck this option box to enable or disable the use of the intellisense in the object's script code.

### Force Writing On Server

This property is set to false for default. When set to true, forces the value of a variable to be written using script unconditionally from the previous one written and therefore the same value can be rewritten within a tag.

### Map Zoom level Visibility

This property is used to set the Zoom level for map visibility.

## Screen Execution Properties

The screen's execution properties are grouped into the following settings as follows:

### Close Screen Delay

This value represents the delay time in milliseconds that once elapsed will unload screen from memory once closed.

### Keep Always in Memory

This property is used to specify whether the screen should be kept in memory or not after being closed. Setting this property to 'True' will keep the screen loaded in memory after being closed.



*Attention! This property has no effect on pop-up windows.*

### Load Symbols Delay

This value represents the time in milliseconds with which the screen's symbols will be loaded when the screen is opened.

### Load Objects Synchronized UI

When this property is enabled, the screen will open in synchronization with the User Interface. This means it will open only when all of its object contents have been loaded.

### Monitor Number

This parameter is used for setting the number of monitors in which screen it to be displayed for PC systems using two or more monitors. In this case the Windows Operating System identifies each monitor with a number.

In most cases this simple setting is enough to ensure the destination monitor. However, in cases in which differences were found, for example, indicating monitor 2 in this property when in reality the destination monitor happened to be monitor 1 at runtime, you will need to configure a map of correspondence between the monitor indicated in the Movicon NExT editor and the actual destination monitor.

To define this configuration you will have to manually edit the ScreenManager.dll.config file contained in the DocumentManagers folder in the Movicon NExT installation path by modifying the **MultiMonitorMap** item.

Default value:

```
<setting name="MultiMonitorMap" serializeAs="String">
  <value />
</setting>
```

Value changed for exchanging monitor 1 with monitor 2:

```
<setting name="MultiMonitorMap" serializeAs="String">
  <value>1,2|2,1</value>
</setting>
```

Practically, the map defined a series of pairs separated by '|' in which the first element represents the monitor configured as the property in the Movicon.NExT project and the

second element represents the actual destination monitor in the system. Great care must be taken when modifying the file of interest and to avoid any mishaps from happening we strongly advise you do a backup copy of the ScreenManager.dll.config file before changing it.

#### **Write Timeout**

This is used to set a timeout which displays when writing a tag through using script code.

#### **Apply Window Settings On Load**

When this property is enabled, the screen settings will be applied when screen is being loaded.

### **Style**

The screen's graphical properties are displayed once the screen has opened and given focus within the workspace. These are some of the properties that are available:

#### **Background Color**

This is used for setting the screen's background color.

#### **Hide Scroll-bars**

Enabling this property will hide the screen's scroll bars.

#### **Show Navigation Buttons**

Shows the navigation buttons at the side of the window.

#### **Show Header**

Adds a header to the screen showing the screen's name.

#### **Show Tab Strip**

Shows a navigation bar indicating the last screens displayed.

#### **Fit In Window**

When this property is enabled, the screen will adapt to the size of the window containing it.

#### **Show Toolbar**

This property is used for showing or hiding the toolbar at the top of the screen. When this property is set to 'True', the toolbar will appear when moving the cursor at the top of the screen to allow you to perform various operations which include: return to home, user Login etc.

#### **Window Style**

This property is used for defining the screen opening mode. The options are:

- **None:** screen will open in a window without any title bar.
- **SingleBorderWindow:** screen will open in a window with a title bar and simple border.
- **ThreeDBorderWindow:** screen open in a window with the title bar and a 3D border.
- **ToolWindow:** screen will open with title bar and a simple border styled to the minimum.

#### **Window State**

This property is used for defining the screen opening mode. The options are:

- **Normal:** the screen window will open with the default sizes defined in the "Width" and "Height" properties.
- **Minimize:** the screen window will open minimized, that is reduced to an icon in the Windows bar.
- **Maximize:** the screen window will open maximized. In this case as set in "FitInWindow" property to either adapt to the window containing it or remain with its default sizes.

### Resize Mode

This property is used for defining the screens resizing mode. The options are:

- **NoResize:** screen cannot be minimized or resized
- **CanMinimize:** screen can be minimized using the button on the top right of the screen's title bar
- **CanResize:** screen can be minimized by using the appropriated button on the top right of the screen's title bar as well as resized
- **CanResizeWithGrip:** screen can be minimized by using the appropriated button on the top right of the screen's title bar as well as resized. The sizing grip will appear in the right bottom corner of the window.

### Tile Size

This property is used for setting the size of the frame with which the screen is displayed in the "Tile Page". This parameter is only considered when the "Visible" property is equal to "True".



Editable when selecting the screen from the project tree structure.

This property can be set with the following values:

- ExtraSmall
- Small
- Large
- ExtraLarge
- live

which correspond to an increase in size of the frame displayed in the "Tile Page". When passing from one value to the next t enlarge size, the frame will double in size.

### Tile BackColor

When a screen is visible in the Tile Page, the background color of the rectangle, which identifies the screen, will assume a color at random. This Color property is used to specify a background color for the square that identifies the screen in the Tile Page. When this property is left with the default value, the background color of the square will change to a different color at each startup.



Editable when selecting the screen from the project tree structure.

### Tile Description

This property is used for associating a description to the screen. This description will be displayed within the square representing the screen on the Tile Page. In this case, an

automatic scroll between the screen's name, preview image and description will be applied within the square.



Editable when selecting the screen from the project tree structure.

### Show in Taskbar

Allows the screen button to be shown on the Taskbar.

### Window Startup Location

This property is used for defining the screen's opening position when the "WindowState" property has been set to "Normal". In this case the screen will open with its default size and the window's position will be based on the following options:

- **Manual:** screen will open according to the coordinates defined in the "Top" and "Left" properties.
- **CenterScreen:** screen will open in the center of the screen window.
- **CenterOwner:** screen will open in the center of the window from which it was called. For example, this can be used for modal and Popup screens that are opened in the foreground in respect to the screen from which they were called.

## Layout Properties

These properties are used to define the layout of the screen and its object contents.

### Zoom Level Cache Mode X

This property is used to define which zoom level the screen's objects are to keep in memory (low zoom quality) when screen has been set with the 'Always keep in memory' option. If the value defined in this property is exceeded, the object will be reloaded and updated in memory with the high quality zoom.

### Zoom Level Cache Mode Y

This property is used to define which zoom level the screen's objects are to keep in memory (low zoom quality) when screen has been set with the 'Always keep in memory' option. If the value defined in this property is exceeded, the object will be reloaded and updated in memory with the high quality zoom.

### Layout View

Allows the screen's layout to be viewed on condition that one has been configured beforehand.

### Layout View Editable

Allows the screen layout to be edited in runtime.

### Top

This is used for setting the screen's Y coordinate position in pixels when opened. This setting is only considered if the "Window Startup Location" property has been set to "Manual".

### Left

This is used for setting the screen's X coordinate position in pixels when opened. This setting is only considered if the "Window Startup Location" property has been set to "Manual".



**Width**

This is used for setting the screen's width in pixels. When this value is set at zero, the screen's width will be automatically adapted according to the position of the screen's object content without leaving any unused space.

**Height**

This is used for setting the screen's height in pixels. When this value is set to zero, the screen's height will be automatically adapted according to the position of the screen's object content without leaving any unused space.

**User Access**

The User Management properties are used to protect the screen from unauthorized persons.

**User Visibility**

This is used to set a list of users who can display this screen in GeoPage.

**Roles Visibility**

This is used to set a list of roles that can visualize this screen in GeoPage.

**Login Required**

When enabled this feature will ask user to enter a password before opening the screen. If user login is denied, a warning message will show and the screen will open with nothing in it.



This feature is very handy when navigating different screens using the system's automatic functions such as the Startup "Tile Page" or "Gallery Page". In cases like this, controls are automatic and do not require user area and level definitions. Conversely user access level and areas can be indicated in cases where the screen is opened by using a command button.

**GeoScada**

This section includes various properties specific to geo-localization:

**Screen Latitude Coordinate**

This property is used to define the screen latitude coordinates so that it can be geo-localized.

**Latitude Tag**

Used to set the tag which will contain the value of the screen's latitude.

**Screen Longitude Coordinate**

This property is used to define the screen's longitude so that it can be geo-localized.

**Longitude Tag**

This property is used to set the tag which will contain the screen's longitude value.

**Connection Settings**

These connection properties determine the modes relating to the screen's data connections and are grouped in the following settings:



All the properties relating to the connection settings will only be available when the session is indicated using the 'Session Name' property.

**Session Name**

This property is used to indicate a private session within which the screen will be managed. When a session is not indicated here, the entire project session will be used for default.

**Remove Item Delay**

This is used to set the delay time in seconds to be used for removing the OPC UA subscriptions of non-active items.

**Remove Item Num**

This is used to set the number of non-active OPC items to be removed at each time interval.

**Only Secure Connections**

When enabled this property permits connection to those servers declared and certified safe.

**Fast Sampling Interval**

This parameter is only managed when the screen's "Session Name" property has been set. This property is used to define the refresh rate of existing in-use tags. This parameter is passed to the Server and any eventual Driver when the screen is loaded and the tags go into use. When the screen has not been defined with a session, the one defined for the project will be used and the general value used in the 'Fast Sampling Interval' property will be used as a consequence.

**Slow Sampling Interval**

This parameter is only managed when the screen's "Session Name" property has been set. This property is used to define the refresh rate of existing tags that are entering not in-use mode. This parameter is passed to the Server and any eventual Driver when the screen is loaded and the tags go into use. When the screen has not been defined with a session, the one defined for the project will be used and the general value used in the 'Slow Sampling Interval' property will be used as a consequence.

**Disable When Not In Use**

This parameter is only used when the screen's 'Session Name' property has been set. This property is used to indicate when to disable items that are not in use. If a 'Session Name' has not been defined, the project session and the project's 'Remove Item Delay' property value will be used instead.

**Publishing Interval**

This parameter is only managed if the screen has been set with the "Session Name" property. Based on the value set in the "Remove Item Delay" property, this value indicates after how many seconds the screen takes to unload and place tags in the not in use mode. If a 'Session Name' has not been defined, the project session and the project's 'Remove Item Delay' property value will be used instead.

## 2.4. Screen Navigation

Navigating the various screens inserted in the project can be performed by using the command objects from the toolbox. These command objects can be freely inserted on screen and assigned the "change page" command. As an alternative the system commands can also be used without needing to set any programming commands.

### Using Command Objects to open Screens

The change page commands enable you to set navigating from one screen to another in the way that suits you best. The objects which are normally used to do this are those found in the Toolbox as **Command Buttons** which are predisposed to for executing commands, in this case the "Open Screen" command. This command closes the active screen and opens the one desired. Naturally other command objects or events (menus, keyboard commands, logic, etc.) can be used in the same way.

For more information on using command objects please refer to the topic on "Using Command Objects"

### Using System Commands

Movicon projects are predisposed for executing screen page navigation without obliging the programmer to insert navigation buttons that are nevertheless always available for use if needed.

The system commands provide:

#### Project Startup in "Tile" or "Gallery Page" mode:

You can select the main page type you wish to set from the project properties. When selecting Tile Page or Gallery Page, the project will startup displaying the main page with a system page representing the screens that have been inserted in the project in tile format (Windows 8 coloured squares style) or by displaying a screen preview in scrolling gallery mode. The user can select the screen they wish to open from the main page.

#### Touch Screen Navigation

The "swipe" scroll commands can be used for touch screen systems to change pages in a sequence. Even though this type of command has been designed to change screen pages that have not been loaded in memory, they can nevertheless be loaded in memory using the "\_AutoLoad\_<name>" system name or once opened they can be kept in memory by setting their "Keep in memory" properties beforehand.

In order to swipe the screen pages, touch and scroll them with three fingers simultaneously towards the left or right.



The 'Main Page' can be defined at project startup to contain the change page commands defined by the programmer.

#### Return to Main Page

When using the system command for navigating, Movicon will enable you to set the Startup page with a Return to Main Page command which will be displayed on the system toolbar in runtime as shown below:



The system Toolbar is displayed when clicking or touching the top border area of any screen.



The change page commands can be inserted in any screen as pleased, independently from using the system navigation commands.

The system Toolbar commands in runtime are listed below starting from the left:

- Return to Startup Page
- Previous Page
- Enlarge page
- Zoom In- Zoom Out
- Reset object manipulation
- Edit parameters
- User Log In
- User Log Out
- Print

## Zooming Screens in Runtime

The zoom can be applied to the screen in runtime if the "Disable Zoom" option has not been checked.

The screen can be zoomed by using the mouse wheel while keeping the CTRL key pressed down or by using the "pitch and zoom" in Touch Screen systems. When using the zoom in Touch Screen systems, double tap to return screen to its original size.

If the "HideScrollBars" has not been ticked, the scroll bars will show when the zoom enlarges the screen beyond the size of the display screen. The Pan techniques can be used when zooming the screen beyond the size of the screen in Touch Screen systems.

### Fit in Window Option

When the 'Fit In Window' option is enabled, the screen zoom is disabled. In order to use the zoom you must restore the screen to the size it was before being adapted to the window size. When using Touch screen systems, double tap on the screen to restore it to "Non Fit" mode and then use the zoom. The screen can be returned to Fit mode by double tapping it again. In systems that do not have touchscreens, the 'Full screen' command in the screen's toolbar (the screen's "Show Toolbar" option must be enabled) is used to pass from "Fit" mode to "Non Fit" mode in order to zoom the screen.

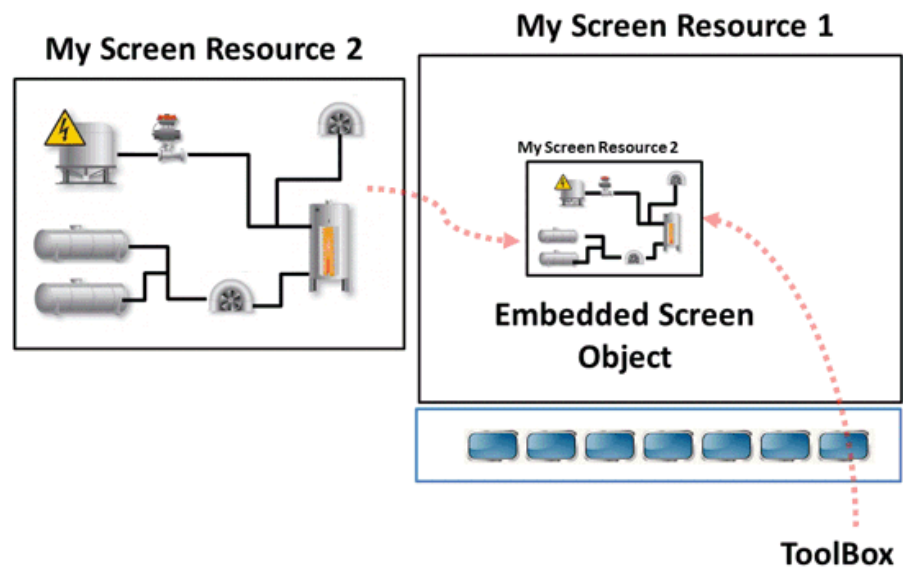
### Screen Synchro and Frame

The zoom will be applied to the entire window when opened in Synchro (modal), Frame and "Normal" mode and thus the window will be expanded within the frame.

## 2.5. Embedded Screens

The embedded screens give you more flexibility when creating user interfaces. By implementing the embedded screen concept, a screen resource created in the project can be displayed and managed as pleased not just as screen in itself but as an object inserted in another screen.

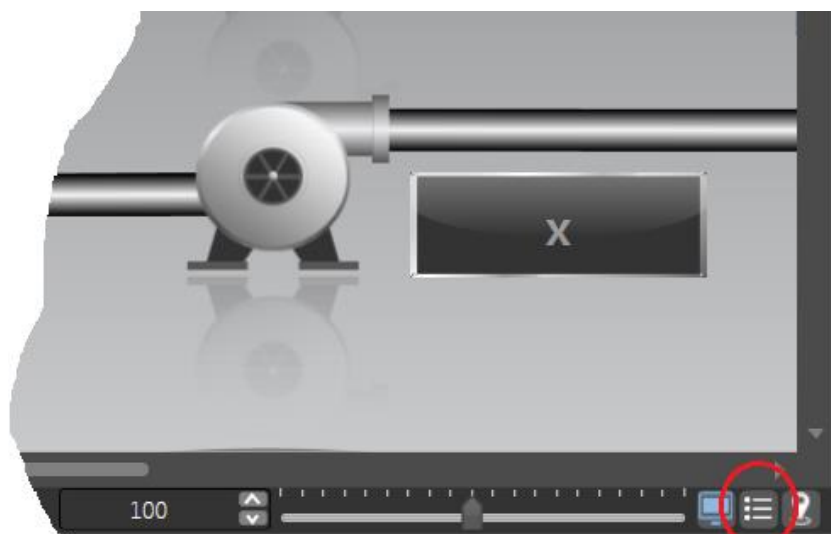
A "container" object in the toolbox has been purposely designed to allow you to insert embedded screen objects in screens by specifying which screen resource the object must be displayed in.



Using Embedded Screens as objects will enable you to have objects that point to screen resources to allow their functions to be called from various parts of the project's user interface, by centralizing the container screen as a "resource".

## 2.6. Object Browser

The Object Browser shows a list of all the screen's object contents and can be accessed using the icon found on the toolbar at the bottom of the screen.



The objects are identified by their Name and listed in a grid together with their relative details:

- Visual: number of visual objects that are used to compose the control
- Logic: number of logic objects that are used to compose the control
- Script: indicates whether script code has been inserted in the control
- Dynamics: indicates whether the control has been configured with dynamic properties such as an associated tag, command, animation, etc.

The two "Visual" and "Logic" parameters define the full WPF control index. The higher these values become the more complex and 'heavy' the control becomes to handle within the screen.

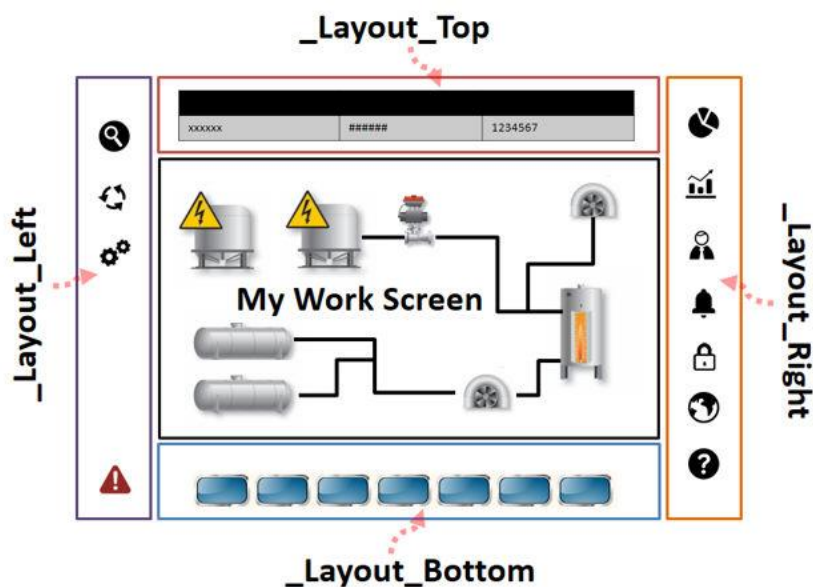
Control objects selected on screen will be highlighted on the list shown by the browser window when opened. When clicking on one of the control objects from the Object browser, it will automatically be selected on screen.

The button, in the shape of a small dot that appears on the top left corner of the popup window, will keep the window displayed at the forefront when pressed.

## 2.7. Screen Layout

A user interface can be created based on Screen Configurations containing predefined portions of the video page using screens with predefined names and positions within the layout. For example, additional screens can be used as Headings and Footers leaving space in the screen layout for the desired graphical content.

To create synoptic type Header or Footer, you must assign a predetermined name to the synoptic, as described below:



### Screen as general Header: **\_Layout\_Top**

Assigning the special name "**\_Layout\_Top**", in runtime Movicon displays the screen with this system name positioned **horizontally** starting from the video screen page's **top left**.

This screen can be sized as pleased in pixels and if necessary Movicon will scale it down to size if the width value in pixels is more than that of the video screen page's.

#### **Screen as general Footer: \_Layout\_Bottom**

Assigning the special name "\_Layout\_Bottom",In runtime Movicon displays the screen with this system name positioned **horizontally** starting from the video screen page's **bottom left**.

This screen can be sized as pleased in pixels and if necessary Movicon will scale it down to size if the width value in pixels is more than that of the video screen page's.

#### **Screen as general left border: \_Layout\_Left**

Assigning the special name "\_Layout\_Left",In runtime Movicon displays the screen with this system name positioned **vertically** starting from the video screen page's **top left**.

This screen can be sized as pleased in pixels and if necessary Movicon will scale it down to size if the height value in pixels is more than that of the video screen page's.

#### **Screen as general right border: \_Layout\_Right**

Assigning the special name "\_Layout\_Right",n runtime Movicon displays the screen with this system name positioned vertically starting from the video screen page's **top right**.

This screen can be sized as pleased in pixels and if necessary Movicon will scale it down to size if the height value in pixels is more than that of the video screen page's.

In this way the layout screens can easily be used and constantly displayed along with the workspace within the video screen page, for example the bottom navigation bar ( \_Layout\_Bottom) is always displayed for use independently of which screen is being displayed within the workspace.

## **2.8. Auto-show and swipe hidden screens**

User interfaces can be created to use predefined auto-show hidden screens that appear when the area they are hidden in is tapped and swiped or clicked on. This feature will enable you to create modern and enjoyable user interface techniques that are generally used on touchscreens to make hidden screens and their contents show by simply tapping and swiping the screen in the desired direction once it appears.

To create auto-show hidden screens, you will need to assign them with a predefined name as described below:

#### **Show Screen to swipe from top: \_AppBar\_Top**

By assigning the screen with the "\_AppBar\_Top" special name, Movicon will display the screen by making it appear **at the top** of the page in runtime allowing the use of the auto-show at top to swipe downwards and the auto-hide with a tap anywhere on screen functionalities ( "swipe" touch command). To call the hidden screen in systems without touchscreens, simply right click on the screen page.

#### **Show Screen to swipe from bottom: \_AppBar\_Bottom**

By assigning the screen with the "\_AppBar\_Bottom" special name, Movicon will display the screen at the bottom of the page in runtime so that it can be swiped upwards using the auto-show and automatically hidden by tapping on the screen ( "swipe" touch command). To call the hidden screen in systems without touchscreen use, simply right click on the screen page.

### Gadget Screens: **\_Gadget\_<name>**

Assigning the screen with a "\_Gadget\_<name>" special name, in Runtime Movicon will display it **within the workspace** in "Gadget" format in runtime to be positioned as pleased. The gadget screens also support data storage so that their position and visibility are kept persistent within the workspace. The Ctrl+Shift+F8 shortcut key can be used to reset their positions and visibilities.

## 2.9. AutoLoading Screens in memory

Normally, screens are loaded in memory the moment in they are called to display and then unloaded from memory when (after a time that can be set in the properties) once closed. However, it is also possible to set the project to automatically load and keep one or more screens in memory at startup.

This will enable quicker page changing but will commit more of the PC's memory resource. Therefore, It is completely at the discretion of the program engineer to assess and configure the project with this in mind.

### Preloaded Screen: **\_AutoLoad\_<name>**

In order to startup the project by directly loading the screen in memory simply assign it with "\_AutoLoad\_<name>". By doing this the screen will automatically load in memory at project startup despite whether being displayed or not.



This functionality can be used to obtain a sequence of pre-loaded screens starting with the project's main page. The screens are loaded in alphabetical order according to their assigned names and can be scrolled in order to pass from one screen to another.



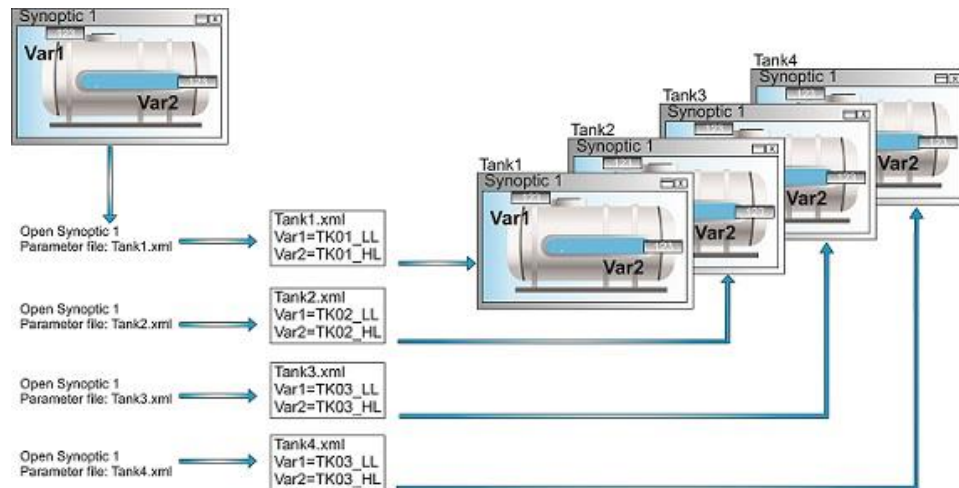
The swipe function, using the three finger gesture, has also been provided to scroll screens with. This should cause less confusion when using the other touchscreen functions (one finger gesture used for panning when scrollbars are showing, two finger gesture for pinching and zooming and the three finger gesture for swiping the pages back and forth).

## 2.10. Screen Parameterization

Modern automation often demands the possibility to allow users to parameterize resources with the aim to create just the one resource with which the desired parameters can be passed.

By using the parameterization technique, users can configure one individual screen to be called many times using different parameters. The parameters in this case are variables from the Movicon Real Time DB.





*This diagram shows an example of the openness concept where the same screen is parameterized four times for four different tanks.*

Movicon allows users to index functions that are associated to vectorial and template screen drawings. These functions can be extremely useful to programmers working on projects that contain screens that have identical graphics but different variables associated.

For example, if you have a plant containing 4 identical tanks (see example above) and you want to create just the one screen using the same tank graphics with four buttons, one for each tank, you can do this by parameterizing the screen by using dummy variables. In Runtime these dummy variables will be substituted with the effective variables associated to each individual tank.

The parameterization techniques enable the variables, which have been associated to the screen when programmed, to be substituted by the effective variables needed at Runtime in function with the parameterization file used for opening the Screen.

The parameterization file is a simple .XML file that contains the association specifications between "parameter-variable" and "effective-variable". This file is created and saved in the ProjectFolder\ProjectName\ScreenParameter folder.

For further information on screen parameterization please refer to "Comands for Screen"

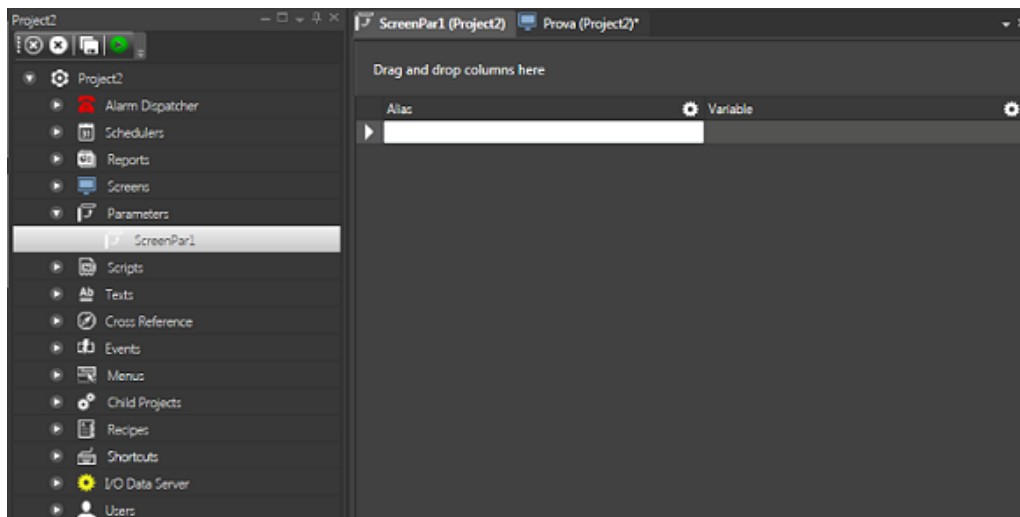


The parameterized screens load the Parameterization File while the screen is being loaded. Therefore it is essential that the:

- parameterized screen is not already loaded in RAM
- "Keep Always in memory" option is disabled
- "Close Screen Delay" property is set to "zero".

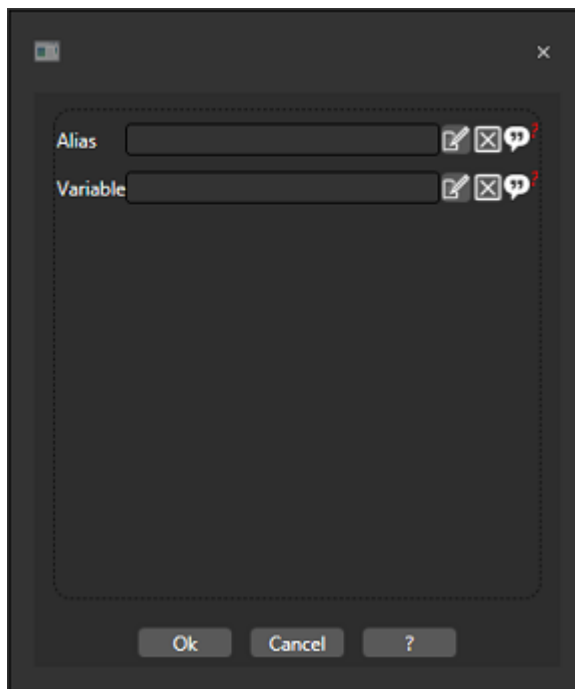
## Parameterization File

Movicon provides a resource through which it is possible to edit these parameterization files directly from the Movicon design environment. By executing the 'New Parameterization File' command from the Project Explorer's 'Parameters' group, it will be possible to directly edit a page in table format. The following window will open to do this:



The "Alias" column represents the name of the "Parameter Variable" which is the name of the variable that will be substituted each time with the name of the variable to be displayed on screen. The "Variable" column represents the effective variable to be passed to the screen instead of the Alias one. The parameterization files associated to the same screen contain the same Aliases, while the variables associated to the aliases are different.

Right mouse click and then select 'Add new parameter' to open a dialog window with the commands to insert a new parameter:



where:

**Alias:** used to select the server variable to be used as alias.

**Tag:** used to select the server variable to replace the alias when opening the parameterized screen during runtime.



The 'Alias' and 'Tag' parameters can be selected from the project's Server Tag list only. 'Local Tags', 'System Tags' or Tag from other Servers cannot be selected.

The parameterization files are always available for editing by using a normal text editor as long as the characteristics described above are respected. The end result should be that the files created with the Movicon resource and the files created manually be the same.

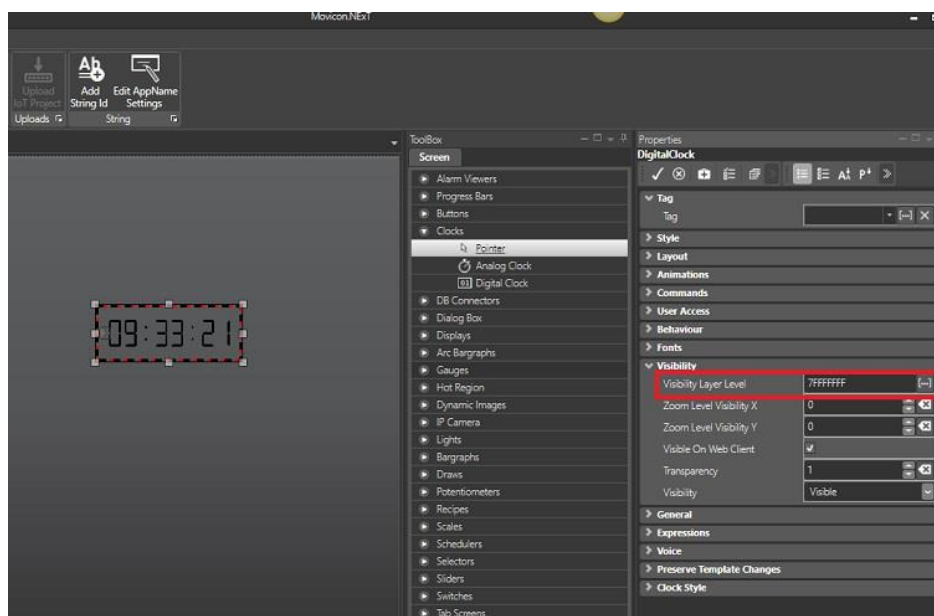


In addition to using the Movicon 'Command List' the parameterized screen can be opened by using the appropriate Basic Script functions as well.

## 2.11. Screen Levels

It is possible to associate objects with different display levels when editing a screen to enable users to then decide which levels to display or hide. In this way some objects on screen can be displayed while others are hidden in order to make work easier for the user especially when there are many objects overlapping each other in the same screen.

The display levels are managed by using a bit mask through which 32 display levels can be defined. Each mask bit represents a level. Each screen object can be set with this mask using the "Visibility Level" property. By using a bit Mask it is possible to associated each object several display levels.



Once object level mask has been defined, you can then select which level to make visible by using the "Visibility Level Editor" command from the "Screens - Layout" ribbon or by using the the "CTRL+E" shortcut. This command will open a settings window of the levels to be displayed. When confirming the settings in this window, only those objects enabled with the same bit mask level as the one defined by this command will remain visible on screen.

Each screen can be set with a custom visibility level mask so that each screen can display its own levels independently from the other screens. The visibility level mask can be reset when closing the screen so that when the screen is next opened, the mask will be set with all the enabled levels again.

The visibility level management only has effect when editing the project but not in Runtime.

## 3. Graphic

### 3.1. Graphics Editor

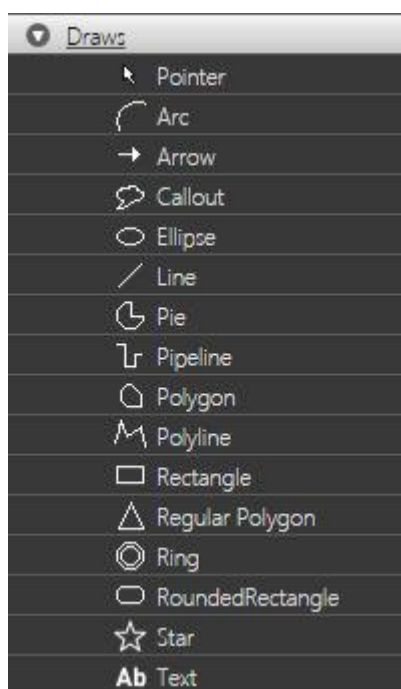
Movicon.NExT has an integrated graphics editor based on the WPF vectorial technology. This editor enables the user to configure screen HMI interfaces by using the primitive draws which are the basic shapes predisposed in the Symbol gallery or the object toolbox.

In addition users can create their own symbols with the Movicon editor, by using the Movicon.NExT drawing techniques, and customize libraries or import their own vectorial designs using a preferred graphics editor with reference to the XAML format as required by the WPF technology.

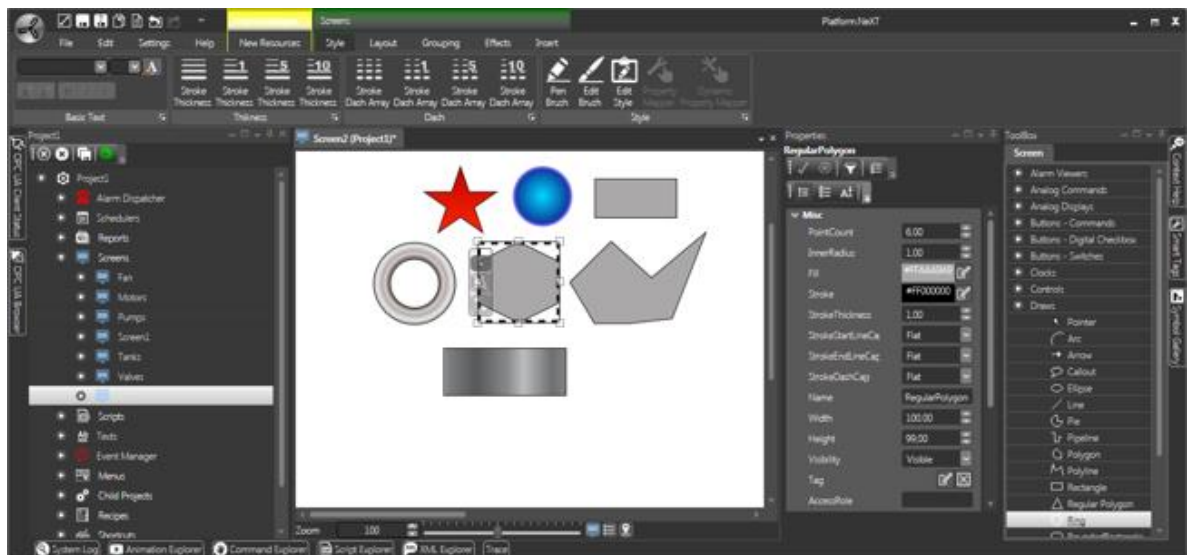
#### Graphics editor using basic shapes

Basic shapes from the ToolBox 'Draws' category can be used to create designs on screen as you please. These shapes can be inserted on screen and configured using the standard techniques common to most vectorial graphics editors.

Below are the different shapes available from the ToolBox "**Draws**" category:



Click to drag the shape desired from the Draws category to the position desired on the screen and click to release it. Movicon will insert it as a "vectorial entity" ready for sizing, coloring and configuring as pleased using its properties window.

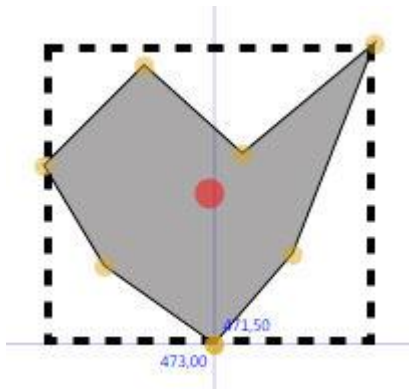


An example of the Movicon.NExT graphics editor using basic shapes from the ToolBox "Draws" category

## Changing polygon points

The polygon shape has five intersection points that can be sized and positioned as pleased by clicking on the intersection point and dragging it to the point desired. To add or remove an intersection point, simply click on the point of interest while pressing the SHIFT and CTRL keys as described:

- **SHIFT + CLICK** = Remove/Add an intersection point



The basic drawing shapes are also visible in their descriptive XML structure code in the XML Explorer Window. This can be copied and pasted for example in another XAML graphics editor.



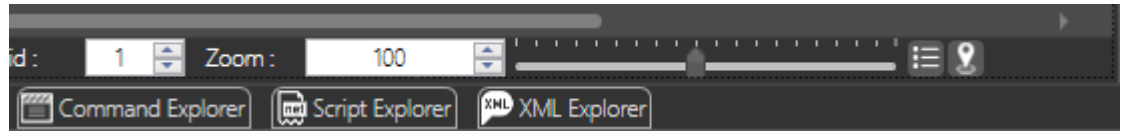
The basic drawing shapes can be grouped together and saved as Symbols in custom graphics libraries as described below.



If the last point of an polyline or pipeline object intersects another object, when you move this object by keeping the CTRL key pressed down, the polyline or pipeline will result anchored to the object and will be moved together with it.

## Zoom

All the Zoom In and Zoom Out operations during screen editing , can be set using the "Zoom Cursor" command located on the screen's bottom border or by using CTRL+Mouse wheel.



*This shows the zoom level settings located at the bottom of the screen.*

When using multitouch systems screens can be zoomed using the "pinch & zoom" gestures if enabled in runtime.

## 3.2. Editing Operations

The Movicon Graphics Editor provides users with variety of graphics editing operations for effective design engineering management and improved productivity.

### Using Grids

Graphics editing can be made easier by using grids which are accessed from the **Layout** group's "**Grid**" **Ribbons**. The commands provided by the Ribbons allow you to:

- **Toggle Grid**: Display or Hide the Grid according to the predefined resolutions (5, 10, 20, 40, 100 pixels).
- **Snap Grid** : Snaps objects to the Grid.
- **Smart Snap** : Snaps objects to align with other objects on screen.

### Stack Order

The objects are inserted on screen in overlapping TAB order which is determined by the order in which they were inserted. The order and priority with which the objects are displayed can be changed whenever needed using the **Layout** Group's **Ribbon "Order"** and "**Move**" commands, or with the right mouse key.

- **Move First**: Sets the stack order of the object selected at first place.
- **Move Last**: Sets the stack order of the object selected at last place.
- **Move Next**: Sets the stack order of the object selected at the next place.
- **Move Prev**: Sets the stack order of the object selected at the previous place.
- **Set Z Order** : Enables the stack order number of each individual object to display. The stack order can changed by clicking each object in the order desired. This command can also be activated/deactivated by using the "CTRL+D" combo keys.

### Rotate - Flip

Each object inserted on screen can be set to flip by defining its angle of rotation using **Ribbon** commands from the **Layout** group or using the right mouse key:

- **Flip Vertical** : Flips the selected object over vertically.
- **Flip Horizontal**: Flips the selected object over horizontally.
- **Rotate, Ribbon** : Sets a clockwise rotation of 90° for the selected object.
- **Rotate, Right Key**: Sets the value desired for the object's rotation angle.



Note: the object's rotation barycenter can be changed by enabling the barycenter using the "**Show-Hides Rotation Thumb**" command accessible from the object's command settings menu which opens by clicking the object's settings icon. When clicked the command will display a small red circle in the object which can then be dragged as pleases to establish a new barycenter point for rotating the object.

### Alignment

The graphical editing of groups of objects can be made easier by using the alignment commands available from the **Layout** group **Ribbon**. These commands activate after a group of objects is selected and refer to the **reference object**. The object group's reference object is identified because it is highlighted differently to the other objects belonging to the group. When multi-selecting by using the CTRL + CLICK command, the reference object will be the last one clicked on and will always be the one highlighted differently from the others in all cases.

- **Align Left**: Aligns all the selected objects to the left of the reference object.
- **Align Right**: Aligns all the selected objects to the Right of the reference object.
- **Align Top**: Aligns all the selected objects towards the top of the reference object.
- **Align Bottom**: Aligns all the selected objects towards the bottom top of the reference object.

### Size

The graphical editing of groups of objects can be made easier by using the sizing commands available from the **Layout** group **Ribbon**. These commands activate after a group of objects is selected and refer to the **reference object**. The object group's reference object is identified because it is highlighted differently to the other objects belonging to the group. When multi-selecting by using the CTRL + CLICK command, the reference object will be the last one clicked on and will always be the one highlighted differently in all cases.

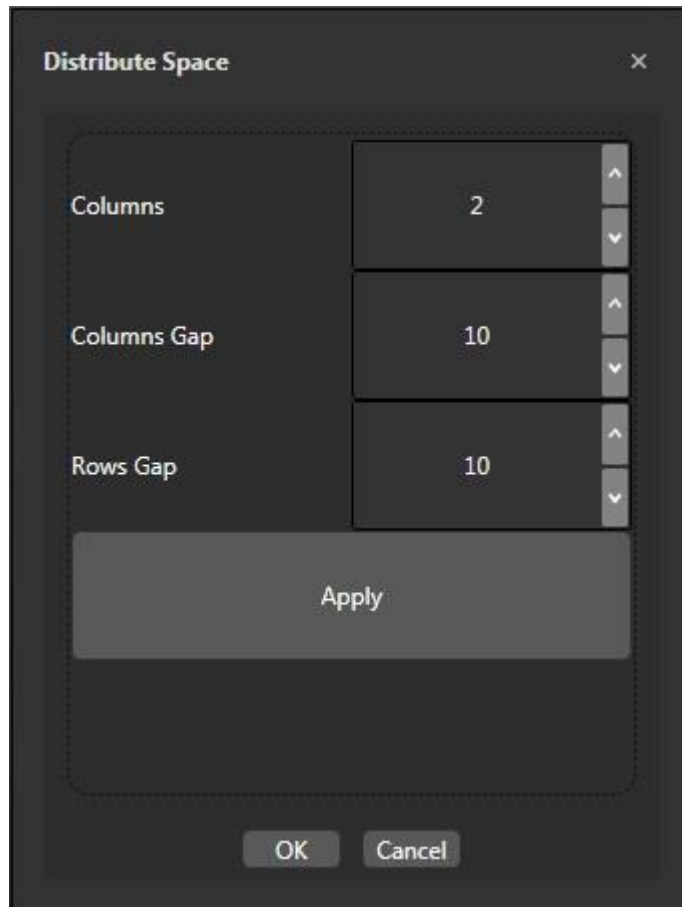
- **Same Width**: Sets the same width for all the selected objects according to the reference object.
- **Same Height**: Sets the same height for all the selected objects according to the reference object.
- **Same Width and Height**: Sets the same Size (Width and Height) for all the selected object according to the reference object.

### Spacing out objects

Spreading out and aligning groups of objects evenly can be facilitated by using the Distribute Space command which is found in the Align ribbon from the Layout group. This command activates when selecting the group of objects and distributes the selected objects by horizontally and vertically by aligning them evenly into rows and columns using the command's configuration window. Object Alignment is done



according to the tab order (overlapping order) in which the objects were selected. Therefore they will be distributed with the first object at the top left, being the object tabbed/overlapped at the bottom, with the last object at the bottom right being the one tabbed at the top.



The end alignment result can be immediately displayed using the "**Apply**" command. Use the 'OK' button to confirm operation and the Cancel button to cancel operation.

### Border Management

The borders of each object can be sized and set whenever needed using the object's properties. This can also be done faster by using the "Style" Group's Ribbon commands:

- **Stroke Thickness** : Establishes the preset thickness of the object's border line.
- **Stroke Dash Array** : Establishes the object's border's stroke dash type.
- **Pen Brush** : Establishes the borders colour selection.

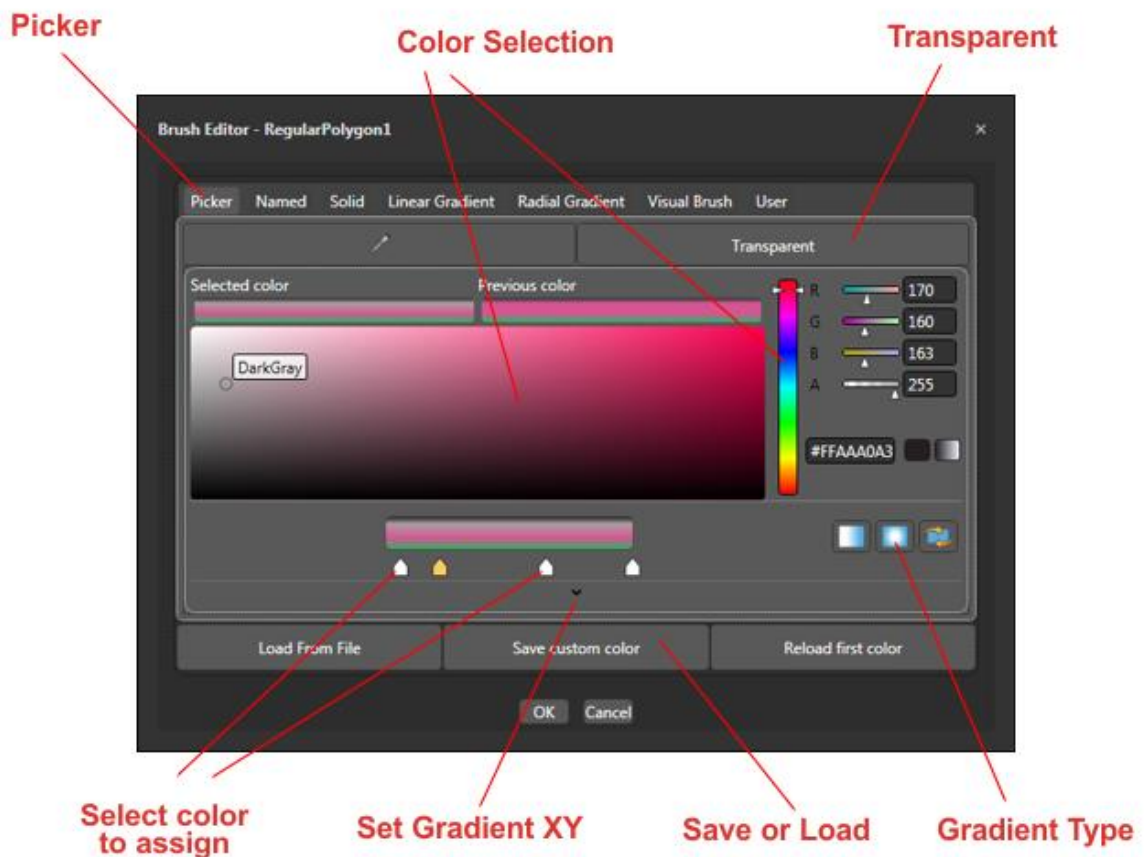
## 3.3. Color Selection

The color of various objects' borders and backgrounds is set using the Color Selection Window.

There are several options to choose as described below:

## Color Picker

By selecting the **Color Picker** Tab the user will be able to define a shading gradient between two or more colors.



By using the above shown window commands and settings, you will be able to assign the gradient colors as desired directly and save the resulting configuration on file for reloading when needed.



We recommend that you use the "Save Custom Color" command for those color gradient configurations you intend to use often with other objects in the future.

Configurations saved on file will then be available for the **"Users Brushes"** Tab.

Please note that the Movicon color Selection tool offers a vast range of preconfigured gradients to choose from.

## Named Colors and Solid Colors

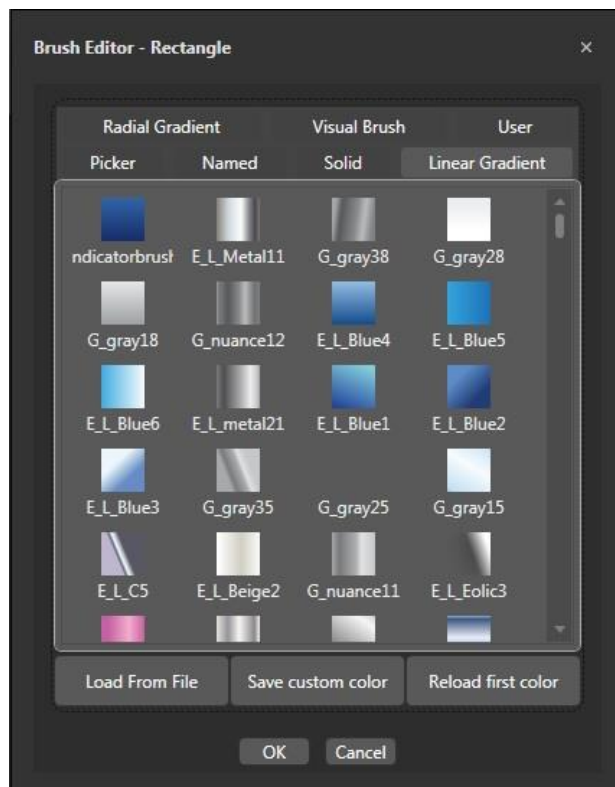
When selecting the **Named Colors** or **Solid Colors** Tab, you will be able to pick whatever color you wish to use from the color palette shown below:



### Radial Brushes and Linear Brushes

Selecting the **Radial Brushes** or **Linear Brushes** Tab, you will be able to pick any of the gradient color types you wish to use from the palette which displays.

Alternatively, the "Color Picker" tool can be used to create the custom gradients of your choice and then save them on file.



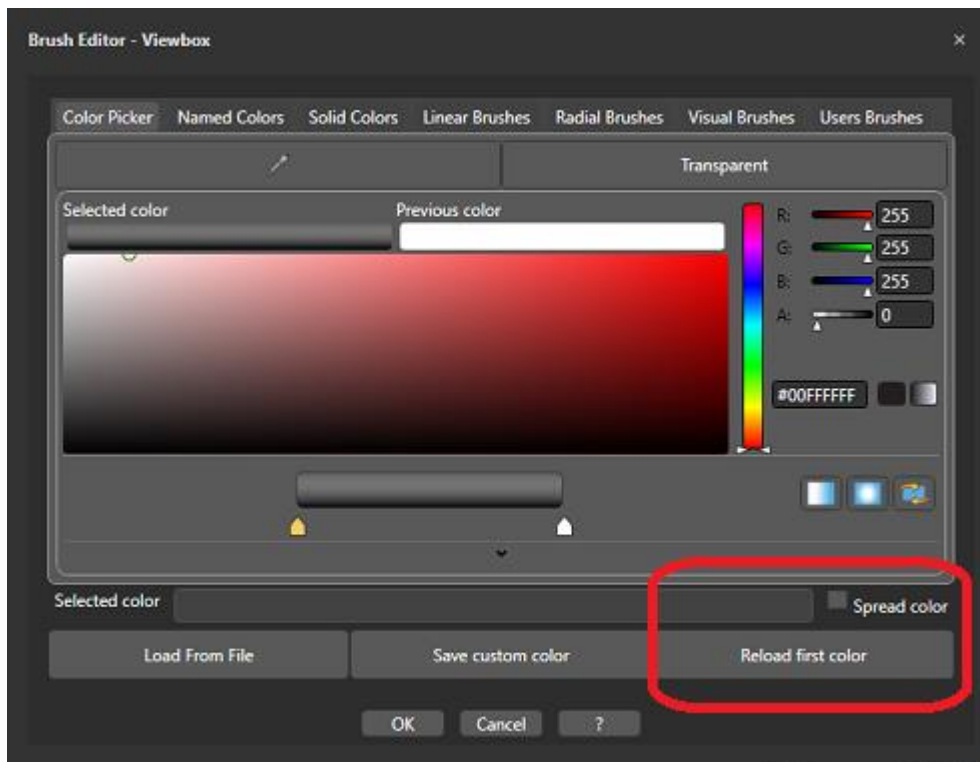
## User Brushes

By selecting the **User Brushes** Tab, you will be able to pick the color of your choice from those previously saved on file.

In order to have a selection available in this tab you will need to have used the "**Save Custom Color**" command beforehand as described above.

## Spread Color - Reload first color - Restore Color

After having selected the object (or symbol) whose color you wish to configure, a check box and two buttons will appear (according to the object type selected) named 'Spread color', 'Reload first color' and 'Restore color' respectively:



### Spread color

When selecting a composed symbol, the back or line color change takes place in symbol as a whole and not its various components. However, if you wish to spread the color to all the symbol's components, you must mark the check box before selecting the color you want to use. While selecting the color, you will be shown how it will be applied to the symbol's components.



When the back color is spread to the symbol's components, it will only be applied to those components which have been provided with the option to change back color.

This command can be applied to **composed symbols** only.

### Reload first color

After having changed a symbol's or object's back or line color, you can change it back to the PREVIOUSLY selected color by using this command before pressing OK. This command UNDOES any changes made while the BRUSH editor is open. This command can also be used for multi-selected objects as well.



This command can be applied to ALL objects.

### Restore color

This command is used to restore the back or line color of a symbol to its original color (the one set for default), with which it was initially inserted on screen with.



Caution: This command can be applied only to the following objects:

- All the objects from the SYMBOL LIBRARY
- BUTTONS WITH STYLES FROM THE TOOLBOX (the other toolbox objects do not have this function)

## 3.4. Static Graphics Effects

Each object inserted on screen can be associated with graphical effect that can be selected by using the appropriate **Ribbons** from the "**Effects**" group:



**Reflection** : Activates or Deactivates the object's reflection towards the bottom border.



**Shadow**: Activates or Deactivates the object's shadow in various positions.



**Glossy**: Activates or Deactivates the glossy effect of the object's border edge.



**Blur:** Activates or Deactivates the object's blur effect.

## 3.5. Object Images

The background of both screens and design graphics can be customized and made multimedia by associating them with external images or video clips.

To associate a video clip or an image to the background of a screen simply drag the desired file onto the screen in question. To perform the same association for object backgrounds, first select the desired object and then drag the file to it.

In this way the image will be associated to the object's background ("tag background").

The contents will always be adapted to the size of the object or screen.



Warning! The Windows application standard techniques used to drag images onto objects are not available when the application has been started up as "Run as Administrator".

To remove an image of a video clip as a screen or object "background tag", re-assign a desired color as background color (color, gradient or transparent).



**Caution! Images and video clips assigned to screens or objects will be copied to the project's "screen" folder. These copies will be deleted when removing the related associations.**

The formats supported for background images are those most commonly used:

.BMP, .GIF, .TIF, .JPG, .PNG

While those supported for movies are:

.WMV, .MPEG, .MP4

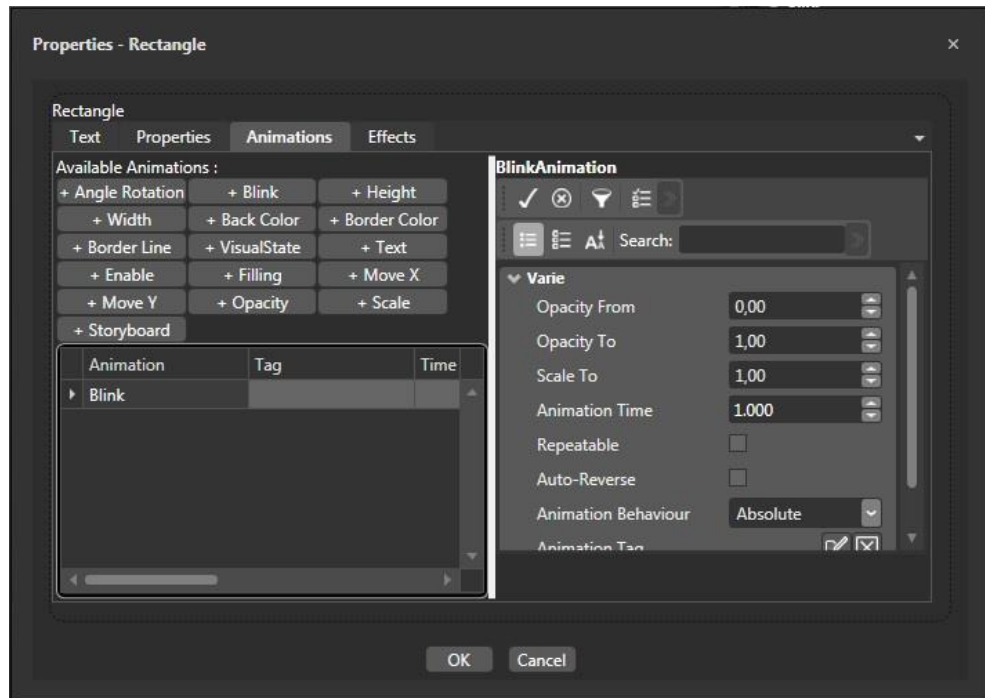


The 'Viewer' objects from the toolbox can be set with a background image or movie by using a property dedicated for this purpose. These objects can be assigned background by using the drag&drop techniques.

## 3.6. Common Property Editor

The "**Common Property Editor**" window is a tool used for quickly accessing all the selected object or symbol's main settings. These settings are divided under four Tabs which can be selected for accessing all the information and configurations that can be set.

In order to activate the "**Common Property Editor**" window simply right click on the appropriate command menu or on the appropriate menu from the object's **Command Menu**.



Window used for setting the object's animations and command list.

The Tabs located on the top window border are used for accessing the below listed settings directly:

<b>Text</b>	This window is used for setting text to associate to the object (e.g. a Title) and the consequent properties of the font to be used.
<b>Animations</b>	This window is used for inserting and configuring the object's animation, such as rotation, blink, and visibility. The animation in question will be executed based on the variable value associated to the object or, if specified based on the variable inserted in the relating animation field. The "Animation Explorer" window can also be used for editing object animations.
<b>Commands</b>	This window is used for inserting and configuring the list of Command functions to be associated to the object, such as opening a screen, setting a variable and so forth.

- This setting window is only available for those objects such as buttons for example that are predisposed for executing commands.

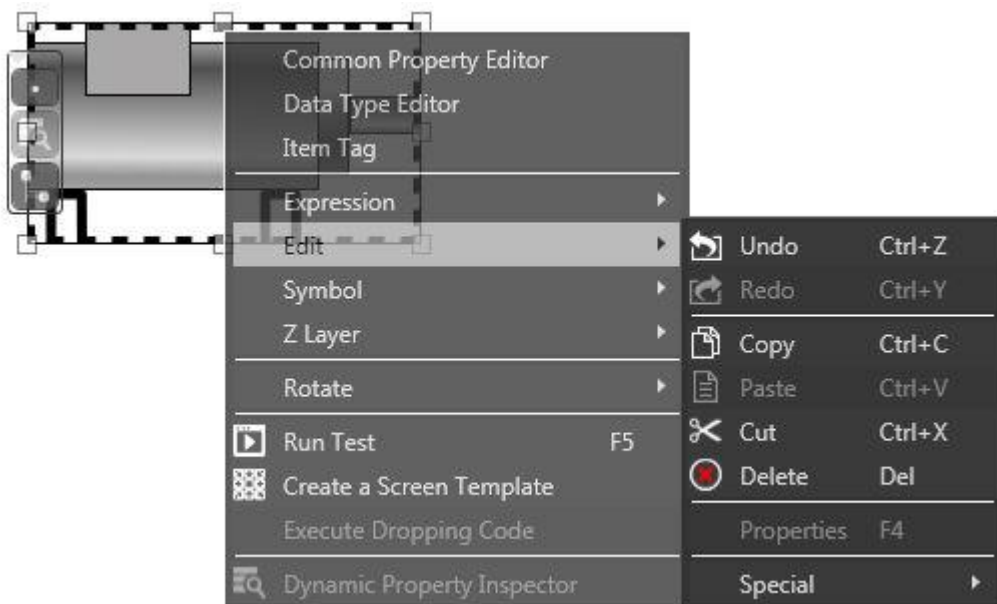
The "Command Explorer" window can also be used for assigning object commands.

## Effects

This window is used for configuring the graphical static effects of the object such as shadow, outer glow and blur.

## 3.7. Object Contextual Menu

The contextual menu is opened by right clicking on a screen object. This menu is used for executing some commands:



### Common Property Editor

Opens the Common Property Editor window through which Dynamic Animations or Command functions can be set for the selected object.

### Data Type Editor

Opens the window which shows the prototypes associated to the Power Template being examined.

### Item Tag

Opens the Tag List selection window for associating a Variable to the selected object.

### Expression

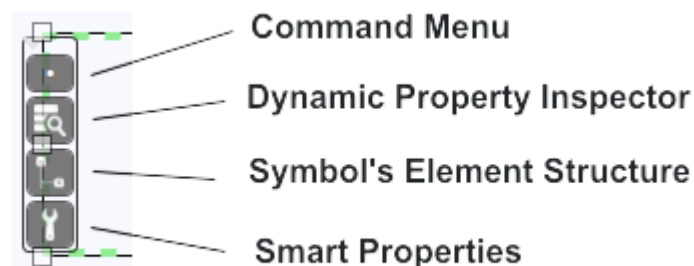
Shows the fields used for assigning Input or Out calculation Expressions for the selected object.

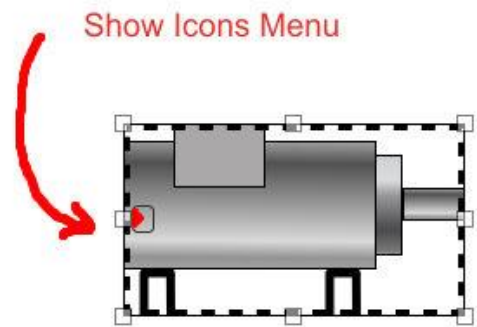
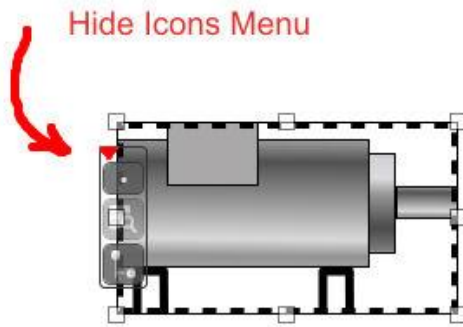


<b>Edit</b>	In addition to the usual copy, paste and other items, a " <b>Special</b> " item has been included for copying and pasting "Command List", "Animation List" and "Reference Variables" from one object to another.
<b>Symbol</b>	This is used for grouping symbols together.
<b>ZLayer</b>	Accesses the editing commands for the overlapping display order in the selected symbol (TAB order).
<b>Rotate</b>	This is used for managing the object's rotation.
<b>Run Test</b>	Starts the screen up in runtime. Only the active screen will be started up for Test running.
<b>Create a Screen Template</b>	Executes the command which inserts the active Screen to the Template list in order to add it to those already available.
<b>Execute Dropping Code</b>	Executes any eventual dropping code associated to the object.

## 3.8. Object Command Menu

When an object is selected on screen it will be highlighted with a dashed line. A command menu with icons is made available at the side of the selection border as shown below:

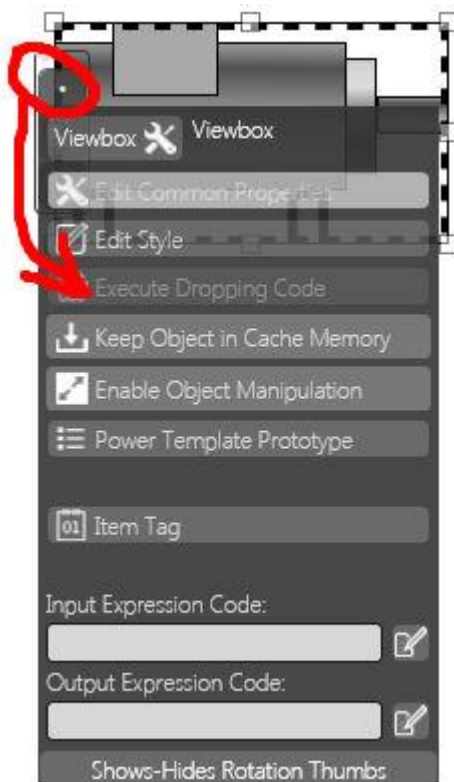




The icon menu can be displayed or hidden using the small triangle button designed for this purpose as shown above. It is very handy to hide this menu when needing to edit the object's borders.

## Command Menu

The command menu of the various objects can be activated by using the first command icon. When clicking this icon the Command Menu will appear with groups of all the important command and functions that can be associated to the object. The image below shows an example of a Command Menu:



**Edit Common Property**

Opens the Common Property Editor window through which Dynamic Animations or Command functions can be set for the selected object.

**Edit Style**

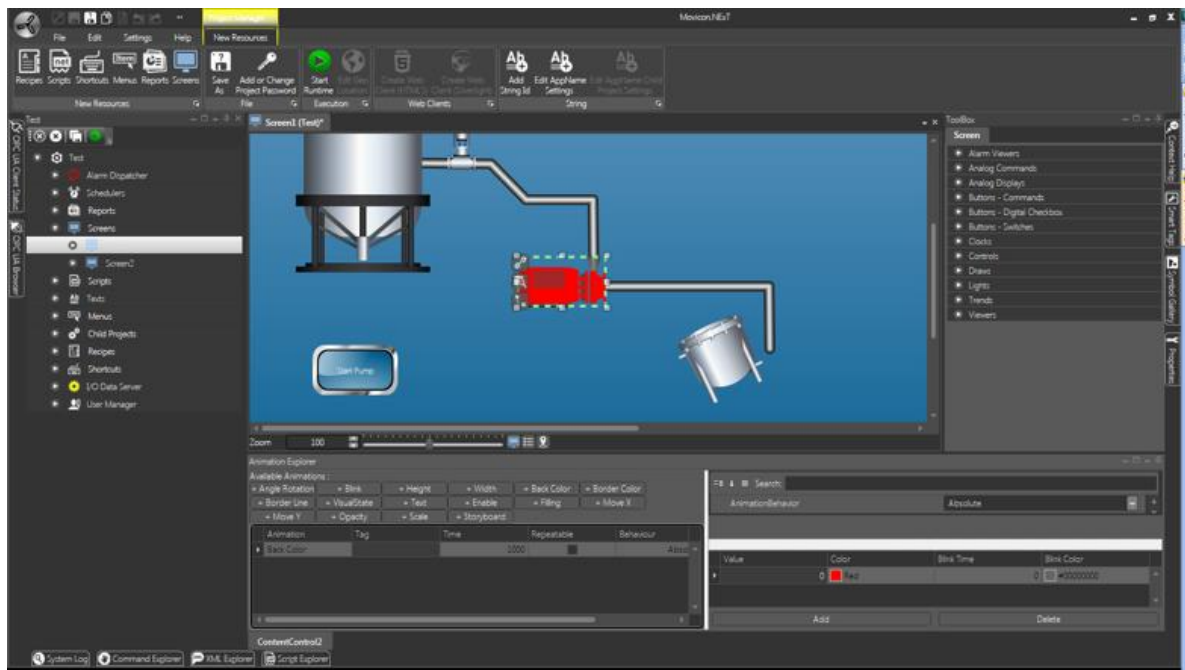
Opens the window used for selecting a new "Style" to assign to the object (only for symbols and objects from the library).

<b>Execute Dropping Code</b>	Executes any VB.NET dropping code associated to the object.
<b>Keep Object in Cache Memory</b>	Keeps the selected object in cache memory when closing screen as well.
<b>Enable Object Manipulation</b>	Enables the object's multitouch manipulation in runtime.
<b>Enable 3D Editing</b>	Enables element selecting in 3D models
<b>Enable 3D Camera</b>	Enables the positioning and recording of a view for 3D objects
<b>Item Tag</b>	Opens the selection window of a Tag list in order to associate a Variable to the selected object.
<b>Expression</b>	Shows the fields used for assigning an Input or Output Calculation Expression for the selected object.
<b>Power Template Prototype</b>	This is used for assigning "Structure Prototype" Tag models for the Power Template
<b>Shows-Hide Rotation button</b>	Enables the management that positions the barycenter point on which the object is rotated by using the "red round button".

## 3.9. Command and Animation Explorer Window

The Dynamic Animations of objects can also be edited by means of using the "Animation Explorer" window. The Animations in this window can be configured in the same way described for the "Animation Settings Window", but instead of being opened through the object's Tooltip, this window always remains open in the Workspace and behaves like the Properties Window. This means that it will automatically refresh with the Dynamic Animations list of the object selected on the screen.

This window, which shows only when a screen is opened in the Workspace, is managed in the same way as the other project windows. Therefore, it can be hidden, docked or positioned in foreground.



The "Animation Explorer" window can be accessed using the Tabs located on the top workspace border

In addition to the "Animation Explorer" window, there is also a "Command Explorer" window that enables you to explore the Command Functions assigned to the selected object as described above for animations.

## 3.10. Using XAML in Movicon.NExT

As previously mentioned, the Movicon.NExT graphics engine is based on the modern WPF technology and, therefore, the graphical elements are managed according to this technology.

### XAML definition

XAML is a declarative Extensible Application Markup Language (pronounced "zammel") applied to the .NET Framework programming model. XAML simplifies the creation of user interfaces for .NET Framework applications. User interface elements can be created and made visible in the declarative XAML markup therefore separating interface definition from the runtime logic using the code-behind file united with the markup using partial class definitions. XAML represents a direct way of creating object instances in a specific set of supported types defined in the assembly. XAML is different from the major part of other markup languages that are generally interpreted languages without direct links to supported system types. XAML enables work flow by operating in specific parts of the user interface and the application's logic being substantially different tools.

### Movicon symbol XAML code

The Movicon.NExT Symbol Library has been designed by Progea using graphics editing tools and creating XAML files optimized for representing graphics in a real-time environment. However, for product sale policy reasons, the symbols are distributed in encryption mode. This means they can only be used in projects without having access to view their XAML code. However, it is possible to obtain an unlocking code, within the

limits of the specified library's user license, to allow the user to remove the protection and get access to the symbols' XAML code.



Please contact the Progea Sales Offices or distributors in order to get more information on the symbol library's XAML code protection.



When you import 3D symbols please take into account the following:

- If you wish to manage the animation of each individual element, for example, to associate an InnerScreen in cases in where there are elements embedded at different levels in the imported XAML, the symbol to be imported will need to have the Tag="ProblematicXamlWriter".
- When the symbol is imported and contains textures, you will be asked whether the image file should be imported to the project's Images folder. If you confirm with 'NO', the image's original absolute path will be saved in the screen's xaml.settings file. In any case, an absolute path will be saved in the screen's xaml.settings file which will mean that if you rename or move the project in different path, it will be no longer possible to find the image.

## Importing XAML symbol graphics

Movicon.NExT enables users to import symbols or part of the product design to external editors. This gives users the chance to use third party editors (such as Microsoft Blend) and to enable them to use their graphical work to obtain symbols with remarkable graphical impact that can be used in Movicon screens or for creating custom symbol libraries.

By using the "**Import XAML**" command available from the "**Insert**" group ribbon, users can select the desired graphics file and import it within their project to create a new Movicon symbol.



Make sure that the file's .xaml format is specified in the browser window (selection filter).

## How to assign the Background tag (coloring) to a XAML object

Automation Platform.NExT offers you the option to vary the coloring of imported XAML objects. To create and choose the specific parts of an object that will undergo color variation, you will need to follow some rules which are fundamental to XAML programming. In order to demonstrate this we will use a simple vector button drawing as shown below:



Once the XAML code has been generated (by means of using the appropriate tools such as Microsoft Blend for VisualStudio or various automatic plugins Adobe Illustrator and such like) the code will be presented more or less like the image below.

```
<Viewbox Height="85.191" Stretch="Fill" Width="90.709" >
  <Canvas x:Name="Livello_2" Height="85.191" Width="90.709">
    <Path Data="F1M155.354,120.1895C155.354,125.9515,150.698,130.5955,144.948,130.5955L75.045,
130.5955C69.295,130.5955,64.645,125.9515,64.645,120.1895L64.645,55.8085C64.645,50.0625,
69.295,45.4045,75.045,45.4045L144.948,45.4045C150.698,45.4045,155.354,50.0625,155.354,55.8085z"
Fill="#FF76777A" Height="85.191" Canvas.Left="0" Stretch="Fill" Canvas.Top="0" Width="90.709"/>
    <Path Data="F1M75.82,46.347C70.196,46.347,65.65,50.902,65.65,56.52L65.65,56.52L65.65,
119.475C65.65,125.11,70.196,129.654,75.82,129.654L75.82,129.654L144.175,129.654C149.796,
129.654,154.351,125.11,154.351,119.475L154.351,119.475L154.351,56.52C154.351,50.902,149.796,
46.347,144.175,46.347L144.175,46.347z" Height="83.307" Canvas.Left="1.005" Stretch="Fill" Canvas.Top="0.943" Width="88.701" Fill="#FFCACA"/>
    <Path Data="F1M79.414,50.73C74.385,50.73,70.314,54.807,70.314,59.834L70.314,59.834L70.314,
116.166C70.314,121.208,74.385,125.271,79.414,125.271L79.414,125.271L140.578,125.271C145.609,125.271,
149.685,121.208,149.685,116.166L149.685,116.166L149.685,59.834C149.685,54.807,145.609,50.73,140.578,50.73L140.578,50.73z"
Height="74.541" Canvas.Left="5.669" Stretch="Fill" Canvas.Top="5.326" Width="79.371" Fill="#FF00826D"/>
  </Canvas>
</Viewbox>
```

The "path" or elements that in this case compose the image are three. However we only want to vary the green color within Automation Platform.NExT. To activate this function we will add the Tag="Background" code snippet within the line of code relating to the chosen path as shown below:

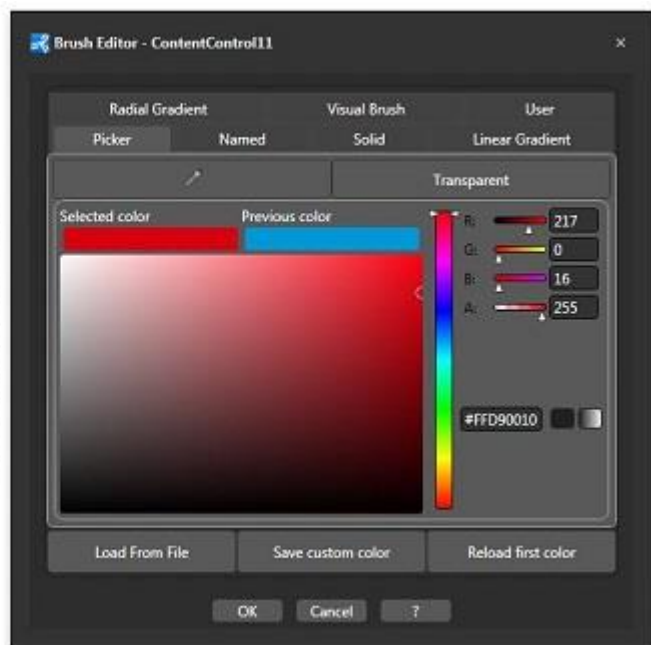
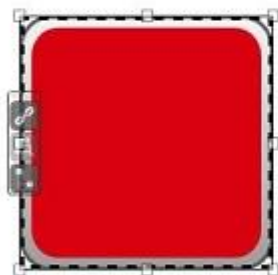
```
<Viewbox Height="85.191" Stretch="Fill" Width="90.709" >
  <Canvas x:Name="Livello_2" Height="85.191" Width="90.709">
    <Path Data="F1M155.354,120.1895C155.354,125.9515,150.698,130.5955,144.948,130.5955L75.045,
130.5955C69.295,130.5955,64.645,125.9515,64.645,120.1895L64.645,55.8085C64.645,50.0625,
69.295,45.4045,75.045,45.4045L144.948,45.4045C150.698,45.4045,155.354,50.0625,155.354,55.8085z"
Fill="#FF76777A" Height="85.191" Canvas.Left="0" Stretch="Fill" Canvas.Top="0" Width="90.709"/>
    <Path Data="F1M75.82,46.347C70.196,46.347,65.65,50.902,65.65,56.52L65.65,56.52L65.65,
119.475C65.65,125.11,70.196,129.654,75.82,129.654L75.82,129.654L144.175,129.654C149.796,
129.654,154.351,125.11,154.351,119.475L154.351,119.475L154.351,56.52C154.351,50.902,149.796,
46.347,144.175,46.347L144.175,46.347z" Height="83.307" Canvas.Left="1.005" Stretch="Fill" Canvas.Top="0.943" Width="88.701" Fill="#FFCACA"/>
    <Path Tag="Background" Data="F1M79.414,50.73C74.385,50.73,70.314,54.807,70.314,59.834L70.314,59.834L70.314,
116.166C70.314,121.208,74.385,125.271,79.414,125.271L79.414,125.271L140.578,125.271C145.609,125.271,
149.685,121.208,149.685,116.166L149.685,116.166L149.685,59.834C149.685,54.807,145.609,50.73,140.578,50.73L140.578,50.73z"
Height="74.541" Canvas.Left="5.669" Stretch="Fill" Canvas.Top="5.326" Width="79.371" Fill="#FF00826D"/>
  </Canvas>
</Viewbox>
```

Insert the Tag="Background" code snippet after <Path

Example: `<Path Tag="Background" Data="F3M20...`

It is important that you leave a space between each code snippet in order to avoid compromising the object. Therefore, be careful to space the characters appropriately next to each other. Once this has been done, the object will be active with the "Edit Back Color" function when imported to Platform.NExT.

This object can be given a background color by using the Movicon Editor or dynamically animated with the Back Color function.



### 3.11. Aliases in Objects

Handling Aliases in objects consists of the possibility to insert Aliases in the variable type fields of objects instead of using the variable's actual name. Once having done this, a table can be edited to correspond the Alias names with the Server variables to be then used in the object at runtime. When loading the symbol/ object at runtime, Movicon will substitute the Aliases with their corresponding Tag elements defined in the table. In this way it will be possible to have all the same objects, with the same Alias defined in the property but with different Table correspondence.

Using the Alias mechanism can also result quite handy when using Symbols from the library. In this case, the 'Preserve Variables' property is essential to use for inheriting the variables from the reference symbol that in this case would result as Aliases.



The Aliases can only be managed in object properties (Command or Animation). They are not supported in the object's Basic Script code. However, they can be used in the object's expressions.



Aliases can only be connected to the project's Server Tags and not to those of other Servers.

#### Alias Syntax

To be able to discriminate whether an Alias has been inserted in a field, it will need to respect a certain syntax. The Alias identification syntaxes should be characterized with the double angled brackets: <<AliasName>>.

As previously mentioned, the Aliases can be replaced by the full name of the tag (therefore include the tag path), however, it can also be replaced with just part of the tag path and used in the Expression property of an object, Command or Animation. Some possible examples of syntax uses of an Alias could be:

- Full name of a tag: <<AliasVariable>>
- Partial name of a tag: <<AliasFolderTag>>\Variable1
- Structure type tag: <<AliasStruct>>: Member1, StructVar1:<<AliasMember>>, <<AliasStruct>>:<<AliasMember>>, etc.
- Expression: <<AliasVariableBit>>, <<AliasVariableArrayElement>>, <<AliasExpression>>, ecc.
- Basic Script Code use: Document.GetVariableValue("<<AliasContextTag>>"), Document.SetVariableValue("<<AliasContextTag>>", \_value)



The "GetVariableValue()" and "SetVariableValue()" script functions only operate in the Screen's Aliases and not of those of each individual object.

The corresponding Alias Table would be:

Key	Value
AliasVariable	Folder1\ Variable1
AliasFolderTag	Folder1
AliasStruct	StructVar1
AliasMember	Member1
AliasVariableBit	.1
AliasVariableArrayElement	[0]
AliasExpression	=[x] + 10
AliasContextTag	Folder1\ Variable1

Dove "Folder1\ Variable1", "StructVar1" sono variabili del Server IO.

## Alias Table

The definitions of Aliases are where values are associated to Aliases using the Alias Table.

Aliases can be defined both at simple object level, composed symbol level and also at screen level.

The Alias table is saved in the object's xml code or that of the screen's in which the object was edited.

For example, when creating a composed symbol, an Alias Table can be defined for each individual symbol component and one for the symbol container.

The Alias resolution is done starting from the lowest level to the highest level. For example, If an alias has been inserted in a component, its value will be searched for in that component's table. If the value is not found, it will then be searched for in the table of the parent symbol's containing it. If unsuccessful, the search will continue until the final symbol has been reached (the symbol may be composed of several embedded



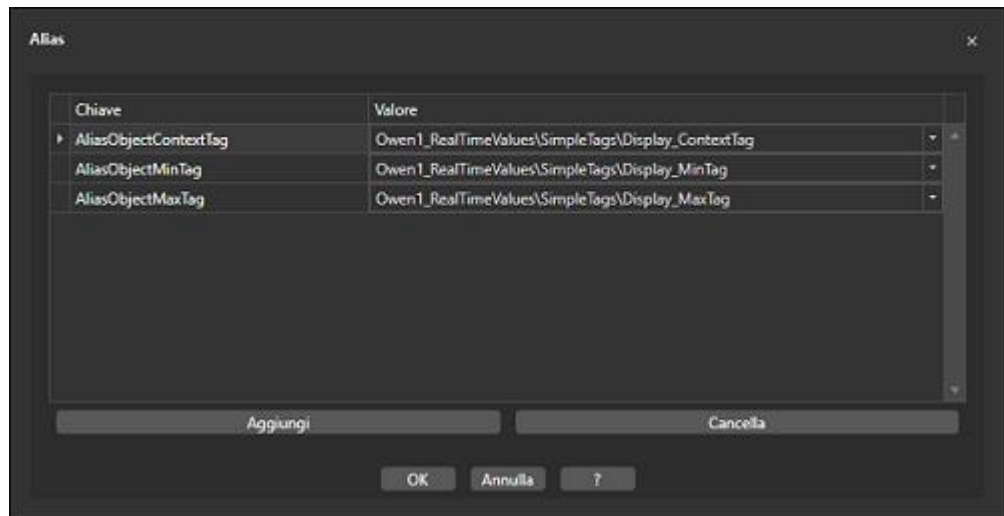
sub-symbols). If the search for the Alias's value is still not resolved, it will then be searched for in the Table defined in the Screen.

When an Alias Table is opened within a object/symbol, Movicon will run a control check to verify whether any Aliases have been inserted in the properties of the object/symbol and if found will report the list of the Aliases to that table. At this point onwards, it will be possible to edit the Alias Table by associating the values to the proposed keys. It will also be possible to insert new Aliases, remove or edit existing ones manually.

Alias Tables for Screens, on the other hand, are opened without anything in them needing you to insert all those you wish to use in them manually.

Alias Tables can be opened with the 'Edit Alias' command which is available from the object's or screen's contextual menu or by right clicking on the actual object or screen.

The Alias Table will appear as below:



The table is composed of two columns. The first one is the 'Key' column which shows the Alias names without the double angled brackets. The second one is the 'Value' column which shows the value to be replaced with the Alias. The value may be a tag or part of it or an expression element.

### Editing/Inserting Aliases

When the Alias Table is opened at object or symbol level, Movicon will search for the Aliases inserted in the symbol and display them. It is possible to associate a value to an Alias by clicking on the field to enter in edit mode or by using the drop-list to open the list of Server tags to select. Otherwise you can right click to open a popup menu and perform the following operations:

- **Cut:** removes the selected Alias row and copy it to the Windows clipboard
- **Copy:** copies the selected Alias row to the Windows clipboard
- **Paste:** Pastes the previously copied Alias row with an incremental index in the Key name.

There are also the 'Add' and 'Cancel' buttons in the Popup window which can be used to add new Alias Table rows or remove the one selected.

### Substituting Aliases at runtime

During runtime when objects/symbols are being loaded, any existing aliases will be substituted statistically with the value defined in the object's or symbol's table containing it, or, ultimately, with the value defined in the screen's table.

## Aliases in the Dynamic Property Inspector window

Aliases of objects or symbols are also displayed in the 'Dynamic Property Inspector' window by means of which they can also be modified.

## Managing Aliases using Basic Script

The Alias Table of a screen can also be access at runtime using the "GetVariableValue()" and "SetVariableValue()" Basic Script functions. By means of using these functions you can read or write the tag corresponding to the Alias specified in the function for example. The syntax to use would be:

```
Document.GetVariableValue("<<AliasContextTag>>")
```

```
Document.SetVariableValue("<<AliasContextTag>>", 10)
```

In this case, the two functions execute the read and the write of the tag associated to the <<AliasContextTag>> Alias in the Screen's Alias table.

Furthermore, by means of the Document and Entity interfaces, you can also access the list of Aliases using the object's or symbol's "MapAlias". In this way, it will be possible to access the list of defined Aliases and interact with them. However, even though the Alias list can be modified during runtime to become a new list, it cannot be activated in runtime.

## Managing Aliases in Dropping Code

In order for the Basic Script functions to access the map of Aliases of an object, they can also be used in Dropping Code of Symbols from the library. This will make it possible to customize symbols when inserting them on screen from the Symbol Library.

In the below code example, the "AliasContextTag" Alias is valued by associating it with the "Folder1\Variable1" tag while inserting the symbol from the library:

```
'#Language "WWB.NET"

Public Const DISPLAY_ENTITY_MERGE As String = "DisplaySymbol-@-ObjDisplay"
Public Const DISPLAY_ENTITY_LINK As String = "-@-DisplaySymbol-@-ObjDisplay"

Dim objDisplayEntityName As String
Dim objDisplayEntity As ScreenSettings.Entities.ScreenEntity

Public Sub DroppingCode_Customizing(ByVal sender As Object, ByVal e As System.EventArgs) Handles DroppingCode.Customizing

    'Check if symbol was dropped as link (ContentControl) or as Merged Code
    If InStr(Entity.ContainedObject.ToString, "System.Windows.Controls.ContentControl") > 0 Then
        'Symbol was dropped as Link
        objDisplayEntityName = Entity.EntityName & DISPLAY_ENTITY_LINK
    Else
        'Symbol was dropped as Merge Code
        objDisplayEntityName = Entity.EntityName & DISPLAY_ENTITY_MERGE
    End If

    'Check if Object is present in MapScreenEntities
```

```

        If
NotEntity.Document.MapScreenEntities.ContainsKey(objDisplayEntityName) Then
        MsgBox "The object '" & objDisplayEntityName & "' was not
found in the MapScreenEntities!" & vbLf & "The 'Dropping Code'
execution will be aborted!", vbCritical, "Movicon.NEXT"
        Exit Sub
    End If

    'Get MapScreenEntity for Display object
    objDisplayEntity =
Entity.Document.MapScreenEntities(objDisplayEntityName)
    'Set "AliasContextTag" in Display Alias Table
    objDisplayEntity.MapAlias.Item("AliasContextTag") =
"Folder1\Variable1"

End Sub

```

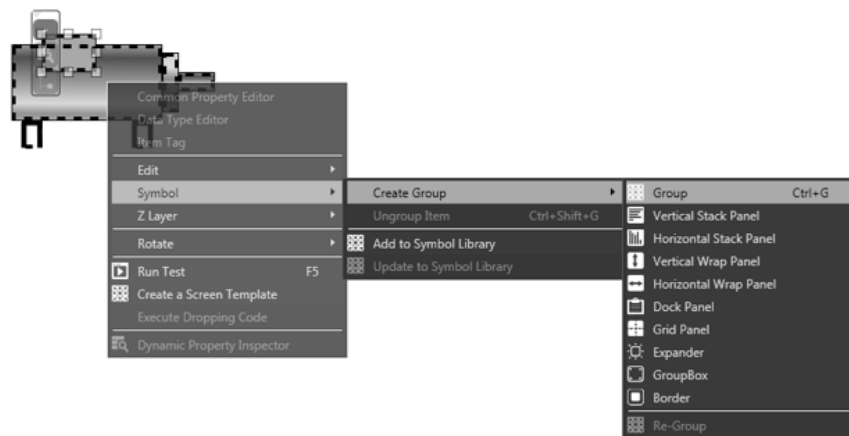


## 4. Symbols

### 4.1. Creating Composed Symbols

The Movicon object graphics can be grouped together in order to create one Symbol composed of different elements. This mechanism is very handy when needing to use the same composed Symbol several times in the project. These Symbols can also be added to the "Symbol Gallery" for later use in other projects.

To create a composed symbol simply select a object or symbol group on screen and use the 'create group' command from the "Screen Manager - Grouping" or from the contextual menu which opens with a right mouse click on the selected elements ("Symbol - Create Group" menu).



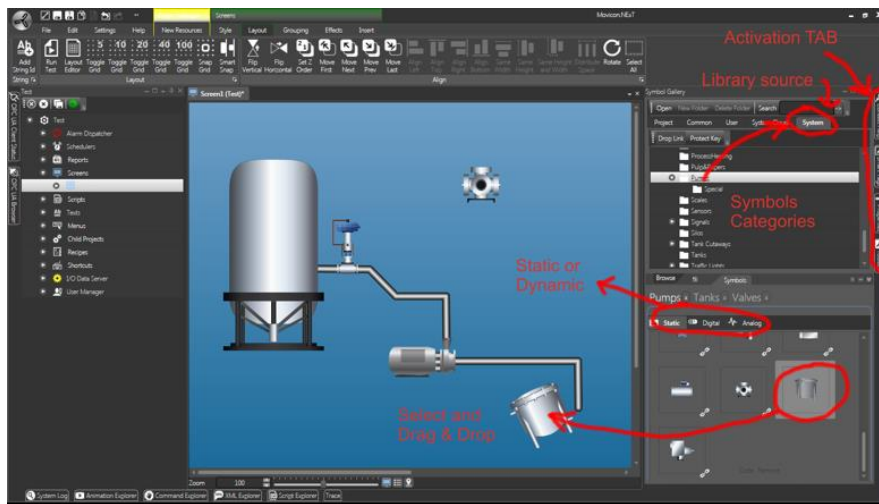
Once you have created the symbol by grouping the selected components together, you can then add it to the Movicon "Symbol Gallery" by using the "Add to Symbol Library".



When you associate a tag to a composed symbol, it will automatically be inherited by the other symbol's components if not already associated with their own tag.

### 4.2. Symbol Gallery

The Movicon Symbol Library contains ready-to-use predefined and configured static and dynamic symbols. This library can be accessed from the screen using the Symbol Gallery tab located to the right of the workspace.

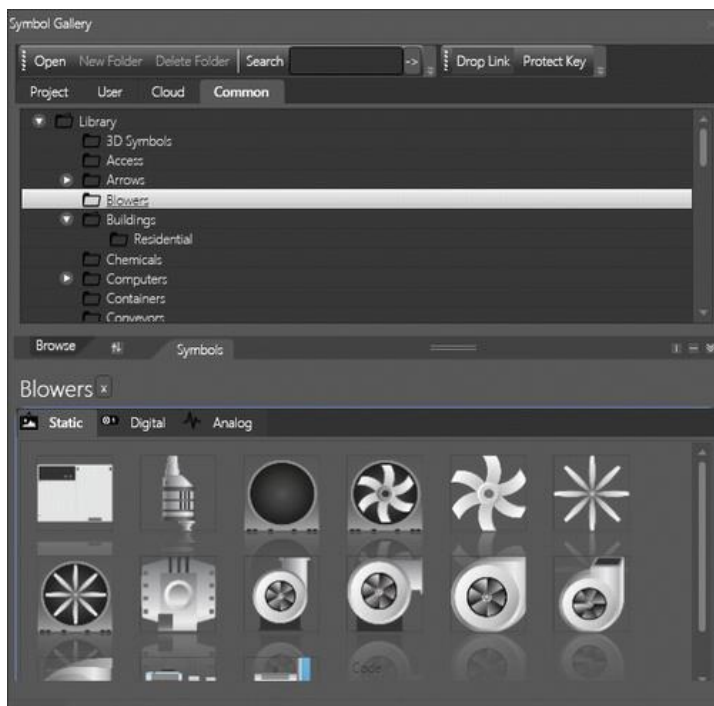


The "Symbols Library" contains graphical design symbols sub-divided into different categories. This library is activated using the Symbol Gallery tab on the right of the screen and can be kept visible or positioned to where pleased within the workspace.



**IMPORTANT!** The Automation Platform.NexT symbols have been designed by Progea using the vector drawing techniques and they are based XAML. However, the symbol library provided with the platform is encrypted and protected by an activation code (Copyright). Therefore the XAML code of each symbol is not displayed unless activated by a code purchased with a sales agreement with Progea.

The symbols are subdivided into different categories in the "Symbol Gallery" window. Each category contains a series of symbols divided into groups using the tree structure:



The "Symbol Gallery" contains Symbols divided into different categories and groups.

Objects and symbols are inserted on screen according to the window from which they are selected.

#### 4.2.1. Inserting symbols from the Symbol Gallery

To insert a symbol using the Movicon Symbol Gallery proceed as follows:

1. Open the screen of interest within the Movicon workspace
2. Activate the Symbol Gallery if not already opened in the foreground
3. Select the one of the Categories, e.g. "Project", "User" or "Common" from which to withdraw a symbol.
4. Select the symbol from the tree structured symbol list and click on the icon to expand the group
5. Select the symbol desired from the bottom part of the window symbol previews are displayed and drag and drop it on screen

Custom symbols created by the user can also be added to the Symbol Gallery. These symbols can be added to the "Project", "User" or "Common" libraries.



Symbols can be referred to different sources that can be selected from different selection TABs:

- **System:** a selection of symbols installed with the Movicon.NExT module
- **System Cloud:** a selection of Progea Symbols available on the Cloud
- **Common Users:** a selection of symbols that can be defined as pleased by users. These symbols can reside in system folders defined by the user or in the Cloud.
- **Current User:** a selection of symbols that can be defined by the current user as pleased. They can reside in a system folder as defined by the user or in the cloud.
- **Project:** a selection of symbols relating to the project and which reside in project folders.



The "**Project**" category is saved in the "Symbols" sub-folder within the project folder. In this way the symbols inserted in this category are only available from within the project itself.

The "**User**" category is saved in the document folder belonging to the current user:

"C:\Users\CurrentUser\AppData\Roaming\Progea\Movicon.NExT\Symbols\"

In this way the symbols inserted in this category are available to all projects but only for the current Windows user.

The “**Common**” category is saved in the program documents folder and to be more exact in:

“C:\ProgramData\Progea\Movicon.NExT\CommonSymbols\”

In this way the symbols inserted in this category will be available for all projects and all Windows users.

## Dropping Symbol Code

Once a symbol has been inserted in the library, its associated script code can then be edited. When the symbol is insert on screen, the script code will then be executed. The scope of this script code is to allow some of the symbol's characteristics to be customized, such as assigning a Tag or modifying one of its properties. The “DroppingCode\_Customizing” event is available in the script code and is automatically executed when the symbol is inserted on screen. The code needed to customize the symbol can be then inserted in this event.

To access the window used for inserting script code, select the symbol while in the library and use the “Code” command (the command can be found in the contextual menu, that appears when right clicking on the symbol, or in the button at the bottom of the library).

This function is very handy for setting the symbol's statical properties. You can do this by means of using the ScreenEntity's Props() function to set the symbol's string type statical properties. If these properties are set while inserting the symbol by means of using the Code contained in the Dropping Code, they will permanently remain in that symbol when they are saved together with the symbol's characteristics. In addition, these properties can also be accessed in read and write at runtime.

Substantially, the Dropping Code is useful for customizing Library Symbols when being inserted on screen.

Once the symbol has been inserted on screen, the Dropping Code can then be executed once again by using the “Execute Dropping Code” command from the contextual menu that appears when right clicking on the symbol. This will allow you to further modify the symbol's properties after the symbol has been inserted on screen.

Below is an example of how simple Dropping Code is used to add three “Props” to the symbol:

```
Public Sub DroppingCode_Customizing(ByVal sender As Object, ByVal e As
System.EventArgs) Handles DroppingCode.Customizing
    **** Create Props ***
    If Not Entity.Props.ContainsKey("Prop_Display_MeasUnit") Then
        Entity.Props.Add("Prop_Display_MeasUnit", "mm")
    End If
    If Not Entity.Props.ContainsKey("Prop_Display_PrecisionDigit") Then
        Entity.Props.Add("Prop_Display_PrecisionDigit", "1")
    End If
    If Not Entity.Props.ContainsKey("Prop_Button_Text") Then
        Entity.Props.Add("Prop_Button_Text", "Increase")
    End If
    *****
    'Force Screen to be Saved
    Entity.Document.NeedsSave = True
```



## 4.3. Symbol Library Concepts

This paragraph aims to provide important information about the concepts on using the Movicon.NExT Symbol Library containing vectorial graphics with exceptional designs and quality created by Progea and included in the Movicon.NExT project.

### Symbol Protection

The Movicon.NExT Symbol Library is the fruit of industrial design work realized by Progea, and protected by Copyright. Therefore, the symbols provided in the library are protected and can only be used for the purpose of static or dynamic graphical representations on Screen. These symbols can be assigned with chosen colors, dynamic animations and relative tags according to specific dynamic representations one wishes to achieve.

These symbol cannot be decomposed or their XAML disposed for advanced customizing.

A User Licence must be purchased from Progea containing a site Key code to unlock the protected library in order to access each symbol's structure. The symbol's structure it made up of different components that can be ungrouped in order to access their XAML code.



Please be reminded that when using a site key to unlock the library, its symbols remain subject to the type of User License you have purchased and are protected by Progea Copyright. You must not share or disclose the site key protection code or its use with others. You are only allowed to use these symbols within Movicon projects and no where else.

### Link or Merge?

The following concepts on how the project functions when managing symbols must be taken into consideration when inserting a symbol from the system library onto a screen.

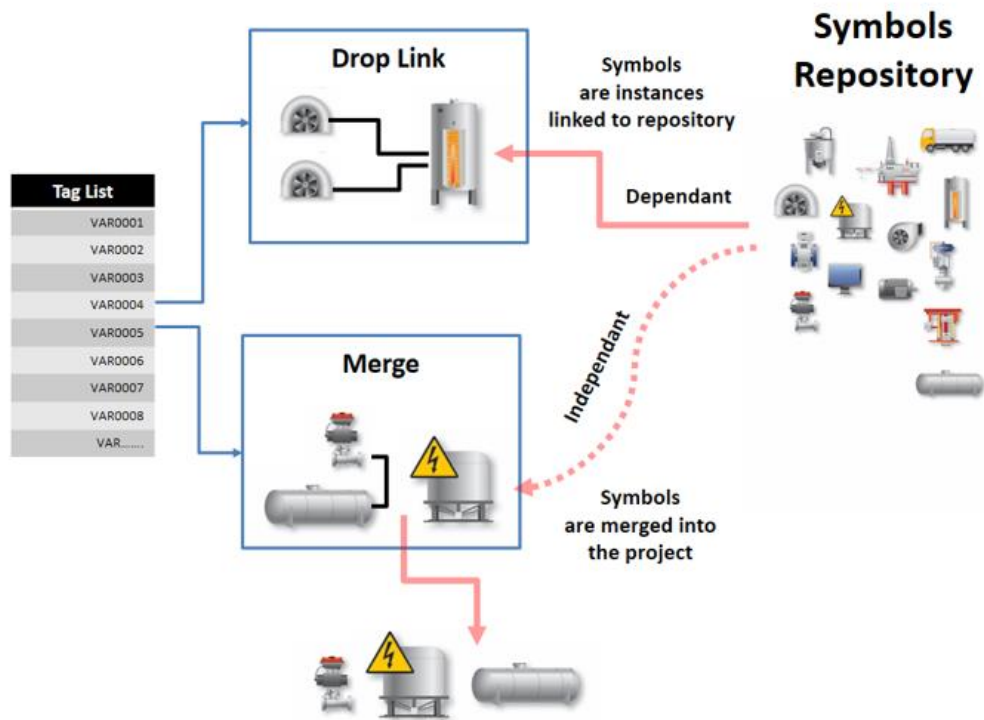
#### Drop Link

Symbols are inserted for default by using a "Drop Link", meaning that an instance will be created for the symbol on screen which will remain linked to the system symbol repository, collocated in the project's symbol installation folder (for default in C:\ProgramData\Progea\Movicon.NExT\Symbols).

This means that the screen will display the local instance of the symbol whose file remains in the repository. Changes made to the corresponding repository file will consequently update all the instances of the symbol in the project.

#### Merge

When using the 'Merge' option, the symbol will be physically copied and integrated in the project. Therefore it will become a local project symbol and independent from the Symbol Library and its Repository.



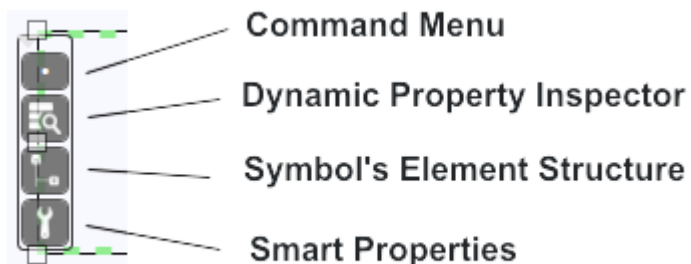
*This scheme shows the difference between managing symbols 'Drop linked' to the repository and those 'Merged' with the project.*

From a graphics point of view there are no differences in the symbol's representation and animation. However, what changes is the method used to design engineer symbols and their permanency: When using the Drop link method, the symbols are linked to the repository only and not the project. When using the Merge method, the symbols are not linked to repository and are independent due to the fact that they have been inserted in the project.

The choice between the two methods is subjective.

## 4.4. Selecting Composed Symbol Components

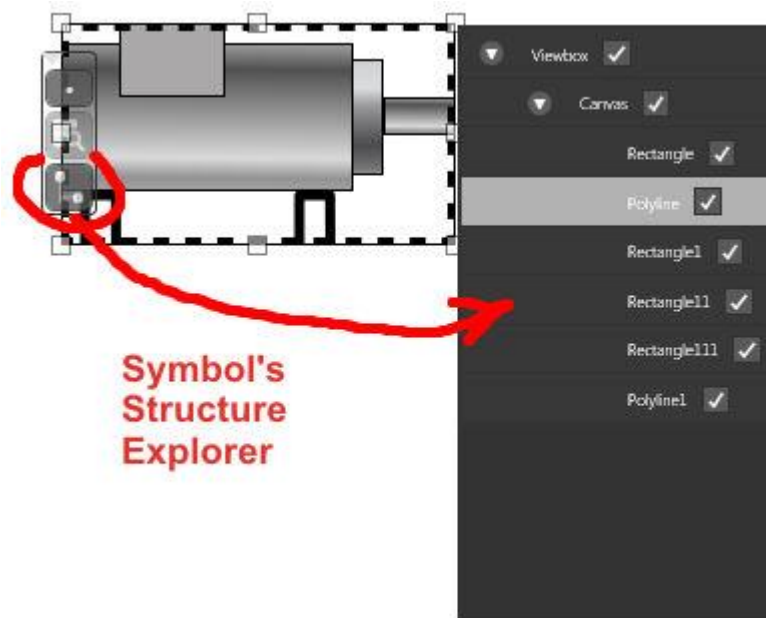
When selecting an object within the screen, it will become highlighted with a moving dashed outline. A menu containing icons will be made available on the left borderline of the selected object as shown in the image below:



By using the icon button for selecting symbol structure elements, as shown below, it will be possible to select the composed symbol's individual components without having to ungroup the symbol to access them.

Selecting one of the items listed on the menu, which correspond to each one of the symbol's elements, will highlight it within the composed symbol.

Once selected, the element's static or dynamic properties can be assigned by using the properties window or the Common Property Editor window.



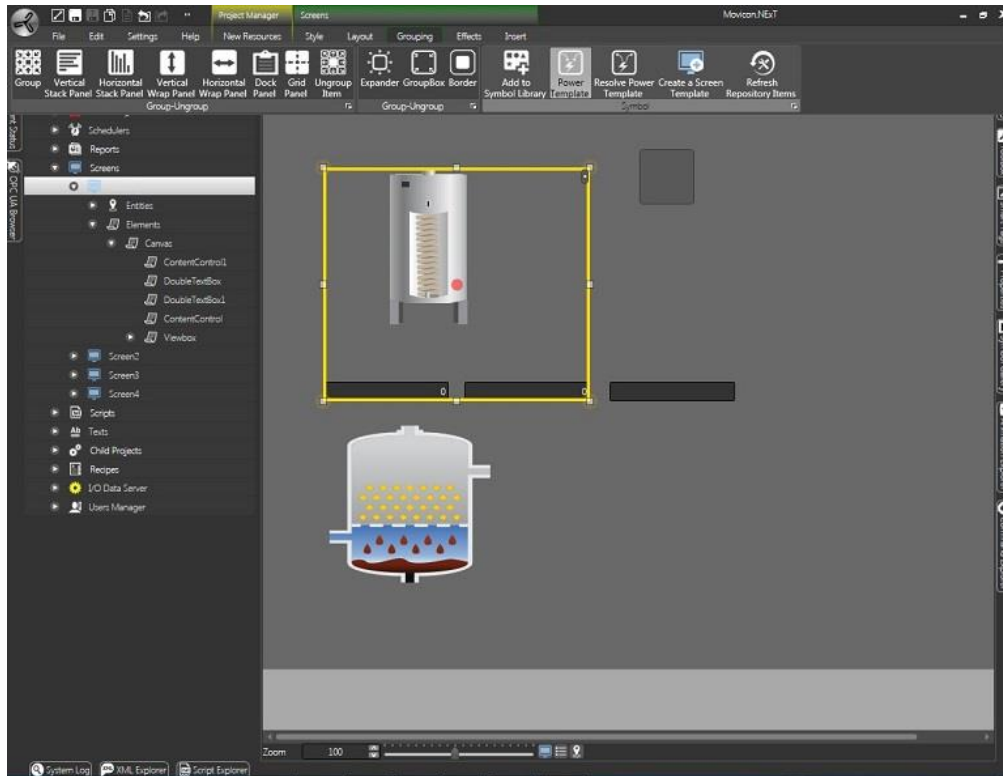
As an alternative this procedure can also be done by clicking on the component while pressing down the "CTRL + SHIFT" keypad keys.

## 4.5. Power Templates

The Power Templates are symbols defined as prototypes to which elements are assigned as variable members of structure variables and then created by using the 'Power Template' command located in the 'Grouping' ribbon. This command abstracts the symbol from the 'Structure Variable' instance and assigns its members by referring to the "Structure Prototype" only. In this way, once the symbol has been dragged from the Symbol Library and dropped on screen, it can then be assigned with a new 'Structure Variable' that must be defined with the same original 'Prototype', and the symbol will use this structure variable as a reference point to assign values. The Power Templates are very handy to have when using the same symbol several times in the same project whereby different variables are used for each instance. However, these variables must be structure type defined for the same prototype.



Attention! Certain viewer objects, such as the 'Trend' and 'Data Analysis' for example, are not editable at Runtime when they are part of a Power Template.



*Use the 'Power Template' command from the 'Grouping' Ribbon to create a symbol as a Power Template*

The basic steps to creating a Power Template are:

1. Define a Structure Prototype with the necessary members.
2. Create one or more Structure Variables defined with the Prototype you have just created.
3. Insert the components needed to build the final Symbol and assign them each a Tag as a member of one instance variable (the members must all belong to the same instance variable).
4. Create the symbol as a whole using the 'Group' command from the 'Grouping' ribbon (the type of grouping may also be different, for example you can create the symbol as a simple Group or as a 'Border', etc.)
5. Launch the project's Server. Take care when doing this as this plays an essential role in creating the Power Template successfully.
6. Create the Power Template using the 'Power Template' command from the 'Grouping' ribbon. upon terminating this procedure a Power Template Prototype window will open showing the associations between the objects, system and the prototype's Tags. Confirm to close window.
7. Now add the Power Template you have just created to the Symbol Library using the 'Add to Symbol Library' command from the 'Grouping' Ribbon.
8. At this point the process to create a Power Template operation is complete.
9. Now insert one or more Power Template symbols on screen and drag a different Structure Variable onto each symbol. The Structure Variables can either be those created previously or you can always create new ones as long as you use the initial Prototype.
10. Launch the project into run mode and check to see if each Power Template symbol refers to the Structure Variable assigned them as default tag.

As stated before, the creation of a Power Template relates to the paths of the variables that it has been associated with. The resolution of the prototype's instance associated to the symbol is performed in runtime when loading the symbol. Tags may be slower connecting to symbols than they are connecting to simple objects.

To speed up the runtime operations you can inverse the operations to create a Power Template by using the 'Resolve Power Template' command. In this case the Power Template is 'resolved' by design in respect to the Instance Variable associated to it and as a consequence the variable paths within it will be made explicit. Obviously in this case, the Power Template will lose its functionality and will function as a normal symbol to all effect.

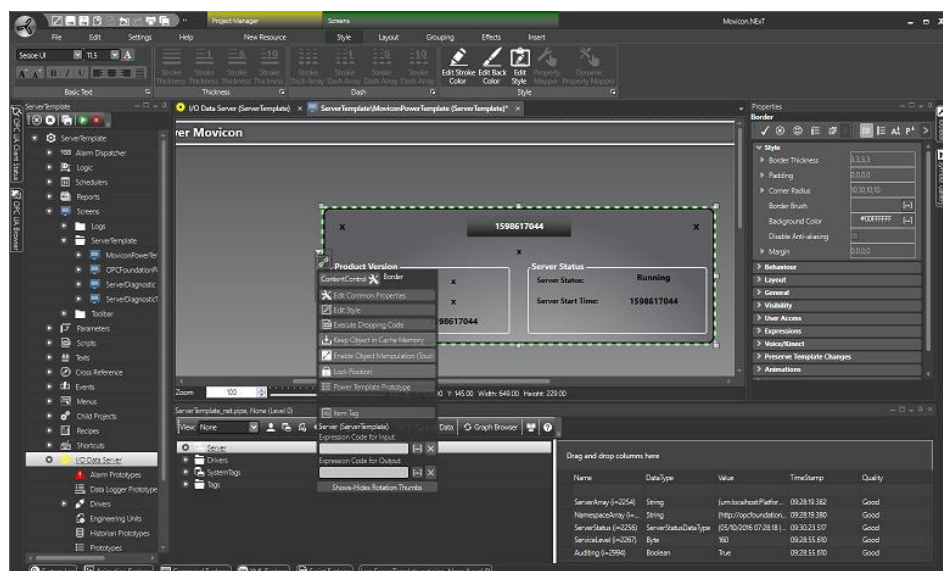
In order to for the 'Resolve Power Template' operation to have effect, certain things need to be done. If the Power Template was added to the library as "Merge Code", simply apply the 'Resolve Power Template' command making sure to startup the Server beforehand. If, however, the Power Template was added to the Symbol Library as 'Drop Link', you will first need to make sure that the 'Preserve Tag' property had been checked for each individual Symbol element that has been associated a tag. Once you have done this, startup the Server and apply the 'Resolve Power Template' command. Once the operation to resolve the Power Template has finished, you can then remove the structure variable associated to the symbol as this will no longer be used.



The Power Template identifies the Prototype Structure used based on its NodeID. It is for this reason that if you wish to use The Power Template in a project that is different from the one in which the Power Template was created, you will need to export the Prototype Structure from the original project and import it into the new project. In this way the imported Prototype Structure will retain the same NodeID and be recognised by the Power Template.

## Power Templates with Server system tags

Power Templates can be created by associating system variables from the OPC UA Server. These are diagnostic and system variables that the Server publishes but which are not part of the project's Tag List. This may result very handy for displaying information from a Server that may belong to the actual project or from a different OPC UA Server. In this case, when dragging the Power Template on screen, the Server object must be assigned to the template as the default structure (drag the "Server" object from the "OPC UA Browser" window onto the template).



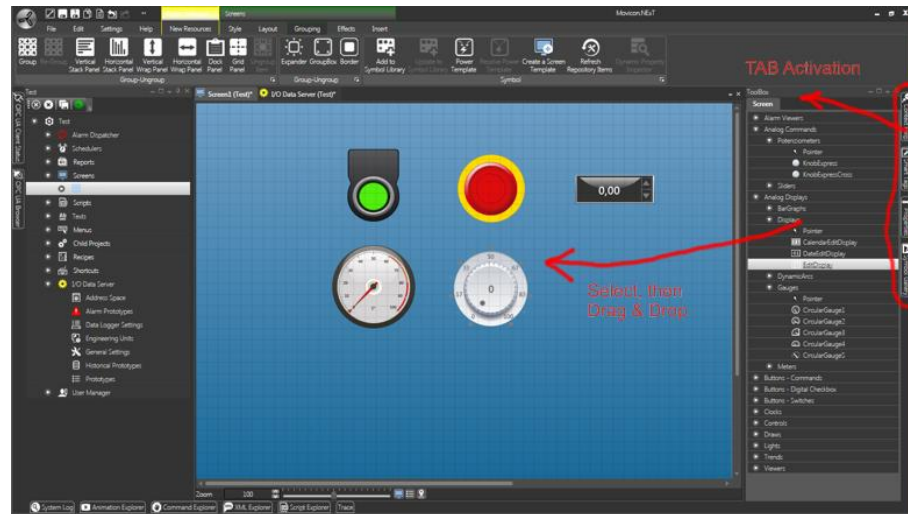


Associating the 'Server' to the template as the default structure, will make the symbol's container object border show in red. This is because the Server does not have a "Quality" property like the Tags do. In order to stop the red border propagating to the symbol's elements as well, you will need to create the symbol as "Border" and not as "Group". This is because the graphical effects applied to the container for 'Group' Symbol types are also propagated to the child elements as well.

# 5. Toolbox Object and Properties

## 5.1. Toolbox

Movicon has a library called the Toolbox which is full of ready-to-use, predefined and configured objects. This Toolbox can be accessed from the screen using the Toolbox tab located to the right of the workspace.



The "Toolbox" window contains Objects and Controls subdivided into different groups. It is activated using the relative tab on the right hand side of the screen and can be kept open or positioned as pleased within the workspace.

### 5.1.1. Inserting objects from the Toolbox

To insert an object from the Movicon Toolbox follow these quick and simple steps:

1. Open the screen of interest within the Movicon workspace
2. Activate the Toolbox if not already opened in foreground
3. Open the object group desired by clicking on the icon to expand the group
4. Select the object you wish to insert on screen
5. Position mouse pointer with attached object onto the desired area within the screen and click to release the object on that point.
6. You can also drag and drop the object on screen directly from the Toolbox as well.



Each object inserted on screen can be configured as pleased using its properties by means of the Properties Window.



If you wish to insert static text in the object's title you can get text to go to the beginning of the row underneath by inserting the "\n". For example, by digiting the following row:

**Flour\Egg\nSugar**

the result will show the text on three rows as follows:

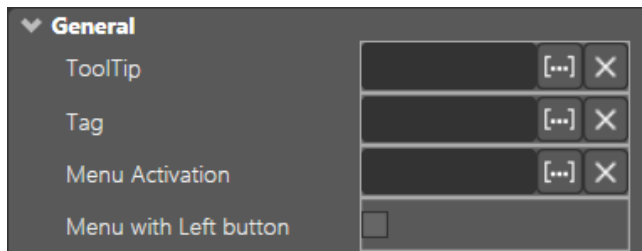
**Flour**  
**Egg**  
**Sugar**

However, when you use an ID String, you will only need to manage text on as many rows needed directly in the string table.

## 5.2. General Draws and Object Properties

Most of the draws and objects from the Toolbox are configurable using the General properties which they all have in common. These properties have general functionality characteristics which describe a major part of the objects.

- If these properties are not displayed, check to see if the "Easy Mode" function is deactivated using the "Easy-Filter Mode" button from the selected object's property taskbar.



### General

**UTC Source Timestamp:** Enabling this will show the UTC Source Timestamp column, otherwise the local PC time will be used as timestamp column.

**Menu with Left button:** This is used to display the object's contextual menu using the left mouse key.

**Menu Activation:** This is used to set the name of the contextual menu that displays when clicking on the object with the right mouse key.

**UseIntellisense:** This is active for default and when deactivated you will need to use "GetVariableValue" and "SetVariable" to access variables. This option is deactivated to improve performances when opening screens.

**ToolTip:** This is used to set the text to display when the user keeps mouse pointer on the control.

**Unit Converter:** This is used to set a converter type for variable associated to the object. Before applying this property set the Unit Converter resource accordingly.

**Show Tooltip When Disabled:** Enabling this will show the Tooltip when the object is disabled.

### Layout

**Horizontal Content Alignment:** Sets the horizontal alignment for all the contents within the object.



**Vertical Content Alignment:** Sets the vertical alignment for all the contents within the object.

**Width:** This allows you to change the object's width.

**Height:** This allows you to change the object's height.

**Horizontal Alignment:** Sets the object's horizontal alignment.

**Vertical Alignment:** Sets the object's vertical alignment.

**Position X:** This is used to set the object's horizontal position in pixels.

**Position Y:** This is used to set the object's vertical position in pixels.

**Lock Object Position:** Locks the object into position on screen.

#### Tag

**Tag :** This is used to display and assign project tags to the selected object.

## 5.3. Draws and Object Behaviour Properties

The major part of drawings and objects from the ToolBox have the same 'Behaviour' properties. This group of properties are used to manage some commands that will permit users to perform certain operations on the object by selecting the various functions.



If this property group is not shown, make sure that 'Easy Mode' function has been deactivated using the "Filter-Easy Mode" button from the selected object's property bar at the top.

#### Behaviour

**Force as Dynamic on Web Client:** Enables/disables the object's animations on the Web Client side.

**TAB Stop:** When enabled permits the selection of the object at runtime using the TAB button.

**Flow Direction:** This is used to set in which direction the text and all the other UI elements go within the object container.

**Is Interactive:** Permits interactivity with the object's user interface (UI) using the click of the mouse.

**Is Focusable:** Enables/disables the focus on object.

**Enable Touch Manipulation:** Activates the object's manipulation options at runtime.

**Enable Mouse Over:** Enables the "Mouse Over" option for moving the move on object at RunTime.

## 5.4. Preserve properties of local Draws and Objects

It is important to remember that for default graphical symbols collocated in screens there are instances of original symbols placed in the Symbol Libraries. These symbols

can be managed in the library using the parent symbol concept whereby any modifications to the original library symbol will be propagated to those placed on the project screens.

However if you wish to keep a symbol, that has been inserted and configured in the project, unaltered and therefore independent (despite the fact that it still remains an instance of the original symbol) you will need to use the 'Preserve' properties that have been provided for each symbol or object.

In this way, if you decide to preserve one or more properties, the symbol in question will retain its local configuration independently from the modifications applied to the parent symbol in the library.

### Preserve Template Changes

**Preserve Style:** when enabled this property preserves the local style property values when the object is updated from the library.

**Preserve Code:** when enabled this property preserves the basic script code of the local style properties when the object is updated from the library.

**Preserve Colors:** when enabled this property preserves the local color property group values when the object is updated from the library.

**Preserve Size:** when enabled this property preserves the local size property group values when the object is updated from the library.

**Preserve Commands:** when enabled this property preserves the list of command executions when the object is updated from the library.

**Preserve Animations:** when enabled this property preserves the animation list when the object is updated from the library.

**Preserve Menu:** when enabled this property preserves the set contextual menu when the object is updated from the library.

**Preserve Expression:** when enabled this property preserves the value conversion expression when the object is updated from the library.

**Preserve Security:** when enabled this property preserves the Security property group values when the object is updated from the library.

**Preserve Visibility:** when enabled this property preserves the visibility property group values when the object is updated from the library. As regards to project symbols created by the user, this option preserves their 'Visibility Layer Level', 'Zoom Level Visibility X' and 'Zoom Level Visibility Y' properties.

**Preserve Variables:** when enabled this property preserves the connected tag when the object is updated from the library.

**Preserve Text:** when enabled this property preserves the inserted text string when the object is updated from the library.

## 5.5. Draws and Objects User Access Properties

Each object and drawing can have its data, commands and visualization protected by allowing certain users to access them.

This is done by setting their appropriate User Access Properties according to the Users and Passwords set in the project.

### User Access Properties

**Access Level for Runtime Config.:** This is used to set the allowed user access level to edit the object in runtime.

**Mask Access Area for Runtime Config.:** This is used to mask the access area to allow the user to edit the object in runtime.

**User Access Role:** This is used to set the user's role for accessing the object.

**Access Level From Tag:** This is used to set the user level requested for interacting with this object by inheriting it from the variable linked to it.

**Access Level:** This is used to set the user access level needed for interacting with object.

**Readable Access Mask:** This is used to set the Access Mask Level in Read for the object. This Access Mask Level will not show if the access mask of the user connected at that moment does not correspond to the one set in the object.

**Writable Access Mask:** This is used to set the Access Mask Level in write for the object. This variable value cannot be modified if the access mask of the user connected at that moment does not correspond to the one set in the object.

## 5.6. Animation and Command Properties

These two property groups are used to access the respective Animations and Command property groups belonging to the selected object.



If you cannot see these two groups, check that the "Easy Mode" is enabled by using the "Filter-Easy Mode" button on the selected object's property bar.

### Animations

**Animations:** This property opens the window with a list of dynamic animation commands that can be associated to the object.

### Commands

**Commands:** This property opens the window with a list of dynamic animation commands that can be associated to the object.

**Always execute commands:** By setting this property with a true value, the commands defined on a list associated to a button object for instance, will be always be executed. When this property is set to False, the commands will be executed only if all of them are executable. For example, if a 'Tag Value' command has been included as one of the commands for setting a value of a Tag that has its ExecuteAnyEnabledCommands property set to False, the command list will only be executed if the Tag is connected correctly and has 'Good' quality. Conversely, if the property is set to True, even though the Tag does not have a 'Good' quality and therefore cannot be set, the remaining commands will be executed regardless.

## 5.7. Managing Expressions in Draws and Objects

Each object or drawing provided with an expression management have properties that allow you to assign expression input or out code of which result will be assigned to the value managed by the object.

- See topic relating to Expressions in Objects and the list of Function Reference Section.

#### Expression:

This is used to set the expression using Excel 2013 code whose result will be displayed in the object.

Certain codes can be used in this property for the following scope:

#### Defining one specific Bit:

To connect a specific Bit to a tag, insert the '**nBit**' value (i.e. **.1** to indicate the Bit 1)

#### Defining one specific Array Element:

To connect a specific array element insert the '**nElem**' value (i.e. **[1]** to indicate the item 1)

#### Expressions with Bit Variables

Below is a list of some examples of expressions using a Bit Variable:

= **NOT([x])** this expression will execute the NOT logic of the variable associated to the object's Tag property.

= **NOT([Variable1])** this expression will execute the NOT logic of the *Variable1* variable.

=**AND([Variable1],[Variable2])** this expression will execute an AND logic between the *Variable1* and *Variable2* variables.

=**OR([Variable1],[Variable2])** This expression will execute an OR logic between the *Variable1* and *Variable2* variables.

#### Expressions with Word Variables

Below are some examples of expressions using Word type variables:

= **NOT(BITAND[x],4)** this expression will perform the NOT logic of the bit 2 of the variable associated to the object's Tag property.

= **NOT(BITAND[Variable1],8)** this expression performs the NOT logic of bit 3 of the *Variable1* variable.

=**AND(BITAND([Variable1],4), BITAND([Variable1],16))** this expression performs an AND logic between bit 2 and bit 4 (because  $2^2 = 4$  and  $2^4 = 16$ ) of *Variable1*. This type of expression is to be used as a replacement for a "**=AND([x].2, [x].4)**" expression type which is not permitted because the '**nBit**' expression is not available when used in combination with other expressions.

=**OR(BITAND([Variable1],8), BITAND([Variable1],16))** This expression performs an OR logic between the bit (because  $2^3 = 8$ ) and the bit 4 (because  $2^4 = 16$ ) of *Variable1*. This type of expression is to be used as a replacement for a "**=AND([x].3, [x].4)**" expression type which is not permitted because the '**nBit**' expression is not available when used in combination with other expressions.

=**[Variable1]+[Variable2]** this expression performs a calculation on the two variables. Other mathematical signs can also be used (**- , \* , /**).

=**ABS([Variable1]-[Variable2])** this expression returns the absolute value between the two variables.

=**EXP**([*Variable1*]) Returns the number raised to the power of number. The constant is equal to 2,71828182845904, the base of the natural logarithm.

=**SQRT**([*Variable1*]) Returns the square root of *Variable1*.

=**[*Variable1*]=[*Variable2*]** Returns true or false as a result of the comparison between the two variables.

=**[*Variable1*]<>[*Variable2*]** Returns true or false as a result of the comparison between the two variables.

=**[*Variable1*]>[*Variable2*]** Returns true or false as a result of the comparison between the two variables.

=**[*Variable1*]<=[*Variable2*]** Returns true or false as a result of the comparison between the two variables.

=**COS**([*Variable1*]) performs the cosine of *Variable1*

=**SIN**([*Variable1*]) performs the sine of *Variable1*

=**TAN**([*Variable1*]) performs the tangent of *Variable1*

= **ATAN**([*Variable1*]) performs the arc tangent of *Variable1*

#### Reverse Expression:

Allows you set the expression with Excel 2013 code whose result will be assigned to the tag connected to the object.



This property is usually used for setting the expression specified in the expression property in the reverse.

## 5.8. Using Engineering Units

**During runtime** the object's scale values can be defined dynamically using the **Engineering Units associated to the Tag variables**. When the Tag variable is associated with an Engineering Unit, the variable's Min. and Max. values will automatically be scaled and calculated according to the set conversion values during runtime. In this way, the Tag associated to the objects will automatically determine the Min. and Max. scale values of the associated graphical object.

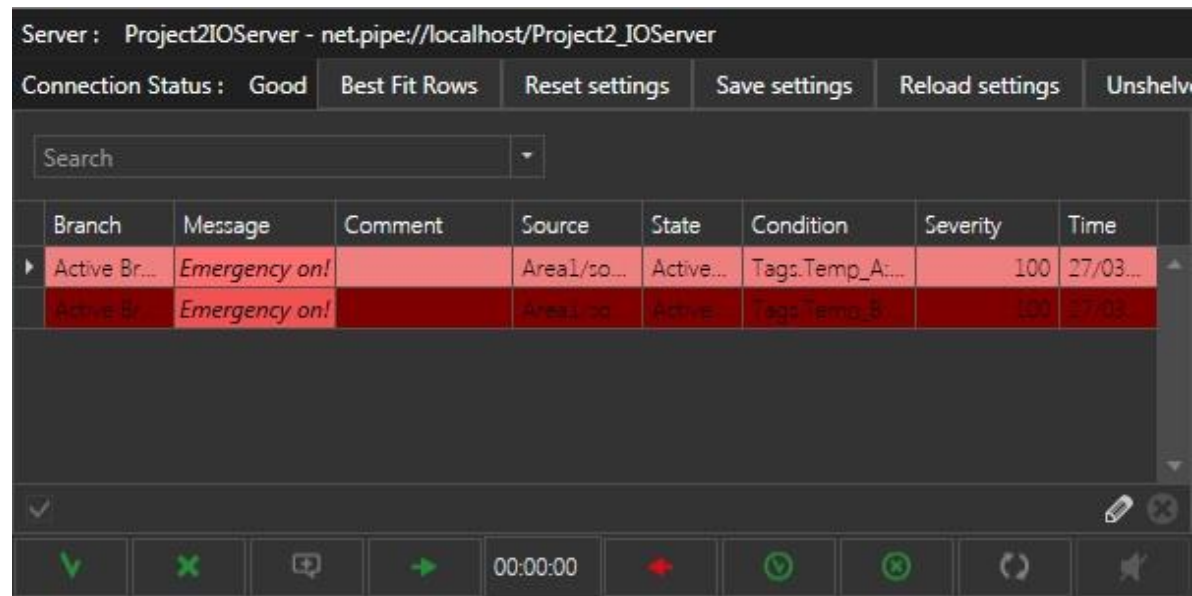
- Example: Let's take a Gauge object set with the default scale values from 0 to 100. If we associate this object with a Tag variable defined with an Engineering unit of a value from 0 to 100 converted to a value from 0 to 380, during edit mode the Gauge will display the default 0-100 value but in runtime mode the Engineering Unit will be calculated displaying the resulted scaled 0-380 value.

## 5.9. Alarms Viewers

### 5.10. Alarm Windows

The Movicon NExT Alarm Window is a configurable object designed for representing and controlling local systems or area alarms.

The "Alarm Window" is available from the "Alarm Viewers" group in the ToolBox.



Movicon NExT Alarm Window

The "Alarm Window" is inserted on screen with default settings. Once the window has been inserted on screen it can be sized as desired by dragging its borders with the mouse.



You can insert as many "Alarm Windows" as you need in the various screens as acquired.

#### Operation Commands executable in runtime from the Alarm Window

The Alarm Window has some buttons that are used to perform operations on the alarms displayed in the Alarm Window. The buttons can be displayed or hidden using the property window. They can also be displayed as icons or with text. A tooltip will display with the button's text.

<b>Ack All</b>	This command button is used to acknowledge all the previously selected alarms in the Alarm window.
<b>Reset All</b>	This command button is used to reset all the alarms in the Alarm window without selecting them beforehand.
<b>Ack</b>	This command is used to acknowledge the alarm selected in the Alarm Window.

<b>Reset</b>	This command button is used to reset the alarm selected in the Alarm Window.
<b>Add Comment</b>	This command button is used for entering a comment for the alarm selected in the window. This button is enabled only when one or more alarms have been selected. When pressing the button, a dialog window will appear through which a comment can be entered and associated to the selected alarm.
<b>Shelve</b>	This command button postpones the alarm for a later time which is set using the box located at the side of the button.
<b>UnShelve</b>	This command button is used to cancel the previous command.
<b>Refresh</b>	This command button refreshes the window's contents.
<b>Disable Sound</b>	This command button enables or disables the alarm acoustics associated to the unacknowledged alarm priorities.

## Alarm Property Window

### Alarm Window Style

**Toolbar Background:** Used to define the background color to apply to the Toolbar.

**Toolbar Foreground:** Used to define the foreground color to apply to the Toolbar.

**Auto Column Layout:** Used to enable 'Show Auto Column Layout Button' at runtime. When this property is set to False, the 'Show Automatic Layout Button' will not be enabled at runtime and therefore unusable.

**Allow Resizing:** Used for resetting the size of the columns manually in runtime.

**Show Quality Status:** Enables the connection quality status label to show in the top alarm window taskbar.

**Show Expand/Collapse Button:** Enables the Expand/Collapse button to show in the top alarm window taskbar.

**Show Group Panel:** Enables a dialog window to show by means of which list of alarms can be grouped according to the column headings dragged into the window.

**Show Search Panel:** Enables the Search Panel to show in the top alarm window taskbar.

**Show Filter Panel:** Enables the Filter panel to show in the top alarm window taskbar.

**Disable Blink Animation:** disables blink animation in alarms that have not yet been acknowledged.

**Alarm Area Font:** Sets the character type for the selected alarm area.

**Header Font:** Sets the character type for the selected alarm's column.

**Horizontal Column Alignment:** Sets the Alarm Columns to align horizontally.

**Date/Time Format:** Allow you to set a string to change the object's Date/Time style. For further information concerning custom format string types please refer to the topic on Strings.

**Use UTC SourceTimeStamp:** When using this property, the time that the alarm triggered will be displayed with the UTC SourceTimeStamp under the 'Time Column' in the Alarm Window.

### Alarm Window Buttons

**Button Size:** Sets the size of the command button.

**Button Position:** Determines the position of the command button.

**Show Best Fit Button:** Enables the Best Fit Button to show on the top alarm window taskbar. This button is used at runtime to automatically reset the column sizes within the window by adapting them to fit in the window in the best way possible.

**Show Settings Buttons:** Enables the Settings buttons to show in the top alarm window taskbar.

**Graphic Buttons:** Replaces the buttons with icons.

**Ack Button:** Displays the ACK button for acknowledging the selected alarm.

**Ack All Button:** Displays the ACK ALL button for acknowledging all the active alarms.

**Reset Button:** Displays the RESET button of the selected alarm.

**Reset All Button:** Displays the RESET ALL button to reset all the alarms showing in the alarm window.

**Refresh Button:** Displays the REFRESH button.

**Shelve Button:** Displays the SHELVE button to postpone selected alarm.

**Comment Button:** Displays the ADD COMMENT button to add comments about the selected alarm.

**Sound Button:** Displays the DISABLE SOUND button to disable the selected alarm sound.

**Execute Button:** Displays the EXECUTE button.

### Alarm Window Advanced

**Show Server Title:** Shows the currently used Server name in a label in the top alarm window taskbar.

**Add/Delete Columns in runtime:** allows columns to be added/deleted in runtime.

**Severity Required for Comment:** Prompts user to insert a comment when acknowledging an alarm with a specific severity level.

**Editable:** Allows object to be edited in Runtime.

**Connect Child Alarms:** This is used to display child project alarms in the same parent project alarm window. The child project alarms will display in the 'Child Project Name' column along with the name of the project that they belong to.

**Text Wrapping:** This is used to select which text wrapping mode to used for displaying the list of alarms represented in the alarm window.

- **WrapWithOverflow:** the text will jump down to the start of the next line when reaching the maximum fixed width occupied by the column. When the text is one continuous string, it will be split over two lines once reaching the maximum column width.
- **NoWrap:** no line breaking will occur and the text will be broken off when overflowing column width edge.
- **Wrap:** line break occurs when text reaches the maximum column width edge and long words will be split over two lines.



### User Access

See topic on "Draws and Object User Management" concerning objects and drawings that define the security parameters based on the project's User and Password management.

### General

see the topic "Common ToolBox Object and Drawing Properties".

### Style

**Border Brush:** This is used for setting the container's border colour.

**Border Thickness:** This is used for setting the container's border thickness.

**Foreground Color:** This is used for setting the colour of the letter and number characters used in the object.

**Background color:** This is used for setting the container's background colour.

**Font Size:** This is used for setting the font size used within the object.

**Disable Anti-aliasing:** Enables/disables the Anti-aliasing pixel correction function.

**Margin:** sets the object's margins

### Fonts

**Font Family:** Sets font model to be used.

**Font Size:** Sets the font's size within the object.

**Font Style:** Sets the font's style.

**Font Weight:** Sets the displayed Font's thickness or thinness ( *light*, *extralight*, *normal* etc...).

### Layout

See topic: "Common ToolBox object and drawing properties".

### Behaviour

See topic: "Objects Executions Properties".

### Visibility

**Transparency:** Sets the object's Transparency level.

**Visibility:** enables the object's visibility.

**Visible on Web Client:** makes object visible on the webclient

**Visibility Layer Level:** permits you to set the mask to determine the object's visibility.

**Zoom Level Visibility X:** permits you to set the Zoom factor on the x axis after which the object will become visible on screen.

**Zoom Level Visibility Y:** permits you to set the Zoom factor on the y axis after which the object will become visible on screen.

### Tag

See topic: "Common ToolBox object and drawing properties".

### Expressions

See topic: "Managing Expression" concerning objects and drawings.

## Preserve Template Changes

See topic: "Preserve local Draws and Objects properties" concerning original library objects



The 'Ack All', 'Reset All' and 'Refresh' buttons are available when the Alarm window is used by the **"HTML5 Web Client"**.

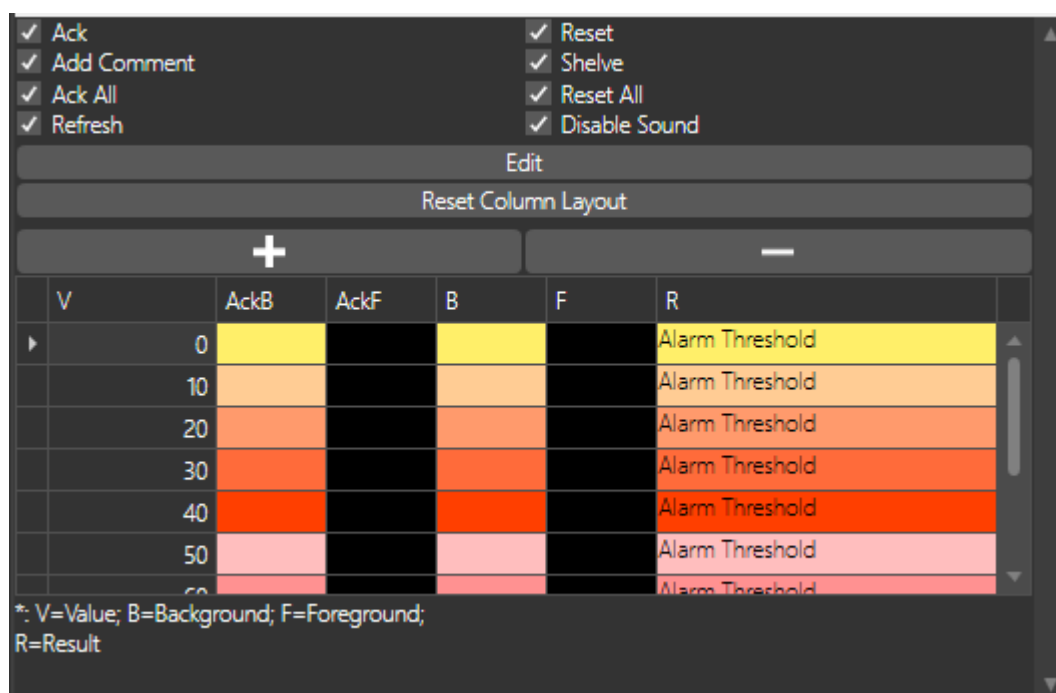
In addition the "Search" field, column resizing and filter with clause are not handled.

Alarm comments are currently not supported on the Web Client.

A web page is always shown in the web client where the 'Alarms?' can be accessed from the initial page. This page allows alarms to be handled on the Web Client to bypass limitation of Alarm Window objects published on the Web Client: the ack. and confirm of individual alarms, editing filters in alarms.

## Additional Properties (Smart Tags)

You can add or remove buttons shown at the bottom of the alarm window by using the check boxes shown in the below window.



Use the **Edit** button to preview the object and see how it will display during runtime.



The "V" value corresponds to the Priority defined in the alarm prototype.

## 5.11. Alarm Banners

The "Alarm Banner Display" object is a derivative of the "Alarm Window" and is used for displaying alarms that have been installed throughout the plant. The "Alarm Banner Display" object is available from the ToolBox's "Alarm Viewers" group.

The "Alarm Banner Display" object enables you to view the alarms row by row. Therefore alarms can be scrolled one at a time automatically or manually.



The 'Alarm Banner' displays the alarm text and date and time of activation (Alarm ON) only. Other information such as acknowledgment date and time, off date and time, etc. will not be shown. Unlike the 'Alarm window', this window does not come with command buttons to acknowledge or reset alarms and no comments can be entered either. It is only used for viewing information.

The "Alarm Banner" will display with its default settings when inserted on screen. Once inserted on screen, it can then be resized as needed by dragging its borders with the mouse. The alarms shown in this window are those that need to be both Acknowledged and Reset. Once these two actions have been done, they will automatically disappear from the window. Simple Messages are also shown in the window and disappear automatically in function with the alarm's ON/OFF status. In order to acknowledge and reset the alarms you will need to either use the "Alarm Window" to do this or insert two command buttons on screen, in addition to Alarm Banner object, one to use as "Ack All" and another to use as "Reset All".

### Alarm Banner Display Property Window

#### Alarm Banner Buttons

**Enable Spin:** Enables the spin buttons.

**Spin Position:** Modifies the spin button positions.

**Spin Size:** Modifies the spin button size.

**Show Refresh Button:** Enables the Refresh button.

**Spin Background:** Sets the Spin button's background.

**Refresh Button Background:** Sets the Refresh button's background colour.

#### Alarm Banner Style

**Cycling Time:** Sets the scroll time for the alarm texts (only available when the "Auto-scroll" is activated).

**Auto-Scroll:** Enables the automatic alarm scroll.

**Alarm List Default Text:** Shows set message when alarms are inactive.

**Text Wrapping:** Use this to specify whether text should continue on the next line when reaching the object's border.

**Disable Blink Animation:** Used for disabling the blink on alarms not yet acknowledged.

**Alarm Font Settings:** Sets the Alarm window's font stile.

## Style

**Border Brush:** This is used for setting the container's border colour.

**Border Thickness:** This is used for setting the container's border thickness.

**Back color:** This is used for setting the container's background colour.

**Stroke Color:** This is used for setting the color of the letter and number characters used in the object.

**Disable Anti-aliasing:** Enables/disables the Anti-aliasing pixel correction function.

**Margin:** sets the object's margins

## Fonts

**Font Family:** Sets font model to be used.

**Font Size:** Sets the font's size within the object.

**Font Style:** Sets the font's style.

**Font Weight:** Sets the thickness or thinness of how the font should be displayed ( *light, extralight, normal etc...*).

## Layout

See topic on "Common ToolBox object and drawing properties".

## Behaviour

See topic on "Objects Executions Properties".

**Alarm Source:** This field is used for selecting an Area or a source from those defined in the project's Alarms. In this way, the Alarm Banner object will display only those alarms belonging to the Area or the Source, selected here, in runtime. It is also possible to dynamically set this property in runtime using the object's appropriate "AlarmsSource" script (please refer to the property's description for further information).



When Drag&Dropping an Alarm Area or Source on the object (this command can be executed by means of the "OPCUA Server Browser" window once the Server has started up) the Area or Source will be set in the Alarm Banner's contextual Tag and will not change the "AlarmSource" property. When the Alarm Banner's "AlarmSource" property is empty, the filter will be executed according to the Area or Source dropped in the object. However, if the "AlarmSource" property has been defined, it will have priority in the contextual field and therefore the filter will be executed according to the "AlarmSource" property.

## General

See topic on "Common ToolBox Object and Drawing Properties".

## Visibility

**Transparency:** Sets the object's Transparency level.

**Visibility:** enables the object's visibility.

**Visible on Client:** makes object visible on the webclient

**Visibility Level:** permits you to set the mask to determine the object's visibility.

**Zoom Level Visibility X:** permits you to set the Zoom factor on the x axis after which the object will become visible on screen.

**Zoom Level Visibility Y:** permits you to set the Zoom factor on the y axis after which the object will become visible on screen.

**Tag**  
See paragraph on "Common ToolBox object and drawing properties".

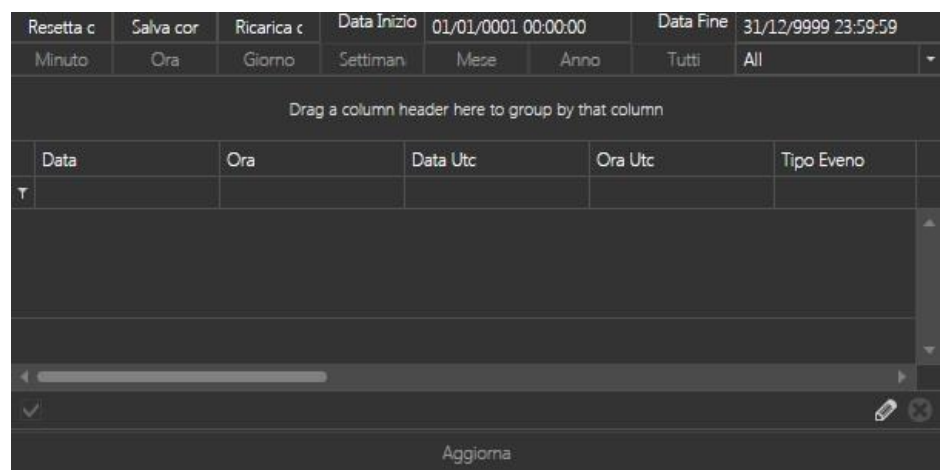
**User Access**  
See topic on "Draws and Object User Management" concerning objects and drawings that define the security parameters based on the project's User and Password management.

**Expressions**  
See paragraph "Managing Expression" concerning objects and drawings.

**Preserve Template Changes**  
See paragraph "Preserve local Draws and Objects properties" concerning original library objects

## 5.12. Historical Log Viewer

The Historical Data Viewers display the various messages recorded by the platform's event Historical.



The Historical Log records all project events (Alarms, System Messages, Driver Messages) that occurred during project runtime chronologically (loads data from the "UFUAAuditLogItem" table which contains project alarm and system events). The events that can be recorded in the Historical Log are:

- System operations (system operator manoeuvres).
- System diagnostics (autodiagnosis, Driver diagnosis).
- Project Alarms and Messages in 4 possible states (ON, OFF, ACK and RST).
- User Log On and Log Offs.
- Basic Script logic messages by means of using the "Debug.Print" function.
- Messages coming from other applications established by the programmer.

## Historical Event Log viewer Properties

### Historical Window Style

**Toolbar Background:** Used to define the background color to apply to the Toolbar.

**Toolbar Foreground:** Used to define the foreground color to apply to the Toolbar.

**Event Area Font Settings:** Used to set the font type to use in the Event Area.

**Header font Settings:** Used to set setting the font type to use in the Header area.

**Editable in runtime:** Use this to set the viewer in editable mode in order to modify it in Runtime. This option also permits the object's configuration created during runtime to be saved and recalled.

### Database Settings

**Connection String:** This is used to set the connection string to the database.

**Filter Event Type:** This is used to set the event type messages to loaded and displayed in the window.

**Max.Rows Nr.:** Maximum number of rows (records) to be shown in the window.

**Time Start:** Used to set the sampling start time.

**Time End:** Used to set the sampling end time.

### User Access

**Editing Write Access Level:** sets the access level requested by the Access Mask to allow editing during runtime.

**Editing Write Access Mask:** sets the Access Mask to allow editing during runtime.

**Access User's Role:** This is used to set the name of the user group which will interact with this object.

**Access Level From Tag:** This is used to set the user level requested for interacting with this object by inheriting it from the variable linked to it.

**Access Level:** This is used to set the user access level needed for interacting with object.

**Readable Access Mask:** This is used to set the Access Mask Level in Read for the object. This Access Mask Level will not show if the access mask of the user connected at that moment does not correspond to the one set in the object.

**Writable Access Mask:** This is used to set the Access Mask Level in write for the object. This variable value cannot be modified if the access mask of the user connected at that moment does not correspond to the one set in the object.

### Style

**Border Brush:** This is used to set the container's border colour.

**Border Thickness:** This is used to set the container's border thickness.

**Foreground Color:** This is used to set the colour of the letter and number characters used in the object.

**Background color:** This is used to set the container's background colour.

**Font Size:** This is used to set the font size used within the object.

**Disable Anti-aliasing:** Enables/disables the Anti-aliasing pixel correction function.

**Margin:** This is used to set the object's margins.

## Fonts

**Font Family:** Sets font model to be used.

**Font Size:** Sets the font's size within the object.

**Font Style:** Sets the font's style.

**Font Weight:** Sets the thickness or thinness of how the font should be displayed ( *light, extralight, normal etc...*).

## Layout

See topic on "Common ToolBox object and drawing properties".

## Behaviour

**Editable in runtime:** sets the viewer to edit mode at runtime. This option also allows the configuration of the created object to be saved and recalled.

See topic on: "Drawing and Object Behaviour Properties".

## General

see topic on: "Common ToolBox Object and Drawing Properties".

## Visibility

**Transparency:** Sets the object's Transparency level.

**Visibility:** enables the object's visibility.

**Visible on Client:** makes object visible on the webclient

**Visibility Level:** permits you to set the mask to determine the object's visibility.

**Zoom Level Visibility X:** permits you to set the Zoom factor on the x axis after which the object will become visible on screen.

**Zoom Level Visibility Y:** permits you to set the Zoom factor on the y axis after which the object will become visible on screen.

## Tag

See topic on: "Common ToolBox object and drawing properties".

## User Access

See topic on: "Draws and Object User Management" concerning objects and drawings that define the security parameters based on the project's User and Password management.

## Expressions

See topic on: "Managing Expression" concerning objects and drawings.

## Preserve Template Changes

**Preserve:** see topic on: "Preserve local Draws and Objects properties" concerning original library objects.

## 5.13. Progress Bars

The Movicon NExT BarGraphs are configurable objects that are used to display specified variable values graphically.

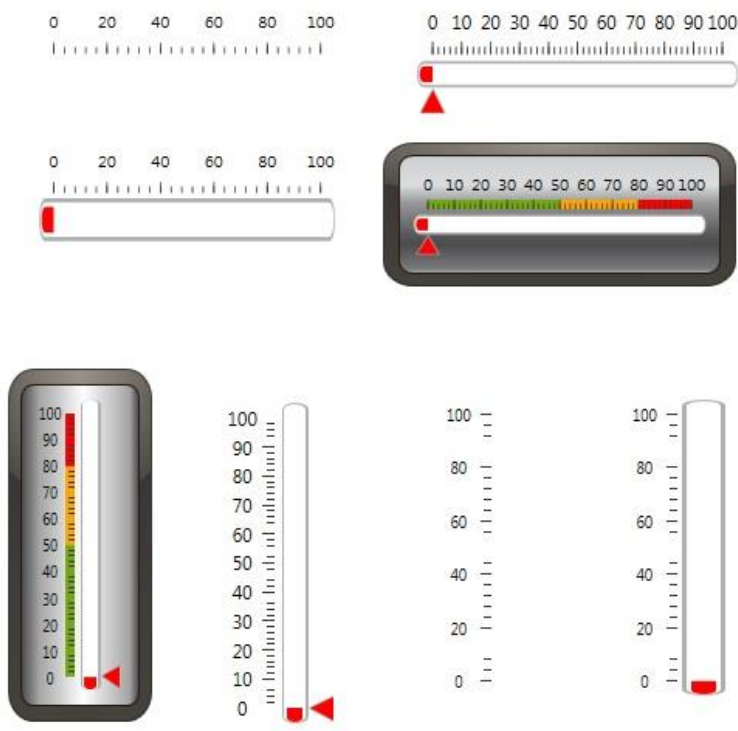
The BarGraphs have scales and bars that can be configured by means of using their properties.



### Bargraph

The Movicon NExT Meters are configurable objects that are used for displaying specified variable values graphically.

The Meters consist of a Scale, Bar and Indicator that can all be configured through their associated properties.



*A range of Movicon NExT Meters*



## Scales

The Movicon NExT scale objects are configurable and show variable values graphically. The Scale object has a scale and an indicator which can both be configured using their associated properties.

## BarGraph Properties

### Meter Scale Values

**Min. Value:** This is used to set the minimum value allowed.

**Max. Value:** This is used to set the maximum value allowed.

**Use Engineering Unit:** Permits the minimum and maximum units to be inherited from the engineering unit relating to the object's associated Tag.

### Tag Advanced

**Tag Min. Value:** This is used to set a tag to contain the min. Value allowed.

**Tag Max. Value:** This is used to set a tag to contain the max. Value allowed.

### Meter Numeric Value

**Show Value:** Shows the exact value indicated by the indicator at that exact moment within the object.

**Value Color:** Changes the colour of the value shown.

**Value Font Settings:** This is used to set the actual value's style character.

**Value Offset:** This is used to set the distance of the displayed value from the object's border.

**Format Value:** This is used to set the value format to be displayed.

### Meter Style

**Border Style:** Sets the Gauge/Bargraph's border basic style.

**Background Color:** Sets the back color for the object's general area.

**Layout Mode:** Allow you to set the object Layout direction.

### Meter Scale Style

**Auto Hide Elements:** When this property is set to True, the control's elements will gradually disappear and the number of ticks will be reduced along with the labels every time it is reduced in size. For example, as the object is gradually reduced in size, the numeric values will disappear by order they come in and the scale's ticks, numeration, indicator and borders will also be reduced.

Conversely, if this property is set to False, the object's elements will remain visible as it is made smaller to the eventually overlap each other.



This behaviour is determined by the fact that even though the object is made smaller in size, its elements, such as the scale and fonts used, are not resized to match and remain the size they were initially set with.

If you wish to resize the object in a way that all its elements are resized proportionately accordingly, you will need to create a symbol with the object's initial sizes. After you

have done this, you can then resize the symbol, which will become a Viewbox, and its elements will resize accordingly.

**Label Orientation:** Sets the orientation of the scale's values.

**Label Offset:** Sets the distance of the scale's values from the border.

**Label Color:** Set the color of the scale's values.

**Label String format:** Allow you to set the format to be used for the label's text.

**Show First Label:** Shows the scale's first value.

**Show Last Label:** Shows the scale's last value.

**Scale Line Color:** Sets the color of the Scale Line

**Major interval Count:** Set the number of major scale interval dividers.

**Minor interval Count:** Set the number of minor scale interval dividers.

### Engineering Unit Label Style

**Show Engineering Unit:** Enables/disables the "Engineering Units" labels

**Engineering Unit Color:** Used for selecting the "Engineering Units" label's text colour

**Engineering Unit Label offset:** Used for setting the offset between label text and borders.

**Engineering Unit Text:** Used to set engineering unit label text.

**Engineering Unit Font:** Used for setting the "Engineering Units" label's font type.

### Meter Tickmark Style

**Tickmark Style:** Sets the object's scale's stile.

**Major Tickmark Length:** Sets the length of the object's scale's major tickmarks.

**Major Tickmark Layout Index:** Sets the Z-index of the major tickmarks.

**Major Tickmark Thickness:** Sets the thickness of the object's scale's major tickmarks.

**Major Tickmark Offset:** Sets the distance of the scale's major tickmarks from the object's border.

**Show first Major Tickmark:** Displays/Hides the first scale's major tickmarks.

**Show last Major Tickmark:** Displays/Hides the last scale's major tickmark.

**Minor Tickmark Length:** Sets the length of the object's scale's minor tickmarks.

**Minor Tickmark Z-index:** Sets the Z-index of minor tickmarks.

**Minor Tickmark Offset:** Sets the distance between the minor tickmarks and the object's border.

**Minor Tickmark Thickness:** Sets the tickmark's thickness.

**Minor Tickmark Color:** Sets the minor tickmark's color.

**Show Minor Tickmark on Major:** allow you to show a minor tickmark instead of a major one

**Minor Tickmark Fill:** Set the color for the minor tickmark fill.

### Meter Marker Style

**Marker Stroke:** Sets the marker thickness

**Marker Visible:** Displays/Hides the marker.

**Marker Height:** Sets the marker's height.

**Marker Width:** Sets the marker's width.

**Marker Layout Index:** Sets the Marker's z-index

**Marker Offset:** Sets the distance between the marker and the border.

**Marker Color:** Sets the color of the marker.

**Marker Stroke Color:** Sets the color of the marker stroke.

**Marker Animation:** Activates or Deactivates the marker's animation.

**Marker Clickable:** Activates or Deactivates marker interactivity with mouse. If activated the user can set the value by clicking and moving it directly.

**Marker Orientation:** Allow you to set the marker orientation inside the object.

#### **Meter Value Range Bar Style**

**Range Bar Visible:** Enable/disables the Value Range Bar.

**Range Bar Layout Index:** Sets the Object's Range Bar depth index.

**Range Bar Offset:** Sets the space between the Value Range Bar segments.

**Range Bar Color:** Sets the colour of the Value Range Bar.

**Range Bar Back Color:** Sets the back color of the range bar.

**Range Bar Animation:** Enables/disables the animation for the Value Range Bar, if enabled the bar will fill until it reaches the relative position during animation.

**Range Bar Clickable:** Enables/disables the Value Range Bar interactivity.

**Range Bar Thickness:** Sets the thickness of the Value Range Bar.

#### **Meter Value Bar Style**

**Value Bar Visible:** Enable/disables the Value Bar.

**Value Bar Offset:** Sets the space between the Value Bar segments.

**Value Bar Color:** Sets the colour of the Value Bar.

**Value Bar Background Color:** Sets the back color of the Value bar.

**Value Bar Animation:** Enables/disables the animation for the Value Bar, if enabled the bar will fill until it reaches the relative position during animation.

**Value Bar Clickable:** Enables/disables the Value Bar interactivity.

**Value Bar Thickness:** Sets the thickness of the Value Bar.

#### **Meter Scale Range Style**

**Scale Zone Visible:** Makes the preselected scale zone (zones 1,2 or 3) visible/invisible.

**Scale Zone Start:** Sets the start position of the preselected scale zone (zones 1,2 or 3).

**Scale Zone End:** Sets the end position of the preselected scale zone (zones 1, 2 or 3).

**Scale Zone Thickness:** Sets the thickness of the preselected scale zone (zones 1, 2 or 3).

**Scale Zone Offset:** Sets the offset between the scale and the zone border (zones 1, 2 or 3).

**Scale Zone Color:** Sets the color of the scale zone (zones 1, 2 or 3).

**Scale Zone Layout Index:** Sets the scale zone's Z-index order value (zones 1, 2 or 3).

#### **Style**

**Orientation:** sets the object's orientation

**Border Brush:** sets the container's border colour

**Border Thickness :** sets the container's border thickness

**Foreground Color:** Sets background colour of internal numbering.

**Background color:** Sets the container's background colour.

**Font Size:** Sets size of characters inside object.

**Disable Anti-aliasing:** Enables/disables the Anti-aliasing pixel correction function.

**Margin:** sets object's margins.

## Fonts

**Font Family:** Sets font model to be used.

**Font Size:** Sets the font's size within the object.

**Font Style:** Sets the font's style.

**Font Weight:** Sets the thickness or thinness of how the font should be displayed ( *light*, *extralight*, *normal* etc...).

## Layout

See paragraph on "Common ToolBox object and drawing properties".

## Behaviour

See topic: "Drawing and Object Behaviour Properties".

## General

see the topic "Common ToolBox Object and Drawing Properties".

## Visibility

**Transparency:** Sets the object's Transparency level.

**Visibility:** enables the object's visibility.

**Visible on Client:** makes object visible on the webclient

**Visibility Level:** permits you to set the mask to determine the object's visibility.

**Zoom Level Visibility X:** permits you to set the Zoom factor on the x axis after which the object will become visible on screen.

**Zoom Level Visibility Y:** permits you to set the Zoom factor on the y axis after which the object will become visible on screen.

## Tag

See topic on: "Common ToolBox object and drawing properties".

## User Access

See topic on: "User Manager Properties" concerning objects and drawings that define the security parameters based on the project's User and Password management.

## Expressions

See topic on: "Expressions in objects" concerning objects and drawings.

## Preserve Template Changes

**Preserve:** see topic on: "Preserve local Draws and Objects properties" concerning original library objects.

#### Miscellaneous

**Object Container:** this is used for modifying the properties of the object container.

## 5.14. Buttons

There are two types of Movicon NExT command buttons. One type have ON-OFF mechanics while the other type execute COMMANDS. However, both these types are applied with the same styles leaving you to select the one most appropriate according to the function type you wish to use.

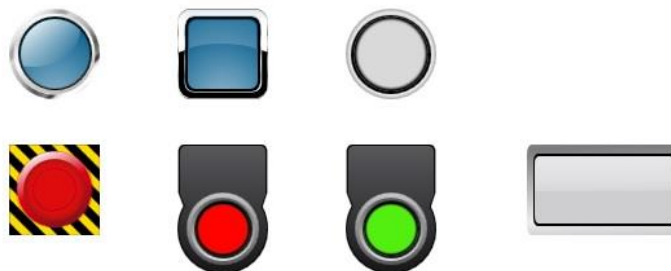


Be reminded that the graphical Styles of each inserted object can be modified as pleased but not their mechanics.

### Buttons-Commands

In this Toolbox section you will find a vast range of buttons categorized by type and style. The main characteristics of these button categories is that they can be associated to execute one or more specific system Commands according to the options provided by the platform as described in "Using command Objects"

Even though they all have the same styles as those in the Check-Box button category, they have been designed to execute commands when the project is launched into runtime, such as Page Change, running Scripts, Reports and performing operations in Tags such as Setting or resetting Tag values for example.



*Some examples of Buttons from the ToolBox*

### Checkbox/lights/Buttons Properties

#### Style

**Back Color:** This is used for setting the container's background colour.

**Stroke Color:** This is used for setting the colour of the letter and number characters used in the object.

**Border Color:** Sets the container border's color.

**Border Thickness:** Sets the container border's thickness.

**Disable Anti-aliasing:** Enables/disables the Anti-aliasing pixel correction function.

**Margin:** Sets the object's margins

## Button Style

**Text Wrapping:** this is used to return text to beginning of next line when reaching the border of the box containing it.

**Text Alignment:** This is used to set the text alignment within the button object.

**Pressed Icon:** This is used to set the image to display the button with when in pressed mode.

**Released Icon:** This is used to set the image to display the button with when in released mode.

**Checked Icon:** This is used to set the image to display the button with when in selected mode. This property is only accepted for Toggle button types ("CheckBox").

**Disabled Icon:** This is used to set the image to display the button when in disable mode (and therefore not clickable).

**Overlap Text on Image:** When this option is selected, the button's text will be displayed over lapping the image otherwise the text will be displayed separately in one half of the button with the image displayed at its side in the other half.

**Image Stretch: This is used to adapt image to button**

- *None:* image will not be adapted
- *Fill:* image will be stretched to the size of the button. In this case the image sizes will adapt to those of the button but may be disproportionate.
- *Uniform:* the image will be adapted in proportion to the entire size of the button. In this case, the image may not always cover the entire surface of the button but it will be kept proportionate.
- *UniformToFill:* the image will adapt to the size of the button keeping the image in proportion to its original size. In this case, one of the image's sizes (either height or length) will always adapt but this may cause the other to be cut off.

## Button Behaviour

**Click Mode:** This is used to set the click type to execute the command with (release, press, hover).

**Is Three State:** This is used to set buttons provided with this option to three state mode (available for Checkboxes only).

## Fonts

**Font Size:** This is used to set the font size within the object.

**Font Family:** This is used to set the font family to be used.

**Font Style:** This is used to set the font style type.

**Font Weight:** This is used to set the font type (*light, extralight, normal ecc...*).

## Layout

See topic: "Draws and Object Behaviour Properties".

### **Behaviour**

See topic: "Draws and Object Behavior Properties".

### **General**

See topic: "General Draws and Object Properties".

### **Visibility**

**Transparency:** This is used to set the object's transparency level.

**Visibility:** Enables the object's visibility.

**Visible On Web Client:** This makes object visible on Web Client.

**Visibility Layer Level:** This is used to set the mask which determines the object's visibility.

**Zoom Level Visibility X:** This is used to set the Zoom level on the X axis after which the object will become visible on screen.

**Zoom Level Visibility Y:** This is used to set the Zoom level on the Y axis after which the object will become visible on screen.

### **Tag**

See topic: "General Draws and Object Properties".

### **User Access**

See topic: "User Access Properties" of objects and draws that define the security parameters according to the project's Users and Passwords management.

### **Expressions**

See topic: "Managing Expressions in Draws and Objects"

### **Voice**

**Speech Command:** This is used to set a "speech command" in the object that will be used during runtime.

### **Preserve Template Changes**

See topic: "Preserve properties of local Draws and Object" in respect to the original object in the library.

### **Miscellaneous**

Allows you to modify the object's container properties.

## **5.15. Clocks**

The Movicon clocks can be found in the 'Clocks' section in the ToolBox under two categories: Analog and Digital. Like all the other toolbox objects, and especially the Analog version, their various components can be customised in style and color.



17:27:46

*The default digital and analog clock versions*

## Clock properties

### Clock Style

**Enable Background:** Enables/Disables the Background.

**Uses System Time Zone:** When this is enabled, the PC's time zone will be used.

**Clock Time Zone:** This is used for setting a custom time zone.

**Show Seconds:** When enabled the seconds will show on the clock

**Enable Background:** Enables/Disables the Background.

**Clock Text Visible:** Allows the time to be shown in text format.

**Clock Text Color:** Sets the color of the time text format.

**Text Color Inactive:** Sets the inactive text color.

**Clock Text Background Color:** Sets the background color of the time text format.

**Clock Font Settings:** Used for selecting the font type to be used by the time in text format.

**Clock Style:** Sets the clock's basic style.

**Clock Background Color:** Sets the clock's background color.

**Hour Label Visible:** Used for showing hour labels.

**Label Orientation:** Sets the label orientation type.

**Label Text Color:** Sets the back color of the labels.

**Label Font Settings:** Sets font type to use in labels.

### Clock TickmarkStyle

**Tickmark Style:** Sets the object's scale style.

**Major Tickmark Length:** Sets the length of the major tickmarks in the object's scale.

**Major Tickmark Z-Index:** Sets the depth level of the scale's major tickmarks.

**Major Tickmark Thickness:** Sets the thickness of the objects's scale's major tickmarks

**Major Tickmark Offset:** Sets the offset of the object's scale's major tickmarks.

**Show first Major Tickmark:** Shows/Hides the scale's first major tickmark.

**Show last Major Tickmark:** Shows/Hides the scale's last major tickmark.



**Minor Tickmark Length:** Sets the object's scale's minor tickmarks.

**Minor Tickmark Z-Index:** Sets the depth level of the scale's minor tickmarks.

**Minor Tickmark Offset:** Sets the offset of the object's scale's minor tickmarks.

**Minor Tickmark Thickness:** Sets the thickness of the object's scale's minor tickmarks.

**Show Minor Tick. On Major:** Shows/Hides the minor tickmarks corresponding the major one, normally superimposed.

### **ClockSpindleStyle**

**Spindle cap Style:** Sets the style of the spindle in the centre of the object.

**Spindle Height:** Sets the height of the spindle in the centre.

**Spindle Width:** Sets the width of the spindle in the centre.

**Spindle Cap Z-Index:** This is used to set the spindle z-index.

### **ClockNeedleStyle**

**Needle Style:** Sets the Needle style

**Needle Color:** Sets the colour of the clock hands.

**Needle Visible:** Hide/Show clock hand.

**Needle Animation Enable:** Activate/Deactivate clock hand animation.

**Needle Z-Index:** Sets the object's needle depth level.

### **Style**

**Border Brush:** This is used for setting the container's border colour.

**Border Thickness:** This is used for setting the container's border thickness.

**Foreground Color:** This is used for setting the colour of the letter and number characters used in the object.

**Background color:** This is used for setting the container's background colour.

**Font Size:** This is used for setting the font size used within the object.

**Disable Anti-aliasing:** Enables/disables the Anti-aliasing pixel correction function.

**Margin:** Sets the object's margins.

### **Fonts**

**Font Family:** Sets font model to be used.

**Font Size:** Sets the font's size within the object.

**Font Style:** Sets the font's style.

**Font Weight:** Sets the thickness or thinness of how the font should be displayed ( *light*, *extralight*, *normal* etc...).

### **Layout**

See paragraph on "Common ToolBox object and drawing properties".

### **Behavior**

See topic: "Drawing and Object Behaviour Properties".

## General

see the topic "Common ToolBox Object and Drawing Properties".

## Visibility

**Transparency:** Sets the object's Transparency level.

**Visibility:** enables the object's visibility.

**Visible on Client:** makes object visible on the webclient

**Visibility Level:** permits you to set the mask to determine the object's visibility.

**Zoom Level Visibility X:** permits you to set the Zoom factor on the x axis after which the object will become visible on screen.

**Zoom Level Visibility Y:** permits you to set the Zoom factor on the y axis after which the object will become visible on screen. n.

## Tag

See paragraph on "Common ToolBox object and drawing properties".

## User Management

See topic on "Draws and Object User Management" concerning objects and drawings that define the security parameters based on the project's User and Password management.

## Expressions

See paragraph "Managing Expressions" concerning objects and drawings.

## Preserve Template Changes

**Preserve Style:** see paragraph "Preserve local Draws and Objects properties" concerning original library objects

## 5.16. DB Connectors

## 5.17. Combo Box



This object comes from the "Dialog Box" category and can be configured by inserting a list of strings separated by the "|" character, or by setting a query to retrieve data from a DB table. Data is not entered but selected from the drop down menu which opens when clicking the arrow.

## Combo Box Properties

### ComboBox Style

**ComboBox Style:** Changes the object's style.

**Toggle-Button Size:** Sets the size of the button.

**Tag Item Source:** This is used to select the variable from which the displayed data source will be to obtained.

### Layout

See paragraph on "Common ToolBox object and drawing properties".

### Behaviour

See topic: "Drawing and Object Behaviour Properties".

### Style

**Border Brush:** This is used for setting the container's border colour.

**Border Thickness:** This is used for setting the container's border thickness.

**Back color:** This is used for setting the container's background colour.

**Stroke Color:** This is used for setting the colour of the letter and number characters used in the object.

**Disable Anti-aliasing:** Enables/disables the Anti-aliasing pixel correction function.

**Margin:** This is used to set the object's margins.

### Fonts

**Font Family:** Sets font model to be used.

**Font Size:** Sets the font's size within the object.

**Font Style:** Sets the font's style.

**Font Weight:** Sets the thickness or thinness of how the font should be displayed ( *light*, *extralight*, *normal* etc...).

### General

see topic "Common ToolBox Object and Drawing Properties".

### Visibility

**Transparency:** Sets the object's Transparency level.

**Visibility:** enables the object's visibility.

**Visible On Web Client:** makes object visible on the webclient

**Visibility Layer Level:** permits you to set the mask to determine the object's visibility.

**Zoom Level Visibility X:** permits you to set the Zoom factor on the x axis after which the object will become visible on screen.

**Zoom Level Visibility Y:** permits you to set the Zoom factor on the y axis after which the object will become visible on screen.

### Tag

See paragraph on "Common ToolBox object and drawing properties".

### User Access

See topic on "Draws and Object User Management" concerning objects and drawings that define the security parameters based on the project's User and Password management.

### Expressions

See paragraph "Managing Expression" concerning objects and drawings.

### Preserve Template Changes

**Preserve:** see paragraph "Preserve local Draws and Objects properties" concerning original library objects.

### Miscellaneous

Permits you to modify the object container properties.

## 5.18. DB List Box



The Movicon NExT List Box objects are components that can be inserted anywhere on screen in order to display lists of items. A numeric or string type variable can be associated to this object. Each row of the list corresponds to a numeric value that is made available in the variable associated to the object if this variable is numeric type. If the variable is string type, it will be updated with the selected Item's text. Each time a row is selected, the List Box's variable is updated with the corresponding numeric value (the value is progressive, starting from 0 to indicate the first row) or with the selected Item's text.

### List Box Properties

#### List Box Style

**Tag Item source:** This is used to select the tag name where to get the Source Data to Display in the Object list.

**Selection Mode:** This is used to set the selection mode type of listed items.

**ListBox Style:** This is used to set the List Box's style type.

#### Behaviour

See topic: "Drawing and Object Behaviour Properties".

#### Style

**Border Brush:** This is used to set the container's border colour.

**Border Thickness:** This is used to set the container's border thickness.

**Foreground Color:** This is used to set the colour of the letter and number characters used in the object.

**Background color:** This is used to set the container's background colour.

**Disable Anti-aliasing:** Enables/disables the Anti-aliasing pixel correction function.

**Margin:** This is used to set the object's margins

## Fonts

**Font Family:** This is used to set the font model to be used.

**Font Size:** This is used to set the font's size within the object.

**Font Style:** This is used to set the font's style.

**Font Weight:** This is used to set the thickness or thinness of how the font should be displayed ( *light, extralight, normal etc...*).

## Layout

See topic on: "Common ToolBox object and drawing properties".

## General

See topic on: "Common ToolBox Object and Drawing Properties".

## Visibility

**Transparency:** This is used to Set the object's Transparency level.

**Visibility:** This is used to enable the object's visibility.

**Visible on Client:** This is used to make the object visible on the webclient

**Visibility Level:** This is used to set the mask to determine the object's visibility.

**Zoom Level Visibility X:** This is used to set the Zoom factor on the x axis after which the object will become visible on screen.

**Zoom Level Visibility Y:** This is used to set the Zoom factor on the y axis after which the object will become visible on screen.

## Tag

See topic on: "Common ToolBox object and drawing properties".

## User Access

See topic on: "Draws and Object User Management" regarding objects and draws that define the security parameters based on the project's User and Password management.

## Expressions

See topic on: "Managing Expression" regarding objects and draws.

## Preserve Template Changes

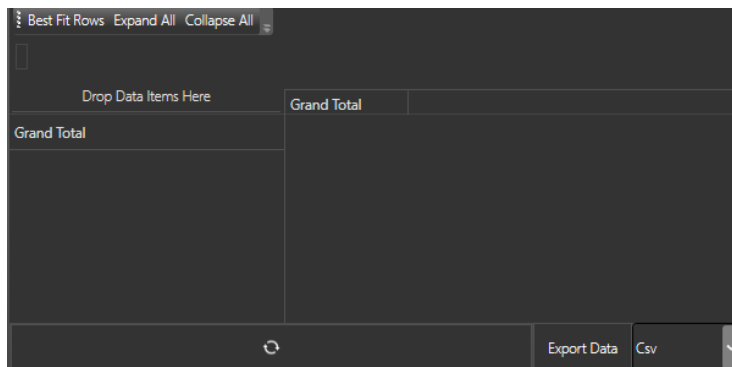
**Preserve:** see topic on: "Preserve local Draws and Objects properties" regarding the original library objects.



When the object is published on the Web Client, a window displays in the middle of the page to select values from those available on the list. In order for this window to function, a variable must be associated to the object. This variable can be string or integer type and will function as the chosen selection index or to show the text of the selected string.

## 5.19. DB Pivot Grid

The Pivot Grid object is used for creating a pivot table to display multi-dimensional data analysis. The Pivot Grid allows large amounts of data to be summarized and represented in a table where it can then be sorted, grouped and filtered. Users can personalize the table layout according to the type of analysis they wish to perform. This is simply done by dragging & dropping the specific items within the table as desired.



### Pivot Grid Properties

#### Pivot Grid Style

**Toolbar Background:** This is used to define the background color to apply to the Toolbar.

**Toolbar Foreground:** This is used to define the foreground color to apply to the Toolbar.

**Show Column Headers:** Enables the column headers to show.

**Show Column Totals:** Enables the Column totals to show.

**Show Column Headers** Enables column headers to show.

**Show Filter Headers:** Enables filter headers to show.

**Row Area Font:** This is used to set the font for the row area.

**Header Font:** This is used to set the alarm column header font.

**Show Row Grand Totals:** Enables the showing of the grand total row.

**Show Row Totals:** Enables the showing of the Total Row.

**Show Row Headers:** Enables the showing of the row headers.

**Show Column Grand Totals:** Enables the grand totals of the columns to show.

**Show Column Grand Total Header:** Enables the grand total column header to show.

**Show Totals for Single Values:** Enables the totals of individual values to show.

**Show Prefilter Panel:** Enables the prefilter panel to show.

**Use Icon:** Enables the use of icons in the toolbar.

**Header Background:** This is used to set the Grid's Header background color.

**Header Foreground:** This is used to set the Grid's Header text color.

**Row Area Foreground:** This is used to set the Grid's row text color.

**Focused Row Background:** This is used to set the background color of the selected row.

**Focused Row Foreground:** This is used to set the color of the selected row's text.

**Show Cells Border:** This is used to show/hide the border of Grid's cells.

**Cell Border Color:** This is used to set the color of the Grid's cells' borders.

**Focused Cell Background:** This is used to set the background color of the selected cell /cells.

**Focused Cell Foreground:** This is used to set the text color of the selected cell/cells.

## General

See topic on "Proprietà comuni oggetti e disegni ToolBox".

## Griglia Pivot Advanced

**Editable:** Enables the window to be edited in runtime mode.

## User Access

See topic on "Proprietà di Gestione Utenti" in objects and draws that define the security and protection parameters according othe project's Users and Passwords management.

## Style

**Border Brush:** This is used to set the container's border colour.

**Border Thickness:** This is used for setting the container's border thickness.

**Back color:** This is used to set the container's background colour.

**Stroke Color:** This is used to set the colour of the letter and number characters used in the object.

**Disable Anti-aliasing:** Enables/disables the Anti-aliasing pixel correction function.

**Margin:** This is used to set the object's margins

## Fonts

**Font Family:** This is used to Set font model to be used.

**Font Size:** This is used to set the font's size within the object.

**Font Style:** This is used to set the font's style.

**Font Weight:** This is used to sets the displayed Font's thickness or thinness ( *light*, *extralight*, *normal* etc...).

## Layout

See topic on: "Common ToolBox object and drawing properties".

## Behaviour

See topic on: "Drawing and Object Behaviour Properties".

## Visibility

**Transparency:** This is used to set the object's Transparency level.

**Visibility:** Enables the object's visibility.

**Visible on Web Client:** Enable this to make object visible on the webclient

**Visibility Layer Level:** This is used to set the mask to determine the object's visibility.

**Zoom Level Visibility X:** This is used to set the Zoom factor on the x axis after which the object will become visible on screen.

**Zoom Level Visibility Y:** This is used to set the Zoom factor on the y axis after which the object will become visible on screen.

#### **Tag**

See topic on: "Common ToolBox object and drawing properties".

#### **Expressions**

See topic on: "Managing Expression" concerning objects and draws.

#### **Preserve Template Changes**

See topic on: "Preserve local Draws and Objects properties" regarding original library objects

## 5.20. Dialog Box

## 5.21. Switch Buttons, Lights and Checkbox

The Movicon NExT buttons are divided into two types according to their mechanics but they all have the same style properties. Some buttons have ON-OFF mechanics while others execute COMMANDS.

Therefore you must insert the button type that is most appropriate for the functionality needed.



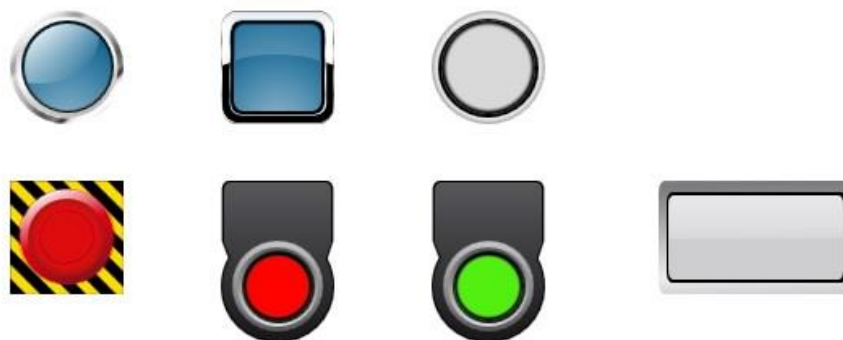
Be reminded that the styles of each object you insert on screen can be modified graphically but their mechanics cannot.

### Buttons-DigitalCheckbox

In Toolbox section you will find a vast range of buttons by type and style. The main characteristic of this category is the possibility to associate a variable for toggling between 0 and 1 (On-Off command).

Even though this object's styles are the same as those of the command buttons, its "Check-Box" mechanics require you to associate it a Tag variable. This can be done by using the Drag & Drop techniques or by associating the Tag using the properties or quick select menu.



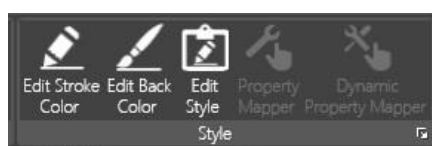


*Some examples of Buttons from the Toolbox*

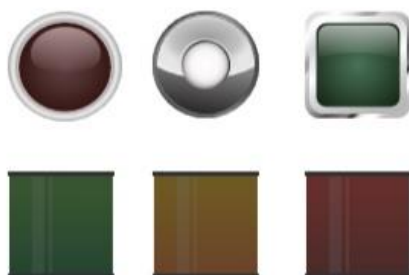
## Lamps

The Lamp or light objects are luminous indicators that are used for representing the status of the connected variable. This is established by setting the object's behaviour properties to define its mechanics based on two or three states.

The colour of the luminous led colours can be configured using the "edit back color" button from the "Style" ribbon.



These objects are read only for default but the assigned variables can also be controlled using the associated properties.

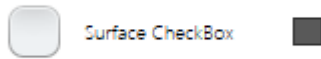


*Some examples of lights and lamps from the Toolbox*

## CheckBox

The checkBox is a classical two state selector button. A Movicon NExT variable can be associated to this control and set to the "0" value (unchecked box) or the "1" value (checked box) according to the state of the box.

The state can be selected with a mouse click or using the keyboard to select the by selecting the component with the TAB or MAIUSC+TAB keys and pressing ENTER.



## Checkbox/lights/Buttons Properties

### Style

**Back Color:** This is used for setting the container's background colour.

**Stroke Color:** This is used for setting the colour of the letter and number characters used in the object.

**Border Color:** Sets the container border's color.

**Border Thickness:** Sets the container border's thickness.

**Disable Anti-aliasing:** Enables/disables the Anti-aliasing pixel correction function.

**Margin:** Sets the object's margins

### Button Style

**Text Wrapping:** this is used to return text to beginning of next line when reaching the border of the box containing it.

**Text Alignment:** This is used to set the text alignment within the button object.

**Pressed Icon:** This is used to set the image to display the button with when in pressed mode.

**Released Icon:** This is used to set the image to display the button with when in released mode.

**Checked Icon:** This is used to set the image to display the button with when in selected mode. This property is only accepted for Toggle button types ("CheckBox").

**Disabled Icon:** This is used to set the image to display the button when in disable mode (and therefore not clickable).

**Overlap Text on Image:** When this option is selected, the button's text will be displayed over lapping the image otherwise the text will be displayed separately in one half of the button with the image displayed at its side in the other half.

**Image Stretch: This is used to adapt image to button**

- *None:* image will not be adapted
- *Fill:* image will be stretched to the size of the button. In this case the image sizes will adapt to those of the button but may be disproportionate.
- *Uniform:* the image will be adapted in proportion to the entire size of the button. In this case, the image may not always cover the entire surface of the button but it will be kept proportionate.
- *UniformToFill:* the image will adapt to the size of the button keeping the image in proportion to its original size. In this case, one of the image's sizes (either height or length) will always adapt but this may cause the other to be cut off.

### Button Behaviour

**Click Mode:** This is used to set the click type to execute the command with (release, press, hover).

**Is Three State:** This is used to set buttons provided with this option to three state mode (available for Checkboxes only).

## Fonts

**Font Size:** This is used to set the font size within the object.

**Font Family:** This is used to set the font family to be used.

**Font Style:** This is used to set the font style type.

**Font Weight:** This is used to set the font type (*light, extralight, normal ecc...*).

## Layout

See topic: "Draws and Object Behaviour Properties".

## Behaviour

See topic: "Draws and Object Behavior Properties".

## General

See topic: "General Draws and Object Properties".

## Visibility

**Transparency:** This is used to set the object's transparency level.

**Visibility:** Enables the object's visibility.

**Visible On Web Client:** This makes object visible on Web Client.

**Visibility Layer Level:** This is used to set the mask which determines the object's visibility.

**Zoom Level Visibility X:** This is used to set the Zoom level on the X axis after which the object will become visible on screen.

**Zoom Level Visibility Y:** This is used to set the Zoom level on the Y axis after which the object will become visible on screen.

## Tag

See topic: "General Draws and Object Properties".

## User Access

See topic: "User Access Properties" of objects and draws that define the security parameters according to the project's Users and Passwords management.

## Expressions

See topic: "Managing Expressions in Draws and Objects"

## Voice

**Speech Command:** This is used to set a "speech command" in the object that will be used during runtime.

## Preserve Template Changes

See topic: "Preserve properties of local Draws and Object" in respect to the original object in the library.

### Miscellaneous

Allows you to modify the object's container properties.

## 5.22. Combo Box



This object comes from the "Dialog Box" category and can be configured by inserting a list of strings separated by the "|" character, or by setting a query to retrieve data from a DB table. Data is not entered but selected from the drop down menu which opens when clicking the arrow.

### Combo Box Properties

#### ComboBox Style

**ComboBox Style:** Changes the object's style.

**Toggle-Button Size:** Sets the size of the button.

**Tag Item Source:** This is used to select the variable from which the displayed data source will be to obtained.

#### Layout

See paragraph on "Common ToolBox object and drawing properties".

#### Behaviour

See topic: "Drawing and Object Behaviour Properties".

#### Style

**Border Brush:** This is used for setting the container's border colour.

**Border Thickness:** This is used for setting the container's border thickness.

**Back color:** This is used for setting the container's background colour.

**Stroke Color:** This is used for setting the colour of the letter and number characters used in the object.

**Disable Anti-aliasing:** Enables/disables the Anti-aliasing pixel correction function.

**Margin:** This is used to set the object's margins.

#### Fonts

**Font Family:** Sets font model to be used.

**Font Size:** Sets the font's size within the object.

**Font Style:** Sets the font's style.

**Font Weight:** Sets the thickness or thinness of how the font should be displayed ( *light*, *extralight*, *normal* etc...).

### General

see topic "Common ToolBox Object and Drawing Properties".

### Visibility

**Transparency:** Sets the object's Transparency level.

**Visibility:** enables the object's visibility.

**Visible On Web Client:** makes object visible on the webclient

**Visibility Layer Level:** permits you to set the mask to determine the object's visibility.

**Zoom Level Visibility X:** permits you to set the Zoom factor on the x axis after which the object will become visible on screen.

**Zoom Level Visibility Y:** permits you to set the Zoom factor on the y axis after which the object will become visible on screen.

### Tag

See paragraph on "Common ToolBox object and drawing properties".

### User Access

See topic on "Draws and Object User Management" concerning objects and drawings that define the security parameters based on the project's User and Password management.

### Expressions

See paragraph "Managing Expression" concerning objects and drawings.

### Preserve Template Changes

**Preserve:** see paragraph "Preserve local Draws and Objects properties" concerning original library objects.

### Miscellaneous

Permits you to modify the object container properties.

## 5.23. Group Box

This object is used to frame groups of object items of the same kind. The object Items can be collocated within the frame and the heading (Groupbox) can be changed accordingly.

### Group Box Properties

#### Style

**Border Brush:** This is used to set the container's border colour.

**Border Thickness:** This is used to set the container's border thickness.

**Back color:** This is used to set the container's background colour.

**Stroke Color:** This is used to set the colour of the letter and number characters used in the object.

**Disable Anti-aliasing:** Enables/disables the Anti-aliasing pixel correction function.

**Margin:** This is used to set the object's margins.

## Fonts

**Font Family:** Sets font model to be used.

**Font Size:** Sets the font's size within the object.

**Font Style:** Sets the font's style.

**Font Weight:** Sets the thickness or thinness of how the font should be displayed ( *light*, *extralight*, *normal* etc...).

## Layout

See paragraph on "Common ToolBox object and drawing properties".

## Behaviour

See paragraph on "Drawing and Object Behaviour Properties".

## General

see topic "Common ToolBox Object and Drawing Properties".

## Visibility

**Transparency:** Sets the object's Transparency level.

**Visibility:** enables the object's visibility.

**Visible On Web Client:** makes object visible on the webclient

**Visibility Layer Level:** Used to set the mask to determine the object's visibility.

**Zoom Level Visibility X:** Used to set the Zoom factor on the x axis after which the object will become visible on screen.

**Zoom Level Visibility Y:** Used to set the Zoom factor on the y axis after which the object will become visible on screen.

## Tag

See topic on: "Common ToolBox object and drawing properties".

## User Access

See topic on: "Draws and Object User Management" concerning objects and drawings that define the security parameters based on the project's User and Password management.

## Expressions

See topic on: "Managing Expression" concerning objects and drawings.

## Preserve Template Changes

**Preserve:** see topic on: "Preserve local Draws and Objects properties" concerning original library objects.

## Miscellaneous

Permits you to modify the object container properties.

## 5.24. DB List Box



The Movicon NExT List Box objects are components that can be inserted anywhere on screen in order to display lists of items. A numeric or string type variable can be associated to this object. Each row of the list corresponds to a numeric value that is made available in the variable associated to the object if this variable is numeric type. If the variable is string type, it will be updated with the selected Item's text. Each time a row is selected, the List Box's variable is updated with the corresponding numeric value (the value is progressive, starting from 0 to indicate the first row) or with the selected Item's text.

### List Box Properties

#### List Box Style

**Tag Item source:** This is used to select the tag name where to get the Source Data to Display in the Object list.

**Selection Mode:** This is used to set the selection mode type of listed items.

**ListBox Style:** This is used to set the List Box's style type.

#### Behaviour

See topic: "Drawing and Object Behaviour Properties".

#### Style

**Border Brush:** This is used to set the container's border colour.

**Border Thickness:** This is used to set the container's border thickness.

**Foreground Color:** This is used to set the colour of the letter and number characters used in the object.

**Background color:** This is used to set the container's background colour.

**Disable Anti-aliasing:** Enables/disables the Anti-aliasing pixel correction function.

**Margin:** This is used to set the object's margins

#### Fonts

**Font Family:** This is used to set the font model to be used.

**Font Size:** This is used to set the font's size within the object.

**Font Style:** This is used to set the font's style.

**Font Weight:** This is used to set the thickness or thinness of how the font should be displayed ( *light, extralight, normal etc...*).

### Layout

See topic on: "Common ToolBox object and drawing properties".

### General

See topic on: "Common ToolBox Object and Drawing Properties".

### Visibility

**Transparency:** This is used to Set the object's Transparency level.

**Visibility:** This is used to enable the object's visibility.

**Visible on Client:** This is used to make the object visible on the webclient

**Visibility Level:** This is used to set the mask to determine the object's visibility.

**Zoom Level Visibility X:** This is used to set the Zoom factor on the x axis after which the object will become visible on screen.

**Zoom Level Visibility Y:** This is used to set the Zoom factor on the y axis after which the object will become visible on screen.

### Tag

See topic on: "Common ToolBox object and drawing properties".

### User Access

See topic on: "Draws and Object User Management" regarding objects and draws that define the security parameters based on the project's User and Password management.

### Expressions

See topic on: "Managing Expression" regarding objects and draws.

### Preserve Template Changes

**Preserve:** see topic on: "Preserve local Draws and Objects properties" regarding the original library objects.



When the object is published on the Web Client, a window displays in the middle of the page to select values from those available on the list. In order for this window to function, a variable must be associated to the object. This variable can be string or integer type and will function as the chosen selection index or to show the text of the selected string.

## 5.25. Option Button

This is a typical Option Button to which a Movicon NExT variable can be associated to obtain a progressive numeric value based on the box selected. For example, the variable will obtain the "0" value when the first box is selected and the "1" when the second box is select and so forth. The number of options to display is set using the object's properties.



The options are selected with a mouse click or by using the keyboard.

## Option Button properties

### Option Button Style

**Option \_Button Style:** This is used to set the option button Style selected.

**Bullet Size:** This is used to set the bullet option size.

**Option Item Size:** This is used to set the option item list size.

### Behaviour

**Keep Elements synchronized:** keeps the selected elements synchronized with the relating source.

**Selected Index:** This is used to set the index value for the first element on the list.

See topic on: "Drawing and Object Behaviour Properties".

### Style

**Is Three State:** This is used to set the button with three states if available as an option for this object (available only for Checkbox objects).

**Click mode:** This is used to select which click mode to generate command execution (release, press, hover).

**Border Brush:** This is used to set the container's border colour.

**Border Thickness:** This is used to set the container's border thickness.

**Foreground Color:** This is used to set the colour of the letter and number characters used in the object.

**Background color:** This is used to set the container's background colour.

**Disable Anti-aliasing:** Enables/disables the Anti-aliasing pixel correction function.

**Margin:** This is used to set the object's margins

**Transparency:** This is used to set the object's Transparency level.

### Fonts

**Font Family:** This is used to set the font model to be used.

**Font Size:** This is used to set the font's size within the object.

**Font Style:** This is used to set the font's style.

**Font Weight:** This is used to set the thickness or thinness of how the font should be displayed ( *light*, *extralight*, *normal* etc...).

### Layout

See topic on: "Common ToolBox object and drawing properties".

### General

See topic on: "Common ToolBox Object and Drawing Properties".

### Visibility

**Transparency:** This is used to set the object's Transparency level.

**Visibility:** Enables the object's visibility.

**Visible on Client:** This makes object visible on the webclient

**Visibility Level:** This is used to set the mask to determine the object's visibility.

**Zoom Level Visibility X:** This is used to set the Zoom factor on the x axis after which the object will become visible on screen.

**Zoom Level Visibility Y:** This is used to set the Zoom factor on the y axis after which the object will become visible on screen.

### Tag

See topic on: "Common ToolBox object and drawing properties".

### User Access

See topic on: "Draws and Object User Management" concerning objects and drawings that define the security parameters based on the project's User and Password management.

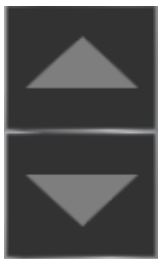
### Expressions

See topic on: "Managing Expression" concerning objects and drawings.

### Preserve Template Changes

**Preserve:** see topic on "Preserve local Draws and Objects properties" concerning original library objects.

## 5.26. Spin Button



This button is used for increasing/decreasing the tag associated to it by one value, specified by the user, using the classic arrow spin function.

This object is usually displayed in other display objects by means of enabling the Enable Spin option.

## Animated Text Properties

### Spin Control

**Use Engineering Unit:** Enabling this will allow the engineering unit values to be used for the object's Min. and Max values.

**Spin Step:** This is used to set the spin step to increase/decrease value.

## Style

**Min. Value:** This is used to set the minimum value consented

**Max Value:** This is used to set the maximum value consented

**Border Brush:** This is used to set the container's border colour.

**Border Thickness:** This is used to set the container's border thickness.

**Stroke Color:** This is used to set the border color.

**Background color:** This is used to set the container's background colour.

**Disable Anti-aliasing:** Enables/disables the Anti-aliasing pixel correction function.

**Margin:** This is used to set the object's margins.

## Tag Advanced

**Tag Min Value:** Allows the object's min. value to be used dynamically by means of the Tag.

**Tag Max Value:** Allows the object's Max. value to be used dynamically by means of the Tag.

## Fonts

**Font Family:** Sets font model to be used.

**Font Size:** Sets the font's size within the object.

**Font Style:** Sets the font's style.

**Font Weight:** Sets the thickness or thinness of how the font should be displayed ( *light, extralight, normal etc...*).

## Layout

See topic on: "Common ToolBox object and drawing properties".

## Behaviour

See topic on: "Drawing and Object Behaviour Properties".

## General

See topic on: "Common ToolBox Object and Drawing Properties".

## Visibility

**Transparency:** Sets the object's Transparency level.

**Visibility:** Enables the object's visibility.

**Visible on Client:** This makes the object visible on the webclient

**Visibility Level:** This is used to set the mask to determine the object's visibility.

**Zoom Level Visibility X:** This is used to set the Zoom factor on the x axis after which the object will become visible on screen.

**Zoom Level Visibility Y:** This is used to set the Zoom factor on the y axis after which the object will become visible on screen.

## Tag

See topic on: "Common ToolBox object and drawing properties".

### User Access

See topic on: "Draws and Object User Management" concerning objects and drawings that define the security parameters based on the project's User and Password management.

### Expressions

See topic on: "Managing Expression" concerning objects and drawings.

### Preserve Template Changes

**Preserve:** See topic on "Preserve local Draws and Objects properties" concerning original library objects.

## 5.27. Displays

The Movicon NExT Displays are components which can be inserted anywhere on screen for dynamically displaying variables.

The display object is used for showing figures or strings whose value is the content of the associated variable that can be set and configured using the 'properties window'.

The Display objects are available from the ToolBox "Displays" Group.

The Display object also allows the operator to modify the value of the associated variable. This is done by selecting the Display and entering the value desired or using the spin arrow command at the side to increase or decrease the value being displayed. The Display object can also be set to read only using the appropriate property.



*Movicon NExT Displays*



The CalendarEditDisplay object is used to select a date but not the time. If a format is defined in the "Display Format String" property that includes the time as well, this format will be applied so that when selecting a date, the hour will also be added and declared as 00.00:00.

The Display shows the value of the variable assigned during project runtime time with the style and size set using its associated properties.

## Display Properties

### Date Edit Style

**Display Format String:** This is used to set the text string to display. For further details please refer **Standard format strings** and **Customized format strings**

**Convert Format String:** String for Conversion Format (date/time).

**Read Only:** This is used to set the object as read only.

**Show border:** This is used to show the border of the object

**Null Text:** This is used to set a text to display when the value to be displayed is null (date/time).

### Calendar Edit Style

**Null Text:** This is used to set a text to display when the value to be displayed is null (date/time).

**Read Only:** This is used to set the object as read only.

**Display Format String:** This is used to set the format of the text string to display (date/time).

### Display Value

**Error Icon Color:** This is used to set the color of the error icon when displayed (numeric).

**Out Of Range:** Tick this to enable the 'values out of range' management (numeric).

**Use Engineering Unit:** Tick this to enable the minimum and maximum limits to be inherited from the engineering unit relating to the Tag associated to the object (numeric).

**Minimum Value:** Use to set the minimum value allowed (numeric).

**Maximum Value:** Use to set the maximum value allowed (numeric).

**Tag min. Value:** This allows you to set a Tag to contain the allowed minimum value (numeric).

**Tag max. Value:** This allows you to set a Tag to contain the allowed maximum value (numeric).

**Password Style Digits:** Enable this to hide the value by using the Password style (numeric).

**Precision Digits:** Number of decimal figures to be displayed (numeric).

### Display Text Style

**Text Color:** Use this to set the color of the display's text.

**Unit Label:** Use this for setting a unit label in the Display. If not managed and if it is defined an "Engineering unit", associated to the variable, the unit defined in "Engineering unit will be retrieved."

**Value Font Settings:** This is used to set the font to be used in the display.

**Unit Label Font Settings:** This is used to set the font used for the unit label.

**Text Horizontal Align:** Use this to align text horizontally.

**Text Vertical Align:** Use this to align text vertically.

**Unit Offset:** Read only value.

**Text Offset:** Use this to set the Spin value which is used to change the displayed value.

**Text Wrapping:** Use this to activate/deactivate the text wrap of values within the display.

### Display Style

**Display Border Color:** Use this to set the color of the display's border.

**Display Border Thickness:** Use this to set the display's border thickness.

**Read-Only:** Sets the editor as read only to stop the user from changing the value.

### Spin Display

**Spin Color :** This is used to change the spin's colour.

**Spin Position:** Moves the position of the spin within the display.

**Spin Size:** This is used to change the side of the spin

**Spin Enabled:** Activates/Deactivates spin function.

**Spin Step:** This is used to set the increase/decrease step value for the spin button.

### Tag Advanced

**Tag Min. Value:** Use this to set a tag to contain the min. Value allowed.

**Tag Max. Value:** Use this to set a tag to contain the max. Value allowed.

### Style

**Border Brush:** This is used for setting the container's border colour.

**Border Thickness:** This is used for setting the container's border thickness.

**Foreground Color:** This is used for setting the colour of the letter and number characters used in the object.

**Background color:** This is used for setting the container's background colour.

**Font Size:** This is used for setting the font size used within the object.

**Disable Anti-aliasing:** Enables/disables the Anti-aliasing pixel correction function.

**Margin:** Use this to set the object's margins.

### Fonts

**Font Family:** Use this to set the font model to be used.

**Font Size:** Use this to set the font's size within the object.

**Font Style:** Use this to set the font's style.

**Font Weight:** Use this to set the thickness or thinness of how the font should be displayed ( *light*, *extralight*, *normal* etc...).

### Layout

See paragraph on "Common ToolBox object and drawing properties".

### Behaviour

See topic: "Drawing and Object Behaviour Properties".

### General

see the topic "Common ToolBox Object and Drawing Properties".

**Show Digit Grouping Symbol:** This property allows the thousands separator to show in displays.

## Visibility

**Visibility:** enables the object's visibility.

**Visible on Client:** makes object visible on the webclient

**Visibility Level:** permits you to set the mask to determine the object's visibility.

**Zoom Level Visibility X:** permits you to set the Zoom factor on the x axis after which the object will become visible on screen.

**Zoom Level Visibility Y:** permits you to set the Zoom factor on the y axis after which the object will become visible on screen.

## Tag

**Tag :** Used to display and assign project variable to the selected object.

**Time Span Format:** This is used to specify a custom format for the value to be displayed for the 'Total Time ON' statistics.

**Stat Def:** When inserting a variable that has been enabled with statistics in the Tag field, you will be able to select to which Tag value to display in the display as follows:

- None: current variable value (default)
- Min: minimum value obtained by the variable
- Max: maximum value obtained by the variable
- Average: calculated average variable value
- NumUpdate: number of variable variations taken place
- TotalTimeOn: how long the variable maintained a value that was not zero

See topic: "Common ToolBox object and drawing properties".

## User Management

See topic: "Draws and Object User Management" concerning objects and drawings that define the security parameters based on the project's User and Password management.

## Expressions

See topic: "Managing Expression" concerning objects and drawings.

## Preserve Template Changes

**Preserve:** see topic on: "Preserve local Draws and Objects properties" concerning original library objects.

## Strings

### Standard format strings for Date/Time values

The standard Date/Time format strings contain one character. This character defines the model used to represent the value (and in which mode to display the year, month, day and time numbers).

Format Specifier	Description	Sample Output
d	Short date pattern	3/12/2003
D	Long date pattern	Wednesday, March 12, 2003

t	Short time pattern	12:00 AM
T	Long time pattern	12:00:00 AM
f	Full date/time pattern	Wednesday, March 12, 2003 12:00 AM
F	Full date/time pattern	Wednesday, March 12, 2003 12:00:00 AM
g	General date/time pattern	3/12/2003 12:00 AM
G	General date/time pattern	3/12/2003 12:00:00 AM

### Customized format strings for Date/Time values

To create models for the Date/Time values, you need to combine the strings listed in the table below. These strings represent the year, month, day and time in different formats.

Symbol	Meaning	Result of formatting a sample value (9/2/2003)
yy	The last two digits of the year.	03
yyyy	A four digit year	2003
MM	The number of the month	09
MMM	A short text description of the month	Sep
MMMM	The full name of the month	September
dd	The number of the day	02
ddd	A short text for the day of the week	Tue
dddd	The full name of the day of the week	Tuesday
/	Date separator. Its appearance depends on regional settings.	

Symbol	Meaning
hh	Hours
mm	Minutes
ss	Seconds
tt	If present represent data in AM/PM format
:	Time separator

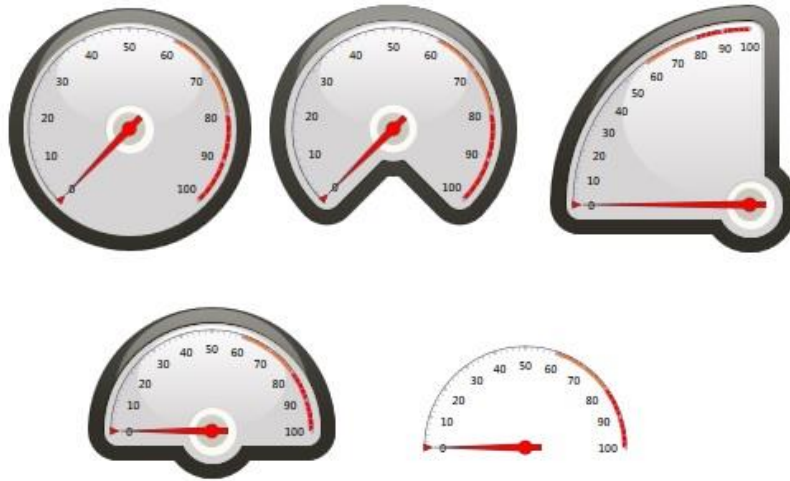


## 5.28. Gauges and Arc Bargraphs

### Gauges

The Movicon NExT Gauges are configurable objects that are used to display specified variable values graphically.

A Gauge comes with a Scale, Bar and indicator that can all be configured using the relative associated properties.

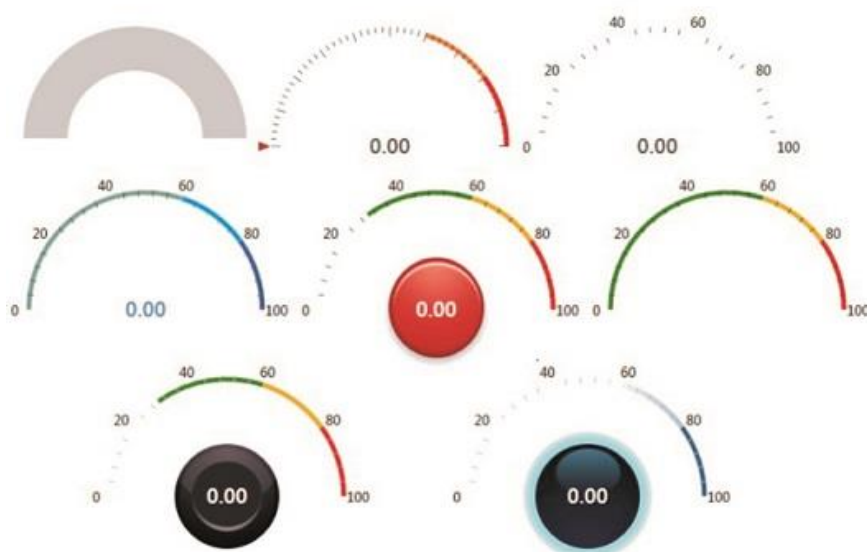


*Some examples of Gauges*

### Arc Bargraph

The Movicon NExT dynamic arcs are configurable objects that display variable values graphically.

A dynamic Arc consists of a Scale and a Bar which can be configured using the relevant properties.



The graphical appearance of these objects can be customized using the theme applications available in the relative stile properties of the various components.

## Gauge Properties

### Gauge Scale Value

**Min. Value:** This is used to set the minimum value allowed.

**Max. Value:** This is used to set the maximum value allowed.

**Use Engineering Unit:** Permits the minimum and maximum units to be inherited from the engineering unit relating to the object's associated Tag.

**Scale Flow Direction:** This is used to set the direction of the scale (from right to left or from left to right).

### Tag Advanced

**Tag Min. Value:** This is used to set the Tag with the allowed minimum value.

**Tag Max. Value:** This is used to set the Tag with the allowed maximum value.

### Gauge Numeric Value

**Show Value:** Shows the current value indicated by the indicator within the object.

**Value Color:** Changes the colour of the value shown.

**Value Font Settings:** Used to set the actual value's style character.

**Value Offset:** Used to set the distance of the displayed valued from the object's border.

**Format Value:** Used to set the value format to be displayed.

### Engineering Unit Label Style

**Show Engineering Unit:** Enables/disables the "Engineering Units" labels

**Engineering Unit Color:** Used to select the "Engineering Units" label's text colour

**Engineering Unit Label offset:** Used to select the offset between label text and borders.

**Engineering Unit Text:** Used to set engineering unit label text.

**Engineering Unit Font:** Used to set the "Engineering Units" label's font type.

### Gauge Scale Labels

**Label Orientation:** Sets the orientation of the scale's values.

**Label Offset:** Sets the distance of the scale's values from the border.

**Label Color:** Set the color of the scale's values.

**Label Font Settings:** Sets the font for scale's values.

**Label String Format:** Sets the labels' text format.

**Show First Label:** Shows the scale's first value.

**Show Last Label:** Shows the scale's last value.

**Scale Line Color:** Sets the color of the Scale Line

### Gauge Style

**Gauge Style:** Sets the Gauge/Bargraph's basic style.

**Auto Hide Elements:** When this property is set to True, the control's elements will gradually disappear and the number of ticks will be reduced along with the labels every time it is reduced in size. For example, as the object is gradually reduced in

**size, the numeric values will disappear by order they come in and the scale's ticks, numeration, indicator and borders will also be reduced.**

Conversely, if this property is set to False, the object's elements will remain visible as it is made smaller to the eventually overlap each other.



This behaviour is determined by the fact that even though the object is made smaller in size, its elements, such as the scale and fonts used, are not resized to match and remain the size they were initially set with.

If you wish to resize the object in a way that all its elements are resized proportionately, you will need to create a symbol with the object's initial sizes. After you have done this, you can then resize the symbol, which will become a Viewbox, and its elements will resize accordingly.

**Gauge Back Color:** Sets the back color for the object's general area.

**Start Angle:** Sets the arc's start angle.

**End Angle:** Sets the arc's end angle.

#### **Gauge Marker Style**

**Marker Visible:** Displays/Hides the marker.

**Marker Height:** Sets the marker's height.

**Marker Width:** Sets the marker's width.

**Marker Offset:** Sets the distance between the marker and the border.

**Marker Color:** Sets the color of the marker.

**Marker Stroke Color:** Sets the color of the marker stroke.

**Marker Animation:** Activates or Deactivates the marker's animation.

**Marker Clickable:** Activates or Deactivates marker interactivity with mouse. If activated the user can set the value by clicking and moving it directly.

#### **Gauge Tickmark Style**

**Tickmark Style:** Sets the object's scale's stile.

**Scale Line Color:** Sets the color of the scale line.

**Major Tickmark Length:** Sets the length of the object's scale's major tickmarks.

**Major Tickmark Layout Index:** Sets the Z-index of the major tickmarks.

**Major Tickmark Thickness:** Sets the thickness of the object's scale's major tickmarks.

**Major Tickmark Offset:** Sets the distance of the scale's major tickmarks from the object's border.

**Show first Major Tickmark:** Displays/Hides the first scale's major tickmarks.

**Show last Major Tickmark:** Displays/Hides the last scale's major tickmark.

**Minor Tickmark Length:** Sets the length of the object's scale's minor tickmarks.

**Minor Tickmark Z-index:** Sets the Z-index of minor tickmarks.

**Minor Tickmark Offset:** Sets the distance between the minor tickmarks and the object's border.

**Minor Tickmark Thickness:** Sets the tickmark's thickness.

**Minor Tickmark Color:** Sets the minor tickmark's color.

**Major Interval Count:** Sets the number of major scale interval dividers.

**Minor Interval Count:** Sets the number of minor scale interval dividers.

### Gauge Spindle Style

**Spindle cap Style:** Sets the style of the object's spindle cap.

**Spindle Color:** Sets the color.

**Spindle Height:** Sets spindle's height.

**Spindle Width:** Sets spindle's width.

### Gauge Needle Style

**Needle Style:** Sets the needle's style.

**Needle Color:** Sets the needle's color.

**Needle Clickable:** Activates/Deactivates the interactivity with the needle. When activated the user will be able to set the value directly by clicking on the needle to move it.

**Needle Visible:** Hides/Shows the needle.

**Needle Animation:** Activates/Deactivates the needle's animation.

**Needle Z-Index:** Sets the Needle's depth level.

**Needle Start Offset:** Sets the needle's distance from the center of the object.

**Needle End Offset:** Sets the needle's distance from the object's border (Pixel).

### Gauge Range Bar Style

**Range Bar Visible:** Enable/disables the Value Range Bar.

**Range Bar Layout Index:** Sets the Object's Range Bar depth index.

**Range Bar Offset:** Sets the space between the Value Range Bar segments.

**Range Bar Color:** Sets the colour of the Value Range Bar.

**Range Bar Back Color:** Sets the back color of the range bar.

**Range Bar Animation:** Enables/disables the animation for the Value Range Bar, if enabled the bar will fill until it reaches the relative position during animation.

**Range Bar Clickable:** Enables/disables the Value Range Bar interactivity.

**Range Bar Thickness:** Sets the thickness of the Value Range Bar.

### Gauge Scale Range Style

**Scale Zone Visible:** Makes the preselected scale zone (zones 1,2 or 3) visible/invisible.

**Scale Zone Start:** Sets the start position of the preselected scale zone (zones 1,2 or 3).

**Scale Zone End:** Sets the end position of the preselected scale zone (zones 1, 2 or 3).

**Scale Zone Thickness:** Sets the thickness of the preselected scale zone (zones 1, 2 or 3).

**Scale Zone Offset:** Sets the offset between the scale and the zone border (zones 1, 2 or 3).

**Scale Zone Color:** Sets the color of the scale zone (zones 1, 2 or 3).

**Scale Zone Layout Index:** Sets the scale zone's Z-index order value (zones 1, 2 or 3).

## Style

**Border Brush:** This is used for setting the container's border colour.

**Border Thickness:** This is used for setting the container's border thickness.

**Back Color:** This is used for setting the container's background colour.

**Stroke Color:** Sets the color of the numbers in the object's foreground.

**Disable Anti-aliasing:** Enables/disables the Anti-aliasing pixel correction function.

**Margin:** Sets the object's margins.

## Fonts

**Font Family:** Sets font model to be used.

**Font Size:** Sets the font's size within the object.

**Font Style:** Sets the font's style.

**Font Weight:** Sets the thickness or thinness of how the font should be displayed ( *light, extralight, normal etc...*).

## Layout

See paragraph on "Common ToolBox object and drawing properties".

## Behaviour

See topic: "Drawing and Object Behaviour Properties".

## General

see topic "Common ToolBox Object and Drawing Properties".

## Visibility

**Transparency:** Sets the object's Transparency level.

**Visibility:** enables the object's visibility.

**Visible on Client:** makes object visible on the webclient

**Visibility Level:** permits you to set the mask to determine the object's visibility.

**Zoom Level Visibility X:** permits you to set the Zoom factor on the x axis after which the object will become visible on screen.

**Zoom Level Visibility Y:** permits you to set the Zoom factor on the y axis after which the object will become visible on screen.

## Tag

See topic on: "Common ToolBox object and drawing properties".

## User Access

See topic on: "Draws and Object User Management" concerning objects and drawings that define the security parameters based on the project's User and Password management.

## Expressions

See topic on: "Managing Expression" concerning objects and drawings.

## Preserve Template Changes

**Preserve:** See topic on: "Preserve local Draws and Objects properties" concerning original library objects.

## 5.29. Hot Region

The Hot Regions constitute zones that can be strategically inserted and positioned on screen and remain invisible but mouse sensitive during project runtime. Like the most of the objects the Hot Regions have execution properties for setting associated commands when the operator clicks within the Hot Region area. When a Hot region is inserted on screen it will appear as a shaded area with default sizes. Once inserted it can be sized as pleased by dragging its borders with the mouse.



The zone of the screen within which the Hot Region is placed, is usually built with a background zone representing a strategic drawing. This zone will be invisible during project runtime but the Hot Region borders will highlight when the mouse pointer hovers within its boundaries. ToolTips can also be made to appear at the same time this happens if desired by the programmer.

### *Hot region properties*

#### Style

**Border Brush:** This is used to set the container's border colour.

**Border Thickness:** This is used to set the container's border thickness.

**Back color:** This is used to set the container's background colour.

**Stroke Color:** This is used to set the color of strokes used in the object.

**Disable Anti-aliasing:** Enables/disables the Anti-aliasing pixel correction function.

**Margin:** This is used to set the object's margins.

#### Fonts

**Font Family:** Sets font model to be used.

**Font Size:** Sets the font's size within the object.

**Font Style:** Sets the font's style.

**Font Weight:** Sets the thickness or thinness of how the font should be displayed ( *light*, *extralight*, *normal* etc...).

#### Layout

See paragraph on "Common ToolBox object and drawing properties".

#### Behaviour

See paragraph on "Drawing and Object Behaviour Properties".

### General

see topic "Common ToolBox Object and Drawing Properties".

### Visibility

**Transparency:** Sets the object's Transparency level.

**Visibility:** enables the object's visibility.

**Visible on Client:** makes object visible on the webclient

**Visibility Level:** permits you to set the mask to determine the object's visibility.

**Zoom Level Visibility X:** permits you to set the Zoom factor on the x axis after which the object will become visible on screen.

**Zoom Level Visibility Y:** permits you to set the Zoom factor on the y axis after which the object will become visible on screen.

### Tag

See topic on: "Common ToolBox object and drawing properties".

### User Access

See topic on: "Draws and Object User Management" concerning objects and drawings that define the security parameters based on the project's User and Password management.

### Expressions

See topic on: "Managing Expression" concerning objects and drawings.

### Kinect

**Speech command:** Used for setting a speech command in the object which will be used during runtime.

### Preserve Template Changes

**Preserve:** see topic on: "Preserve local Draws and Objects properties" concerning original library objects.

## 5.30. Dynamic Images

## 5.31. Animated Images

The "Animated Image" object is used to set a box with an animated image. The image in the box set by the user can also be associated to a variable, or eventually, taken directly from the Image List Editor using the **Smart TAG** settings (located at the end of the object's list of properties).

### Animated Images Properties

#### Animated Image Style

**Control Clip To Bounds:** Clips image to the control's borders.

**Animation Time:** Used to set the transition time between one animation and the next.

**Control's Render Transform Origin:** used to define the control's center point in respect to its animated contents

## Style

**Border Brush:** Sets the color of the container's border.

**Border Thickness:** Sets the thickness of the container's border.

**Back Color:** Set the container's background color.

**Stroke Color:** Sets the stroke color of the displayed text.

**Disable Anti-Aliasing:** This is used for enabling/disabling the Antialiasing function.

**Margin:** Sets the object's margins.

## Fonts

**Font Family:** Sets font model to be used.

**Font Size:** Sets the font's size within the object.

**Font Style:** Sets the font's style.

**Font Weight:** Sets the thickness or thinness of how the font should be displayed (*light, extralight, normal etc...*).

## Layout

See topic on "Common ToolBox object and drawing properties".

## Behaviour

See topic: "Objects Executions Properties".

## General

See topic: "Common ToolBox Object and Drawing Properties".

## Visibility

**Transparency:** sets the object's Transparency level.

**Visibility:** enables the object's visibility.

**Visible on Client:** makes object visible on the webclient

**Visibility Layer Level:** permits you to set the mask to determine the object's visibility.

**Zoom Level Visibility X:** permits you to set the Zoom factor on the x axis after which the object will become visible on screen.

**Zoom Level Visibility Y:** permits you to set the Zoom factor on the y axis after which the object will become visible on screen.

## Tag

See topic: "Common ToolBox object and drawing properties".

## User Management

See topic: "Draws and Object User Management" concerning objects and drawings that define the security parameters based on the project's User and Password management.



## Expressions

See topic: "Managing Expression" concerning objects and drawings.

## Preserve Template Changes

**Preserve:** see paragraph "Preserve local Draws and Objects properties" concerning original library objects.

## 5.32. Media Element Control

The 'Multimedia Viewer' is used to display multimedia, such as video, audio and images, in runtime. This object has a toolbar which can be activated/deactivated through its properties. This toolbar can be used by the user to interact with object such as to control sound volume and reproduction speed.



## Multimedia Viewer Properties

### Media settings

**Source File:** This is used to insert the completed path of the file to be reproduced.

**Loading Command:** This is used to set the command to be executed when object is loaded.

### Style

**Border Brush:** This is used to set the container's border colour.

**Border Thickness:** This is used to set the container's border thickness.

**Foreground Color:** This is used to set the colour of the letter and number characters used in the object.

**Background color:** This is used to set the container's background colour.

**Font Size:** This is used to set the font size used within the object.

**Disable Anti-aliasing:** Enables/disables the Anti-aliasing pixel correction function.

**Margin:** This is used to set the object's margins

## Fonts

**Font Family:** This is used to set the font model to be used.

**Font Size:** This is used to set the font's size within the object.

**Font Style:** This is used to set the font's style.

**Font Weight:** This is used to set the thickness or thinness of how the font should be displayed ( *light*, *extralight*, *normal* etc...).

## Layout

See topic on: "Common ToolBox object and drawing properties".

## Behaviour

See topic on: "Drawing and Object Behaviour Properties".

## General

See topic on: "Common ToolBox Object and Drawing Properties".

## Visibility

**Transparency:** This is used to set the object's Transparency level.

**Visibility:** This is used to enable the object's visibility.

**Visible on Client:** Enable this to make object visible on the webclient

**Visibility Level:** This is used to set the mask to determine the object's visibility.

**Zoom Level Visibility X:** This is used to set the Zoom factor on the x axis after which the object will become visible on screen.

**Zoom Level Visibility Y:** This is used to set the Zoom factor on the y axis after which the object will become visible on screen.

## Tag

See topic on: "Common ToolBox object and drawing properties".

## User Access

See topic on: "Draws and Object User Management" concerning objects and drawings that define the security parameters based on the project's User and Password management.

## Expressions

See topic on: "Managing Expression" concerning objects and drawings.

## Preserve Template Changes

**Preserve:** see topic on: "Preserve local Draws and Objects properties" concerning original library objects.

## 5.33. IP Camera Viewer

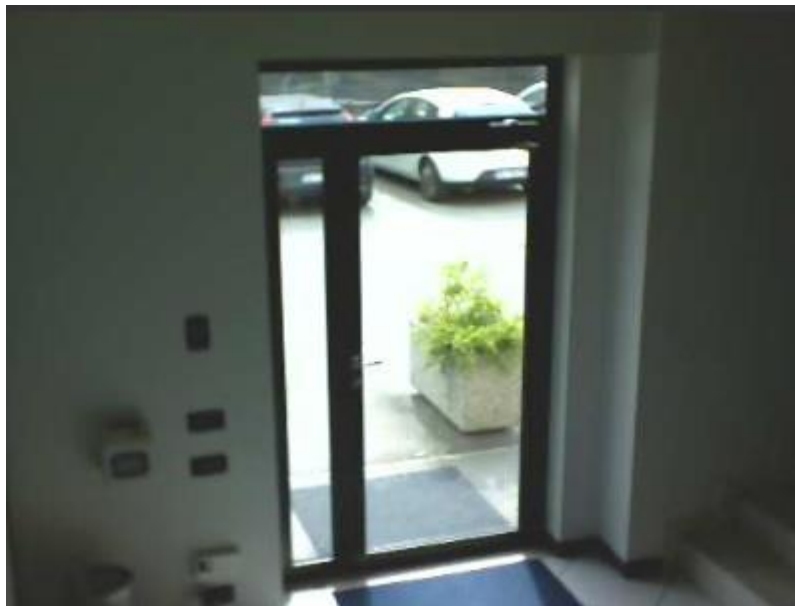
The IP Camera Viewer object is used for streaming video taken by an IP camera. The reproduction of video streaming is only available in runtime mode.



This object can be connected to a protected link by setting the credentials required using the appropriate authentication properties.



The IP Camera object supports streaming only in MJPEG. To show images in other video formats please refer Media Element Control and Stream Media Element



### IP Camera Viewer Properties

#### URL Connection

**IP Camera URL:** This is used to set the URL address for the IP camera image.

**URL User:** This is used to set the URL user's address, if the connection requires a password.

**User Password:** This is used to set the password for user connection.

#### Style

**Border Brush:** This is used to set the container's border colour.

**Border Thickness:** This is used to set the container's border thickness.

**Foreground Color:** This is used to set the colour of the letter and number characters used in the object.

**Background color:** This is used to set the container's background colour.

**Font Size:** This is used to set the font size used within the object.

**Disable Anti-aliasing:** Enables/disables the Anti-aliasing pixel correction function.

**Margin:** sets the object's margins

## Fonts

**Font Family:** Sets the font model to be used.

**Font Size:** Sets the font's size within the object.

**Font Style:** Sets the font's style.

**Font Weight:** Sets the thickness or thinness of displayed font (*light, extralight, normal etc...*).

## Layout

See topic on: "Common ToolBox object and drawing properties".

## Behaviour

See topic on: "Drawing and Object Behaviour Properties".

## General

see topic on: "Common ToolBox Object and Drawing Properties".

## Visibility

**Transparency:** Sets the object's Transparency level.

**Visibility:** Enables the object's visibility.

**Visible on Client:** Enables object visibility on the webclient

**Visibility Level:** This is used to set the mask to determine the object's visibility.

**Zoom Level Visibility X:** This is used to set the Zoom factor on the x axis after which the object will become visible on screen.

**Zoom Level Visibility Y:** This is used to set the Zoom factor on the y axis after which the object will become visible on screen.

## Tag

See topic on: "Common ToolBox object and drawing properties".

## User Access

See topic on: "Draws and Object User Management" concerning objects and drawings that define the security parameters based on the project's User and Password management.

## Expressions

See topic on: "Managing Expression" concerning objects and drawings.

## Preserve Template Changes

**Preserve:** See topic on: "Preserve local Draws and Objects properties" concerning original library objects.

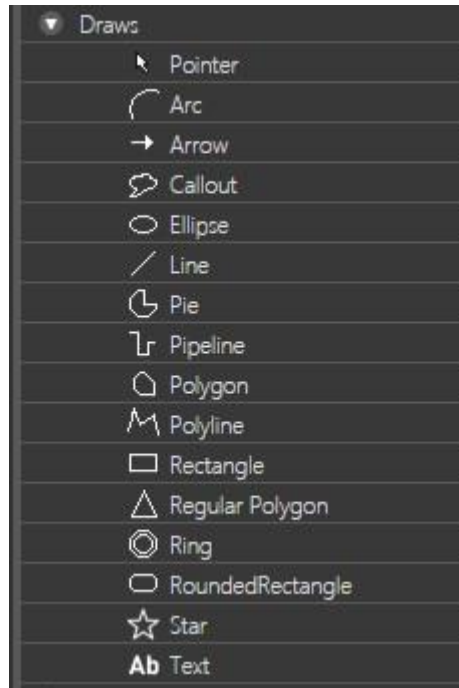
## 5.34. Draws

The Movicon NExT toolbox has a category containing basic shapes called "Draws".

The Movicon 'Draws' category is a collection of primitive shapes that can be used by the user to geometrically configure his or her graphical interface as pleased by using the Graphics Editor. In addition, the user can import their own vectorial drawings using their preferred graphics editor providing that the drawings are defined with the XAML format as required by the WPF technology.



To add new points in a drawing please see the paragraph on "Changing Polygons Points" from the "Graphics Editor" section.



## Common drawing properties

### Style

**Background Color:** Sets the drawing's fill color.

**Stroke Color:** Sets the drawing's border color.

**Stroke Thickness:** Sets the border thickness.

**Disable Anti-Aliasing:** Enables/Disables the Anti-Aliasing function.

**Margins:** Sets the object's margins.

### Style Stroke Details

**Stroke Start Line Cap:** Sets the object's start line cap type (Flat, Square, Round, Triangle).

**Stroke End Line Cap:** Sets the object's end line cap type (Flat, Square, Round, Triangle).

**Stroke Dash Cap:** Sets the drawing's dash cap type (solid stroke dash array types not managed).

**Stroke Line Join:** Sets the line join of two drawing segments < (Miter, Bevel, Round).

**Stroke miter Limit:** used for setting the maximum length of the corner formed by two segments is acute (functions only with the Stoke line join property = miter).

**Stroke Dash Offset:** Sets the dash offset from the drawings staring point.

**Stroke Dash Array:** Sets the drawing's line type. In addition to the options shown on the drop down menu, the user can also set the displayed line type by using an array of values (eg. 1,2,1,3 etc.). In this case, the first value in the array, which corresponds to the 0 index, specifies the dash length, the second value corresponding to the 1 index specifies the space in between, the third specifies the dash length again and so on.

## Fonts

**Font Family:** Sets font model to be used.

**Font Size:** Sets the font's size within the object.

**Font Style:** Sets the font's style.

**Font Weight:** Sets the thickness or thinness of how the font should be displayed ( *light, extralight, normal etc...*).

## Layout

**Horizontal Content Alignment:** Sets the horizontal alignment of all the contents within the object.

**Vertical Content Alignment:** Sets the Vertical alignment of all the contents within the object.

**Width:** Used to change the object's width.

**Height:** Used to change the object's height.

**Horizontal Alignment:** Used to change the object's horizontal alignment.

**Vertical Alignment:** Used to change the object's vertical alignment.

**Lock Object Position:** Used to keep object firm in place on screen.

## General

See paragraph on "ToolBox drawing and object common properties".

## Layout

**Flow Direction:** this is used to set the direction that the user interface text and other elements must flow in the layout within the container.

## Behaviour

See topic: "Drawing and Object Behaviour Properties".

## General

see topic "Common ToolBox Object and Drawing Properties".

## Visibility

**Transparency:** Sets the object's Transparency level.

**Visibility:** enables the object's visibility.

**Visible on Client:** makes object visible on the webclient

**Visibility Level:** permits you to set the mask to determine the object's visibility.

**Zoom Level Visibility X:** permits you to set the Zoom factor on the x axis after which the object will become visible on screen.

**Zoom Level Visibility Y:** permits you to set the Zoom factor on the y axis after which the object will become visible on screen.

**Tag**

See topic: "Common ToolBox object and drawing properties"

**User Management**

See topic: "Draws and Object User Management" concerning objects and drawings that define the security parameters based on the project's User and Password management.

**Expressions**

See topic: "Managing Expression" concerning objects and drawings.

**Preserve Template Changes**

See topic: "Preserve local Draws and Objects properties" concerning original library objects

## Drawing Properties

**Arc**

**Start Angle:** Sets the object's start angle.

**End Angle:** Sets the object's end angle.

**Arc Thickness:** Sets the drawing's thickness.

**Arc Thickness Unit:** Sets the unit of measure for the drawing's thickness.

**Arrow**

**Arrow Orientation:** Set the direction of the arrow.

**Arrowhead Angle:** Sets the arrowhead's angle.

**Arrow Body Size:** Sets the arrow body size in proportion with the arrow head.

**Callout**

**Anchor Point:** Read only property.

**Callout Style:** Allows you to set the drawing's stile.

**Line**

**Start X Point:** Sets the start line coordinates for the X axis.

**Start Y Point:** Sets the start line coordinates for the Y axis.

**End X Point:** Set the end line coordinates for the X axis.

**End Y Point:** Sets the end line coordinates for the Y axis.

**Pipeline**

**Shades Nr.:** Sets the number of shades to show in the pipeline.

**Regular Polygon**

**Point Count:** Sets the number of points to show in the polygon.

**Inner Radius:** creates a concave angle for each side of the polygon: values between 0 and 1 can be used to specify the amplitude of the angles.

**Rounded Rectangle**

**Border thickness:** This is used for setting the border thickness of each of the four rectangle sides.

**Corner Radius:** This is used for setting the bezel for all the four corners.

#### Star

**Point Count:** Sets the number of points to show in the star.

**Inner Radius:** creates a concave angle for each side of the star: values between 0 and 1 can be used to specify the amplitude of the angles.

## 5.35. Potentiometers

The MoviconNExT potentiometers are configurable objects that enable specified variable values to be displayed or set graphically. These objects are available in the "Analog Commands" section in the toolbox.

The Potentiometers consist of a knob (cursor) and the relative Scale which can be both configured by using their associated properties.



### 5.35.1.

#### Potentiometer Properties

##### Potentiometer Scale Style

**Show Labels:** Enables to show or disables to hide the scale value labels.

**Show First Label:** Enables/disables the scales first label

**Show Last Label:** Enables/disables the scales last label

**Label Format:** Used for setting a format type for the labels' values.

**Label Font Settings :** Used for setting the font type to be used in the labels.

**Label Color :** Used for setting the colour of the labels.

**Label Orientation:** Used for setting the label orientations.

**Label Offset:** Used for setting the distance of the lables from the centralised knob (cursor).

##### Potentiometer Tickmark Style

**Major Tick Brush:** This is used to the colour of the major tick.

**Minor tick brush:** This is used to the colour of the minor tick.

**Minor interval count:** This is used to the number of intervals to be shown on knob (for the minor scale).

**Major interval count:** This is used to the number of intervals to show on the object's major scale.



**Tickmark Style:** This is used to set the style of the tickmarks.

**Major Tickmark Length:** This is used to set the length of the major tickmarks.

**Major Tickmark Thickness:** This is used to set the thickness of the major tickmarks.

**Major Tickmark Offset:** This is used to set the major tickmark offset.

**Show First Major Tickmark:** Enable this property to display the first major tickmark.

**Show Last Major Tickmark:** Enable this property to display the last major tickmark.

**Minor Tickmark Length:** This is used to set the length of the minor tickmarks.

**Minor Tickmark Thickness:** This is used to set the thickness of the minor tickmarks.

**Minor Tickmark Offset:** This is used to set the minor tickmark offset.

### Potentiometer Value Ranges

**Value Format:** This is used to set the format for numeric values by defining the number of digits and decimal separators such as 00.0 for example.

**Min. Value:** This is used to set the minimum value allowed.

**Max. Value:** This is used to set the maximum value allowed.

**Use Engineering Unit:** This is used to set the minimum and maximum units to be inherited from the engineering unit relating to the object's associated Tag.

### Potentiometer Knob Style

**Needle Border Brush:** This is used to set the needle's border color.

**Needle Back Brush:** This is used to set the needle's back color.

**Background Brush:** This is used to set the potentiometer's background color.

**Background Border Brush:** This is used to set the potentiometer's knob border color.

**Ring Back Brush:** This is used to set the color of the external ring showing the scale.

**Knob Radius:** This is used to set the potentiometer knob's radius.

### Tag Advanced

**Tag Min. Value:** Permits a Tag to be set to contain the minimum value allowed.

**Tag Max. Value:** Permits a Tag to be set to contain the maximum value allowed.

### Style

**Border Brush:** This is used to set the container's border colour.

**Border Thickness:** This is used to set the container's border thickness.

**Foreground Color:** This is used to set the colour of the letter and number characters used in the object.

**Background color:** This is used to set the container's background colour.

**Disable Anti-aliasing:** Enables/disables the Anti-aliasing pixel correction function.

**Margin:** This is used to set the object's margins

### Fonts

**Font Family:** Sets font model to be used.

**Font Size:** Sets the font's size within the object.

**Font Style:** Sets the font's style.

**Font Weight:** Sets the thickness or thinness of how the font should be displayed ( *light*, *extralight*, *normal* etc...).

#### **Layout**

See topic on: "Common ToolBox object and drawing properties"

#### **Behaviour**

See topic on: "Drawing and Object Behaviour Properties".

#### **General**

see topic on: "Common ToolBox Object and Drawing Properties".

#### **Visibility**

**Transparency:** This is used to set the object's Transparency level.

**Visibility:** This is used to enable the object's visibility.

**Visible on Client:** This is used to make the object visible on the webclient

**Visibility Level:** This is used to set the mask to determine the object's visibility.

**Zoom Level Visibility X:** This is used to set the Zoom factor on the x axis after which the object will become visible on screen.

**Zoom Level Visibility Y:** This is used to set the Zoom factor on the y axis after which the object will become visible on screen.

#### **Tag**

See topic on: "Common ToolBox object and drawing properties"

#### **User Access**

See topic on: "Draws and Object User Management" concerning objects and drawings that define the security parameters based on the project's User and Password management.

#### **Expressions**

See topic on: "Managing Expression" concerning objects and drawings.

#### **Preserve Template Changes**

**Preserve:** See paragraph "Preserve local Draws and Objects properties" concerning original library objects.

## **5.36. Recipe Viewer**

The "Recipe Viewer" is used for displaying a selected recipe layout in a certain part of the screen. Scrollbars will be activated to facilitate navigation in those cases where the recipe layout sizes are bigger than those of the Viewer object.

The screenshot displays the 'Recipe Commands' window, which is organized into several sections. At the top, there are labels for 'Recipe Commands' and 'Recipe Mix Drivers Data'. Below these, the interface is divided into two main horizontal panels. The top panel contains five columns of data fields: 'S7 - Boolean' (with 8 checkboxes labeled BooleanValue01 to BooleanValue08), 'S7 - Byte' (with 3 text boxes labeled ByteValue01, ByteValue02, ByteValue03, each containing 'pz 0'), 'S7 - Int16' (with 3 text boxes labeled Int16Value01, Int16Value02, Int16Value03, each containing 'Kg 0'), 'S7 - UInt16' (with 3 text boxes labeled UInt16Value01, UInt16Value02, UInt16Value03, each containing 'mm 0'), and 'S7 - Int32' (with 3 text boxes labeled Int32Value01, Int32Value02, Int32Value03, each containing 'mm 0'). The bottom panel contains four columns of data fields: 'Modbus - Coils' (with 8 checkboxes labeled Coil01 to Coil08), 'Modbus - Discrete Inputs' (with 8 checkboxes labeled IS01 to IS08), 'Modbus - Holding Register' (with 6 text boxes labeled HR01 to HR06, each containing 'ml 0'), and 'Modbus - Input Register' (with 6 text boxes labeled IR01 to IR06, each containing 'sec 0').



Default object background in development mode remains transparent until associated with a recipe.

## Recipe Viewer Properties

### Recipe viewer style

**Recipe Name:** This is used to set the name of the recipe to be linked to the viewer.

**Allow Edit:** Enables the "edit" command.

**Allow Add New:** Enables the 'Add New' command.

**Allow Remove:** Enables the 'Remove' command.

**Allow Import:** Enables the command for importing recipes.

**Allow Export:** Enables the command for exporting recipes.

**Allow Read:** Enables 'Read' command.

**Allow Write:** Enables the 'Write' command.

**Allow Layout Runtime Edit:** Enables Editing in runtime.

### Style

**Border Brush:** This is used to set the container's border colour.

**Border Thickness:** This is used to set the container's border thickness.

**Foreground Color:** This is used to set the colour of the letter and number characters used in the object.

**Background color:** This is used to set the container's background colour.

**Disable Anti-aliasing:** Enables/disables the Anti-aliasing pixel correction function.

**Margin:** This is used to set the object's margins

## Fonts

**Font Family:** Sets font model to be used.

**Font Size:** Sets the font's size within the object.

**Font Style:** Sets the font's style.

**Font Weight:** Sets the thickness or thinness of how the font should be displayed ( *light*, *extralight*, *normal* etc...).

## Layout

See topic on: "Common ToolBox object and drawing properties"

## Behaviour

See topic on: "Drawing and Object Behaviour Properties".

## General

See topic on: "Common ToolBox Object and Drawing Properties".

## Visibility

**Transparency:** Sets the object's Transparency level.

**Visibility:** enables the object's visibility.

**Visible on Client:** makes object visible on the webclient

**Visibility Level:** permits you to set the mask to determine the object's visibility.

**Zoom Level Visibility X:** permits you to set the Zoom factor on the x axis after which the object will become visible on screen.

**Zoom Level Visibility Y:** permits you to set the Zoom factor on the y axis after which the object will become visible on screen.

## Tag

See topic on: "Common ToolBox object and drawing properties"

## User Access

See topic on: "Draws and Object User Management" concerning objects and drawings that define the security parameters based on the project's User and Password management.

## Expressions

See topic on: "Managing Expression" concerning objects and drawings.

## Preserve Template Changes

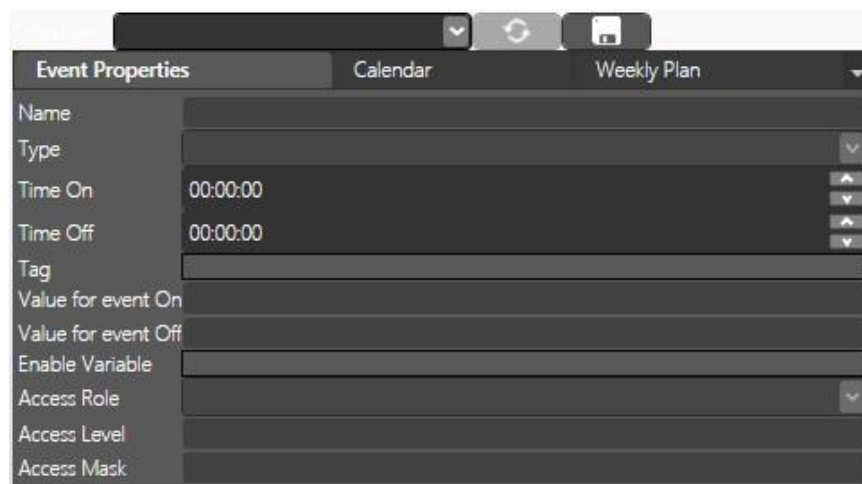
**Preserve:** see topic on: "Preserve local Draws and Objects properties" concerning original library objects.



The "import" and "export" commands in "csv" files are currently not available on the Web Client side.

## 5.37. Scheduler Viewer

The 'Scheduler Viewer' is used to show or edit the project's scheduler resources at runtime. The scheduler resources that are displayed by the Scheduler Viewer object are those that have been enabled with the 'Runtime Editable' in design mode. Those schedulers that have not been enabled with this option will not be displayed or configurable at runtime.



### Scheduler Viewer Properties

#### Connection Strings

**Connection String:** This is used to set the connection string of the object's reference data source.

#### Style

**Border Brush:** This is used to set the container's border colour.

**Border Thickness:** This is used to set the container's border thickness.

**Foreground Color:** This is used to set the colour of the letter and number characters used in the object.

**Background color:** This is used to set the container's background colour.

**Font Size:** This is used to set the font size used within the object.

**Disable Anti-aliasing:** Enables/disables the Anti-aliasing pixel correction function.

**Margin:** sets the object's margins.

#### Scheduler Style

**Start UP Scheduler:** This is used to set the scheduler assigned at startup.

**Weekly event Brush:** This is used to set the Weekly event brush.

**Time Scale:** This is used to set the timescale for the workview.

**Workview Starttime:** This is used to set the workview starttime.

**Workview endtime:** This is used to set the workview endtime.

**Show Workview time only:** This is used to show the workview time only.

#### Fonts

**Font Family:** Sets font model to be used.

**Font Size:** Sets the font's size within the object.

**Font Style:** Sets the font's style.

**Font Weight:** Sets the thickness or thinness of how the font should be displayed ( *light*, *extralight*, *normal* etc...).

### Layout

See topic on: "Common ToolBox object and drawing properties"

### Behaviour

See topic on: "Drawing and Object Behaviour Properties".

### General

See topic on: "Common ToolBox Object and Drawing Properties".

### Visibility

**Transparency:** Sets the object's Transparency level.

**Visibility:** enables the object's visibility.

**Visible on Client:** makes object visible on the webclient

**Visibility Level:** permits you to set the mask to determine the object's visibility.

**Zoom Level Visibility X:** This is used to set the Zoom factor on the x axis after which the object will become visible on screen.

**Zoom Level Visibility Y:** This is used to set the Zoom factor on the y axis after which the object will become visible on screen.

### Tag

See topic on: "Common ToolBox object and drawing properties"

### User Access

See topic on: "Draws and Object User Management" concerning objects and drawings that define the security parameters based on the project's User and Password management.

### Expressions

See topic on: "Managing Expression" concerning objects and drawings.

### Preserve Template Changes

**Preserve:** See topic on: "Preserve local Draws and Objects properties" concerning original library objects.

## 5.38. Selectors

The Movicon.NExT Switches and Selectors are used for executing toggle commands or setting project variables.

These elements are accessible through the "Buttons Switches" group in the toolbox and a major part of their functions and properties resemble those belonging to the

checkboxes. Like the objects from the "Buttons-Checkbox" group, they can be set with tri-state mechanics to assume three different positions.

The tri-state mechanics enables the set variable to assume a new value each time the switch or selector area is clicked on. Therefore with each click the variable passes from the zero value to value one then value two and then back to the zero value.

The switch's/selector's image changes in function with the new variable value,

alternating between three possible images: **switch/selector in zero**

**position, switch/selector in position one** and **switch/selector in position two**. The

variable can also be set by keeping the mouse button pressed down on the selector and moving it to another position. In this case the variable will be set to the new value only when the mouse button is released and the switch/selector is in a different position from the one it started with.



To exploit the tri-state animation to the full use the switched from the "Selector 3P" sub class which are graphically preset for this type of use.



*Some Movicon NExT switches*

## Button-Switch Properties

### Style

**Border Brush:** This is used to set the container's border colour.

**Border Thickness:** This is used to set the container's border thickness.

**Foreground Color:** This is used to set the colour of the letter and number characters used in the object.

**Background color:** This is used to set the container's background colour.

**Font Size:** This is used to set the font size used within the object.

**Disable Anti-aliasing:** Enables/disables the Anti-aliasing pixel correction function.

**Margin:** sets the object's margins

### Button Behaviour

**Is Three State:** This is used to set the selector as tri-state.

**Click mode:** This is used to set the click type to generate command execution (release, press, hover).

## Fonts

**Font Family:** Sets font model to be used.

**Font Size:** Sets the font's size within the object.

**Font Style:** Sets the font's style.

**Font Weight:** Sets the thickness or thinness of how the font should be displayed ( *light*, *extralight*, *normal* etc...).

## Layout

See topic on: "Common ToolBox object and drawing properties"

## Behaviour

See topic on: "Drawing and Object Behaviour Properties".

## General

See topic on: "Common ToolBox Object and Drawing Properties".

## Visibility

**Transparency:** Sets the object's Transparency level.

**Visibility:** enables the object's visibility.

**Visible on Client:** makes object visible on the webclient

**Visibility Level:** permits you to set the mask to determine the object's visibility.

**Zoom Level Visibility X:** permits you to set the Zoom factor on the x axis after which the object will become visible on screen.

**Zoom Level Visibility Y:** permits you to set the Zoom factor on the y axis after which the object will become visible on screen.

## Tag

See topic on: "Common ToolBox object and drawing properties"

## User Access

See topic on "Draws and Object User Management" concerning objects and drawings that define the security parameters based on the project's User and Password management.

## Expressions

See topic on "Managing Expression" concerning objects and drawings.

## Preserve Template Changes

**Preserve:** see topic on "Preserve local Draws and Objects properties" concerning original library objects.

## Miscellaneous

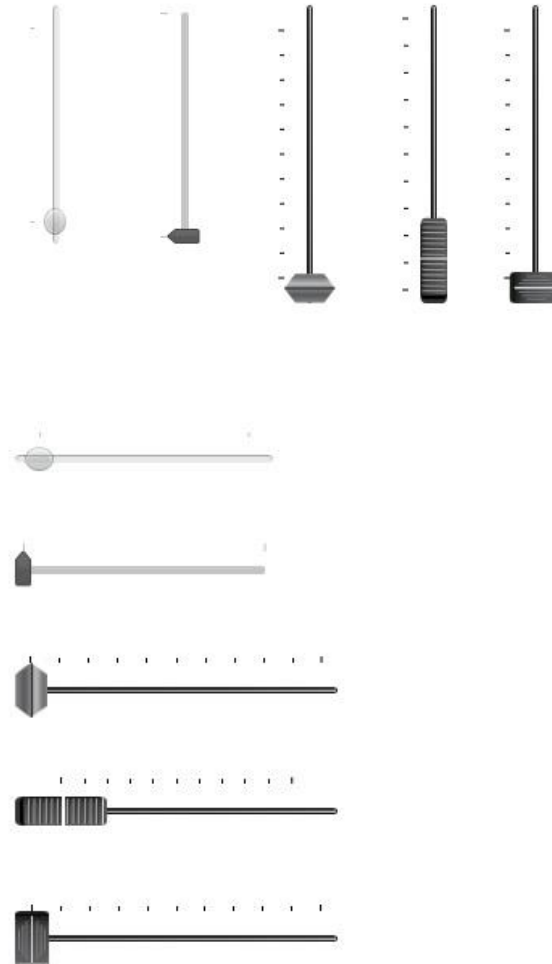
Permits you to edit the object container properties.



## 5.39. Vertical and Horizontal Sliders

The Movicon NExT sliders are configurable sliders that enable valuable values to be displayed or set graphically. The Slider objects are available from the "ToolBox" section in the "Analog Commands" group.

The Slider consists of a Scale and Bar both of which can be configured through their properties.



### Slider Properties

#### Slider Style

**Invert Direction:** This is used to invert the slider direction.

**Bar Color:** This is used to set the color of the bar

**Thumb Color:** This is used to set the color of the cursor

**Tooltip Digits Details:** This is used to set the number in decimals.

**Slider Style:** This is used to set the style type.

#### Slider Scale Style

**Show Scale Labels:** shows or hides the scale labels.

**Tick Major Frequency:** sets the tickmark frequency.

**Tick Color:** This is used to set the tick's color.

**Tick Thickness:** This is used to set the scale's tickmark thickness  
**Tick Minor Size:** This is used to set the tickmark's minor size.  
**Tick Size:** This is used to set the size of the scale's tickmarks.  
**Tick Placement:** This is used to set the position of tick marks along the scale.

### Slider Value Ranges

**Min. Value:** This is used to set the minimum value allowed.  
**Max. Value:** This is used to set the maximum value allowed.  
**Use Engineering Unit:** When enabled the Engineering Units will be used and the scale value limits will be inherited.

### Tag Advanced

**Tag Min. Value:** This is used to set a Tag to contain the minimum value allowed.  
**Tag Max. Value:** This is used to set the Tag to contain the maximum value allowed.

### Slider Behaviour

**Time Rep. Delay:** This is used to set the delay time before the repeat command starts.  
**Time Rep. Interval:** This is used to set the repeat interval time.  
**Snap to Tick:** When enabled the slider will move to the scale marker nearest to where the user clicked in order to change the value.  
**Move to Point:** When enabled the tag value will be forced to the point where the cursor is positioned when clicking.

### Style

**Back color:** This is used to set the container's background colour.  
**Stroke Color:** This is used to set the colour of the letter and number characters used in the object.  
**Decreased Button Color:** This is used to change the slider object's bar color according to the value of the tag assigned to the object.  
**Margin:** This is used to set the object's margins  
**Disable Anti-aliasing:** Enables/disables the Anti-aliasing pixel correction function.  
**Border Thickness:** This is used to set the container's border thickness.  
**Border Color:** This is used to set the container's border colour.

### Fonts

**Font Family:** Sets font model to be used.  
**Font Size:** Sets the font's size within the object.  
**Font Style:** Sets the font's style.  
**Font Weight:** Sets the thickness or thinness of how the font should be displayed ( *light*, *extralight*, *normal* etc...).

### Layout

See topic on: "Common ToolBox object and drawing properties"

### Behaviour

See topic on: "Drawing and Object Behaviour Properties".

### General

**Decrease Button Color:** This is used to decrease the color of the slider object's bar according to the value of the tag assigned to the object.

see topic "Common ToolBox Object and Drawing Properties".

### Visibility

**Transparency:** This is used to set the object's Transparency level.

**Visibility:** This is used to enable the object's visibility.

**Visible on Client:** This is used to make the object visible on the webclient

**Visibility Level:** This is used to set the mask to determine the object's visibility.

**Zoom Level Visibility X:** This is used to set the Zoom factor on the x axis after which the object will become visible on screen.

**Zoom Level Visibility Y:** This is used to set the Zoom factor on the y axis after which the object will become visible on screen.

### Tag

See topic on: "Common ToolBox object and drawing properties"

### User Access

See topic on: "Draws and Object User Management" concerning objects and drawings that define the security parameters based on the project's User and Password management.

### Expressions

See topic on: "Managing Expression" concerning objects and drawings.

### Preserve Template Changes

**Preserve:** see topic on "Preserve local Draws and Objects properties" concerning original library objects.

## 5.40. Switches

### 5.41. Switch Buttons, Lights and Checkbox

The Movicon NExT buttons are divided into two types according to their mechanics but they all have the same style properties. Some buttons have ON-OFF mechanics while others execute COMMANDS.

Therefore you must insert the button type that is most appropriate for the functionality needed.

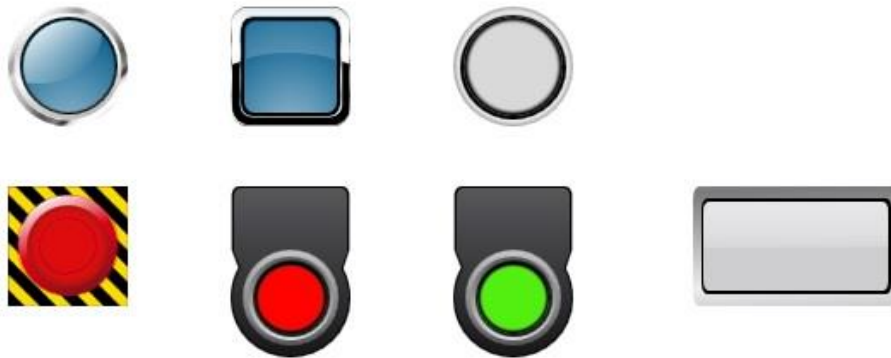


Be reminded that the styles of each object you insert on screen can be modified graphically but their mechanics cannot.

## Buttons-DigitalCheckbox

In Toolbox section you will find a vast range of buttons by type and style. The main characteristic of this category is the possibility to associate a variable for toggling between 0 and 1 (On-Off command).

Even though this object's styles are the same as those of the command buttons, its "Check-Box" mechanics require you to associate it a Tag variable. This can be done by using the Drag & Drop techniques or by associating the Tag using the properties or quick select menu.

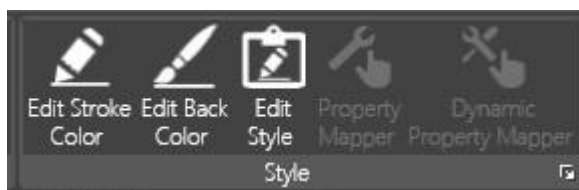


*Some examples of Buttons from the Toolbox*

## Lamps

The Lamp or light objects are luminous indicators that are used for representing the status of the connected variable. This is established by setting the object's behaviour properties to define its mechanics based on two or three states.

The colour of the luminous led colours can be configured using the "edit back color" button from the "Style" ribbon.



These objects are read only for default but the assigned variables can also be controlled using the associated properties.

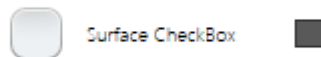


*Some examples of lights and lamps from the Toolbox*

## CheckBox

The checkBox is a classical two state selector button. A Movicon NExT variable can be associated to this control and set to the "0" value (unchecked box) or the "1" value (checked box) according to the state of the box.

The state can be selected with a mouse click or using the keyboard to select the by selecting the component with the TAB or MAIUSC+TAB keys and pressing ENTER.



## Checkbox/lights/Buttons Properties

### Style

**Back Color:** This is used for setting the container's background colour.

**Stroke Color:** This is used for setting the colour of the letter and number characters used in the object.

**Border Color:** Sets the container border's color.

**Border Thickness:** Sets the container border's thickness.

**Disable Anti-aliasing:** Enables/disables the Anti-aliasing pixel correction function.

**Margin:** Sets the object's margins

### Button Style

**Text Wrapping:** this is used to return text to beginning of next line when reaching the border of the box containing it.

**Text Alignment:** This is used to set the text alignment within the button object.

**Pressed Icon:** This is used to set the image to display the button with when in pressed mode.

**Released Icon:** This is used to set the image to display the button with when in released mode.

**Checked Icon:** This is used to set the image to display the button with when in selected mode. This property is only accepted for Toggle button types ("CheckBox").

**Disabled Icon:** This is used to set the image to display the button when in disable mode (and therefore not clickable).

**Overlap Text on Image:** When this option is selected, the button's text will be displayed over lapping the image otherwise the text will be displayed separately in one half of the button with the image displayed at its side in the other half.

**Image Stretch: This is used to adapt image to button**

- *None:* image will not be adapted
- *Fill:* image will be stretched to the size of the button. In this case the image sizes will adapt to those of the button but may be disproportionate.
- *Uniform:* the image will be adapted in proportion to the entire size of the button. In this case, the image may not always cover the entire surface of the button but it will be kept proportionate.
- *UniformToFill:* the image will adapt to the size of the button keeping the image in proportion to its original size. In this case, one of the image's sizes (either height or length) will always adapt but this may cause the other to be cut off.

### Button Behaviour

**Click Mode:** This is used to set the click type to execute the command with (release, press, hover).

**Is Three State:** This is used to set buttons provided with this option to three state mode (available for Checkboxes only).

### Fonts

**Font Size:** This is used to set the font size within the object.

**Font Family:** This is used to set the font family to be used.

**Font Style:** This is used to set the font style type.

**Font Weight:** This is used to set the font type (*light, extralight, normal ecc...*).

### Layout

See topic: "Draws and Object Behaviour Properties".

### Behaviour

See topic: "Draws and Object Behavior Proerties".

### General

See topic: "General Draws and Object Properties".

### Visibility

**Transparency:** This is used to set the object's transparency level.

**Visibility:** Enables the object's visibility.

**Visible On Web Client:** This makes object visible on Web Client.

**Visibility Layer Level:** This is used to set the mask which determines the object's visibility.

**Zoom Level Visibility X:** This is used to set the Zoom level on the X axis after which the object will become visible on screen.

**Zoom Level Visibility Y:** This is used to set the Zoom level on the Y axis after which the object will become visible on screen.

### Tag

See topic: "General Draws and Object Properties".

### User Access

See topic: "User Access Properties" of objects and draws that define the security parameters according to the project's Users and Passwords management.

### Expressions

See topic: "Managing Expressions in Draws and Objects"

### Voice

**Speech Command:** This is used to set a "speech command" in the object that will be used during runtime.

### Preserve Template Changes

See topic: "Preserve properties of local Draws and Object" in respect to the original object in the library.

### Miscellaneous

Allows you to modify the object's container properties.

## 5.42. Toggle Switches

The MoviconNEtT Switches and Selectors are used to execute toggle commands or set project variables.

These elements are accessible through the "Buttons Switches" group in the toolbox and a major part of their functions and properties resemble those belonging to the checkboxes. As with the objects from the "Buttons-Checkbox" group, they can be set with tri-state mechanics to assume three different positions.

The tri-state mechanics enable the set variable to be modified to assume a new value each time the switch or selector area is clicked on. Therefore with each click the variable passes from the zero value to value one then value two and then back to the zero value.

The switch/selector's image changes in function with the new variable value, alternating between three possible images: **switch/selector in zero position**, **switch/selector in position one** and **switch/selector in position two**. The variable can also be set by keeping the mouse button pressed down on the selector and moving it to another position. In this case the variable will be set to the new value only when the mouse button is released and the switch/selector is in a different position from the one it started with.



To exploit the tri-state animation to the full, use the switches from the "Selector 3P" sub class which are graphically preset for this type of use.



*Some Switches available in Movicon.NExT*

## Checkbox/lights/Buttons Properties

### Style

**Back Color:** This is used for setting the container's background colour.

**Stroke Color:** This is used for setting the colour of the letter and number characters used in the object.

**Border Color:** Sets the container border's color.

**Border Thickness:** Sets the container border's thickness.

**Disable Anti-aliasing:** Enables/disables the Anti-aliasing pixel correction function.

**Margin:** Sets the object's margins

### Button Style

**Text Wrapping:** this is used to return text to beginning of next line when reaching the border of the box containing it.

**Text Alignment:** This is used to set the text alignment within the button object.

**Pressed Icon:** This is used to set the image to display the button with when in pressed mode.

**Released Icon:** This is used to set the image to display the button with when in released mode.

**Checked Icon:** This is used to set the image to display the button with when in selected mode. This property is only accepted for Toggle button types ("CheckBox").

**Disabled Icon:** This is used to set the image to display the button when in disable mode (and therefore not clickable).

**Overlap Text on Image:** When this option is selected, the button's text will be displayed over lapping the image otherwise the text will be displayed separately in one half of the button with the image displayed at its side in the other half.

**Image Stretch: This is used to adapt image to button**

- *None:* image will not be adapted
- *Fill:* image will be stretched to the size of the button. In this case the image sizes will adapt to those of the button but may be disproportionate.



- *Uniform*: the image will be adapted in proportion to the entire size of the button. In this case, the image may not always cover the entire surface of the button but it will be kept proportionate.
- *UniformToFill*: the image will adapt to the size of the button keeping the image in proportion to its original size. In this case, one of the image's sizes (either height or length) will always adapt but this may cause the other to be cut off.

### Button Behaviour

**Click Mode:** This is used to set the click type to execute the command with (release, press, hover).

**Is Three State:** This is used to set buttons provided with this option to three state mode (available for Checkboxes only).

### Fonts

**Font Size:** This is used to set the font size within the object.

**Font Family:** This is used to set the font family to be used.

**Font Style:** This is used to set the font style type.

**Font Weight:** This is used to set the font type (*light, extralight, normal ecc...*).

### Layout

See topic: "Draws and Object Behaviour Properties".

### Behaviour

See topic: "Draws and Object Behavior Properties".

### General

See topic: "General Draws and Object Properties".

### Visibility

**Transparency:** This is used to set the object's transparency level.

**Visibility:** Enables the object's visibility.

**Visible On Web Client:** This makes object visible on Web Client.

**Visibility Layer Level:** This is used to set the mask which determines the object's visibility.

**Zoom Level Visibility X:** This is used to set the Zoom level on the X axis after which the object will become visible on screen.

**Zoom Level Visibility Y:** This is used to set the Zoom level on the Y axis after which the object will become visible on screen.

### Tag

See topic: "General Draws and Object Properties".

### User Access

See topic: "User Access Properties" of objects and draws that define the security parameters according to the project's Users and Passwords management.

## Expressions

See topic: "Managing Expressions in Draws and Objects"

## Voice

**Speech Command:** This is used to set a "speech command" in the object that will be used during runtime.

## Preserve Template Changes

See topic: "Preserve properties of local Draws and Object" in respect to the original object in the library.

## Miscellaneous

Allows you to modify the object's container properties.

## 5.43. TAB Screens

Movicon has a component called the "Embedded Screen" which can be inserted within other screens. This object is used for representing various Screens in the project and embedding them into other Screens as vectorial components.

The particular feature of this object is that it displays the designs, components and animation contained in the screen associated to it.

The possibility to create general project layouts with predefined screens will save users a great deal of time when designing engineering projects.

The names of the relative screens are reported in the object's Tabs. These names can be customized and managed with multi-languages using the project's String Table. This is done by simply inserting an ID String with the name and path of the screen so that the ID is automatically substituted at runtime.

For example, if you insert a screen named 'Machine1' in the 'Line' folder within the EmbeddedTabScreen, you will only need to insert an 'Line/Machine1' ID in the String Table in order to display the text corresponding to the ID String in the EmbeddedTabScreen's Tab at runtime according to the language selected.



## Embedded Screen Tab Selector Properties

### Tab Screen Style

**Page Header Alignment:** This is used to set alignment for the tab header.

### Style

**Border Brush:** This is used to set the container's border colour.

**Border Thickness:** This is used to set the container's border thickness.

**Foreground Color:** This is used to set the colour of the letter and number characters used in the object.

**Background color:** This is used to set the container's background colour.

**Font Size:** This is used to set the font size used within the object.

**Disable Anti-aliasing:** This is used to enable/disable the Anti-aliasing pixel correction function.

**Margin:** This is used to set the object's margins

### Fonts

**Font Family:** This is used to set the font model to be used.

**Font Size:** This is used to set the font's size within the object.

**Font Style:** This is used to set the font's style.

**Font Weight:** This is used to set the font's thickness or thinness (*light, extralight, normal etc...*).

### Layout

See topic on: "Common ToolBox object and drawing properties"

### Behaviour

See topic on: "Drawing and Object Behaviour Properties".

### General

**Tab Background:** This is used to set the background color of the TAB to make it consistent with that of the object's.

See topic on: "Common ToolBox Object and Drawing Properties".

### Visibility

**Transparency:** This is used to set the object's Transparency level.

**Visibility:** This is used to enable the object's visibility.

**Visible on Client:** This is used to make the object visible on the webclient

**Visibility Level:** This is used to set the mask to determine the object's visibility.

**Zoom Level Visibility X:** This is used to set the Zoom factor on the x axis after which the object becomes visible on screen.

**Zoom Level Visibility Y:** permits you to set the Zoom factor on the y axis after which the object becomes visible on screen.

## Tag

See topic on: "Common ToolBox object and drawing properties"

## User Management

See topic on: "Draws and Objects User Management" concerning objects and drawings that define the security parameters based on the project's User and Password management.

## Expressions

See topic on: "Managing Expressions" concerning objects and drawings.

## Preserve Template Changes

**Preserve:** see topic on "Preserve local Draws and Objects properties" concerning original library objects.

## 5.44. Text

## 5.45. Texts

The "Text" object is used to set a static text box. The user can also set text within the box and associate a tag to use one of standard animations that are common to all objects.



It is possible to use the following "\n" escape sequence to force the text to go to the beginning of the next line. For example the string:

Item 1\nItem 2\nItem 3

in the visualization phase, for example a text box or button will appear as follows:

- Item 1
- Item 2
- Item 3

## Animated Text Properties

### Style

**Back color:** This is used to set the object's background color

**Stroke Color:** This is used to set the color of the text used within the object.

**Text Wrapping:** This is used to specify whether text should continue on the next line when reaching the box border containing it.

**Enable Horizontal ScrollBar:** Enables the horizontal scrollbar.

**Enable Vertical ScrollBar:** Enables the vertical scrollbar.

**Border Color:** This is used to set the container's border color.

**Border Thickness:** This is used to set the container's border thickness.

**Max. Lines:** This is used to set the maximum number of text lines.

**Min. Lines:** This is used to set the minimum number of text lines.

**Disable Anti-Aliasing:** This is used to Enable/Disable the Antialiasing function.

**Text:** This is used to set the object's Default Text.

**Margin:** Sets the object's margin.

**Text Alignment:** This is used to set the text alignment within the object.

## Fonts

**Font Size:** Sets the font size within the object.

**Font Family:** Sets the font model to be used.

**Font Style:** Sets the font style.

**Font Weight:** Sets the font type to be used (*light, extralight, normal ecc...*).

## Layout

See topic: "General Draws and Object Properties".

## Behaviour

See topic: "Draws and Object Behaviour Properties".

## General

See topic: "General Draws and Object Properties".

## Visibility

**Visibility Layer Level:** Used to set the mask to determine the object's visibility.

**Zoom Level Visibility X:** Used to set the Zoom Level on the X axis after which the object will be made visible.

**Zoom Level Visibility Y:** Used to set the Zoom Level on the Y axis after which the object will be made visible.

**Visible on WebClient:** Makes object visible on WebClient.

**Transparency:** This is used to set the object's transparency level.

**Visibility:** Enables the object's visibility.

## Tag

See topic: "General Draws and Object Properties".

## User Management

See topic: "User Management Properties" of objects and draws that define the security parameters according to the project's User and Password management.

## Expressions

See topic "Expressions in Objects".

### Preserve Template Changes

Preserve: see topic: "Preserve properties of local Draws and Objects" in respect to the original library object.

## 5.46. Animated Text

The animated text box can be configured by using the "Animated Text" object. The text inside this box can be set by the user and then be associated to a tag taken directly from the string table by using the **Smart TAG** settings located at the end of the object's list of properties.



It is possible to use the following "\n" escape sequence to force the text to go to the beginning of the next line. For example the string:

Item 1\nItem 2\nItem 3

in the visualization phase, for example a text box or button will appear as follows:

- Item 1
- Item 2
- Item 3

### Animated Text Properties

#### Animated Text Style

**Default Static Text:** This is used to set the object's Default Text.

**Text Color:** This is used to set the text color.

**Back color:** This is used to set the object's background color

**Text Wrapping:** Use this to specify whether text should continue on the next line when reaching the object's border.

**Text Alignment:** This is used to set the text alignment within the object.

**Text Animation:** Sets the animation style.

**Control Clip To Bounds:** This is used to clip image to control's borders so that it fits to its size

**Animation Time:** Sets the pause time between one animation and the next.

**Control Render Transform Origin:** Allows you to set the control render transform origin

#### Style

**Border Color:** Sets the container's border color.

**Border Width:** Sets the container's border width.

**Stroke Color:** This is used to set the color of the text used within the object.

**Back Color:** Sets the container's back color.

**Disable Anti-Aliasing:** This is used to Enable/Disable the Antialiasing function.

**Margins:** Sets the object's margins.

## Fonts

**Font Size:** Sets the font size within the object.

**Font Family:** Sets the font model to be used.

**Font Style:** Sets the font style.

**Font Weight:** Sets the font type to be used (*light, extralight, normal ecc...*).

## Layout

See topic: "General Draws and Object Properties".

## Behaviour

See topic: "Draws and Object Behaviour Properties".

## General

See topic: "General Draws and Object Properties".

## Visibility

**Visibility Layer Level:** Used to set the mask to determine the object's visibility.

**Zoom Level Visibility X:** Used to set the Zoom Level on the X axis after which the object will be made visible.

**Zoom Level Visibility Y:** Used to set the Zoom Level on the Y axis after which the object will be made visible.

**Visible on WebClient:** Makes object visible on WebClient.

**Transparency:** This is used to set the object's transparency level.

**Visibility:** Enables the object's visibility.

## Tag

See topic: "General Draws and Object Properties".

## User Management

See topic: "User Management Properties" of objects and draws that define the security parameters according to the project's User and Password management.

## Expressions

See topic "Expressions in Objects".

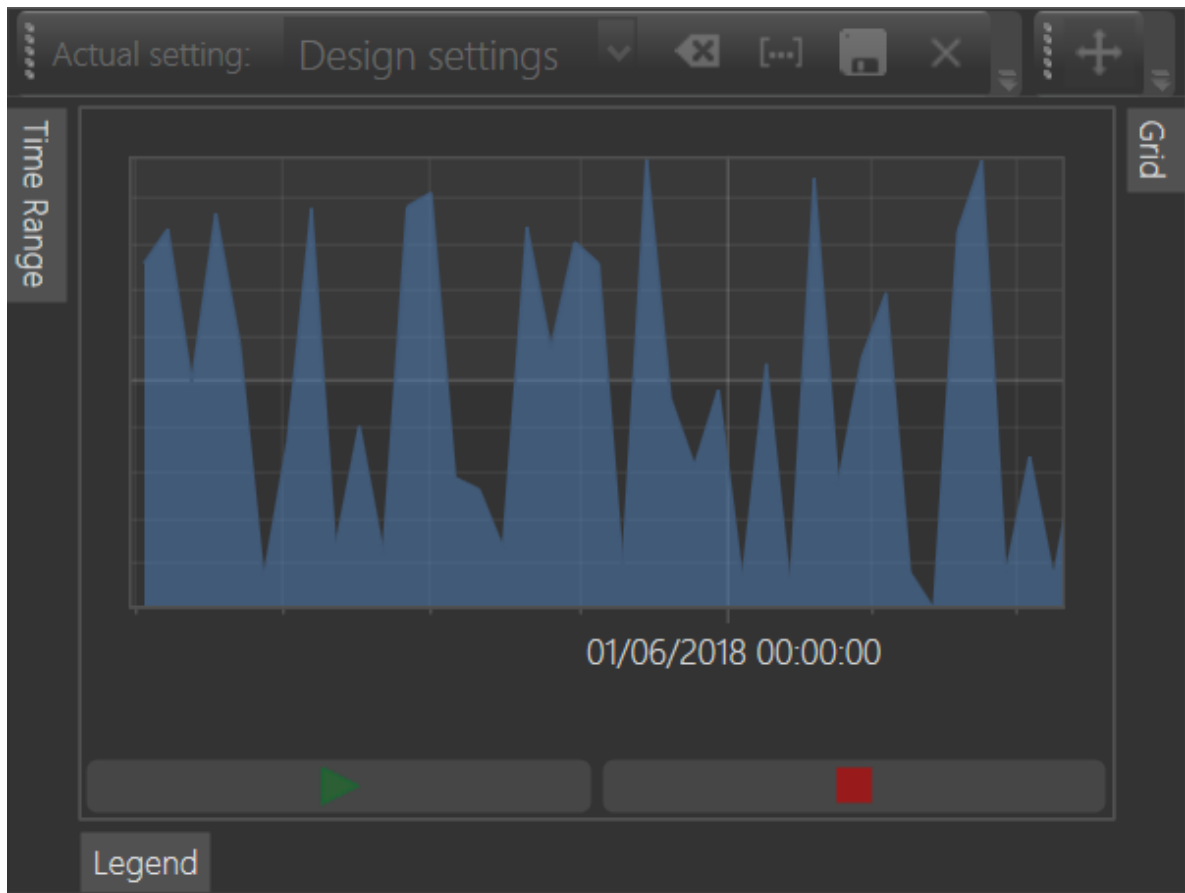
## Preserve Template Changes

Preserve: see topic: "Preserve properties of local Draws and Objects" in respect to the original library object.

## 5.47. Trends

## 5.48. Trend and Data Analysis Viewers

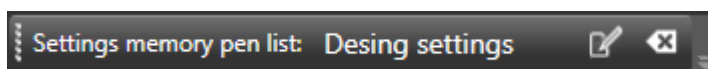
The Trend is used for graphically representing variable trends using curves and analysis of data filed by data recorder engines. Different features are provided according to the trend object being used.



The Trend has a double function. It displays runtime data and logged data (run-pause). The Data Analysis performs analysis on logged data only and provides more advance features for analysing data.

### Configuring the Pen List in Runtime

This functionality can be used in runtime and allows the user to create and manage settings for pen lists to be displayed in the RealTime Trend and Data Analysis. When enabling the 'Allow Runtime Changes' in the Data Analysis/Trend Style property group, a toolbar will display at the top of the object which can be used to access the Pen List's settings editor window.



In addition to displaying the currently active pen configuration in runtime mode, this toolbar also displays two command buttons:

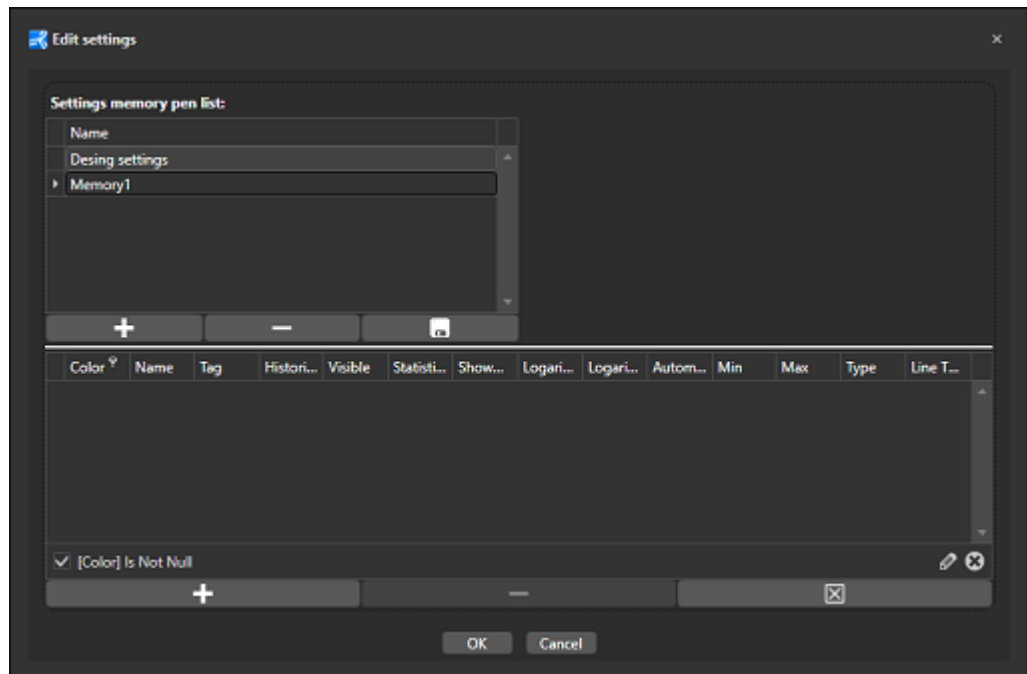
- Edit Settings: when pressing this button a pen list settings editor popup window will appear.



- **Reset Settings:** when pressing this button the default pen settings, defined by the project designer in development mode, will be called and automatically loaded.

### Pen List setting editor window

The popup window used for configuring and managing the pen lists will display as shown in the image below.



This window is composed of two main parts: the top part contains the list of currently available settings for the selected object while the part underneath contains the list of pens associated to the settings shown in the above part.

Different lists of settings can be managed by using the 'Add', 'Remove' and 'Save' commands. These lists will be created by cloning the currently selected settings on the list.

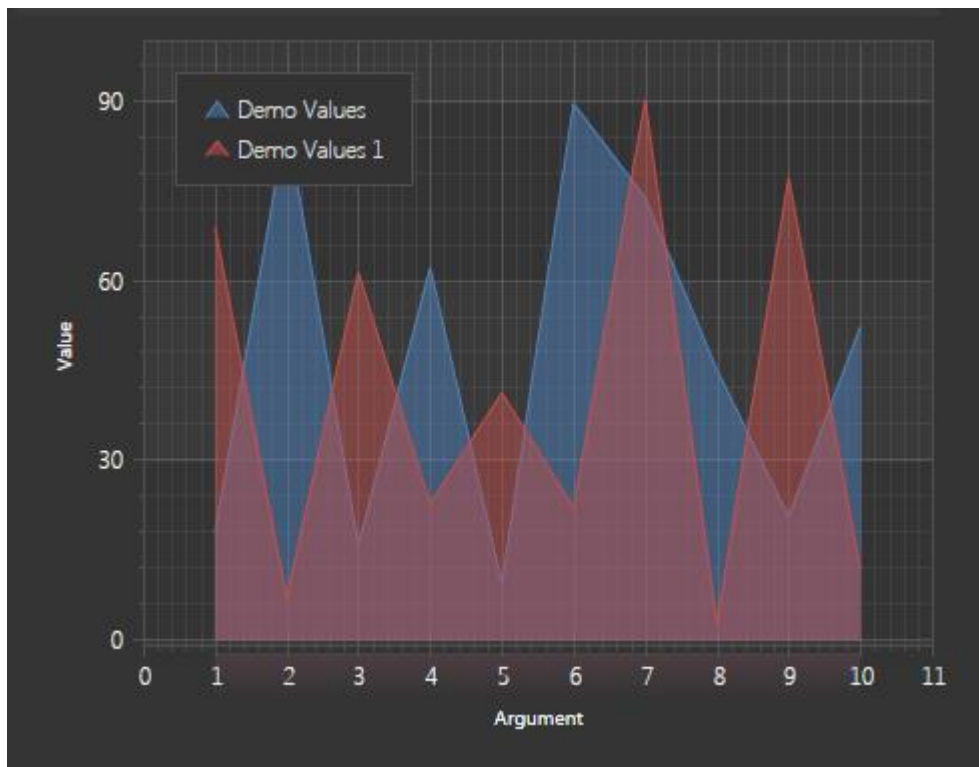
Pens can be added and removed in the same way from the edit settings window using the "Add Pen", "Remove Pen" or "Clear All"

Access to commands linked to the pen list settings in runtime, can be protected using the project user management: simply set the access level desired using the object's "Configuration Write Access Level" and "Configuration Write Access Mask" style properties.



When the currently logged in user does not have the right access level required, they will not be allowed to edit the existing pen list settings but only activate them.

## 5.49. Chart

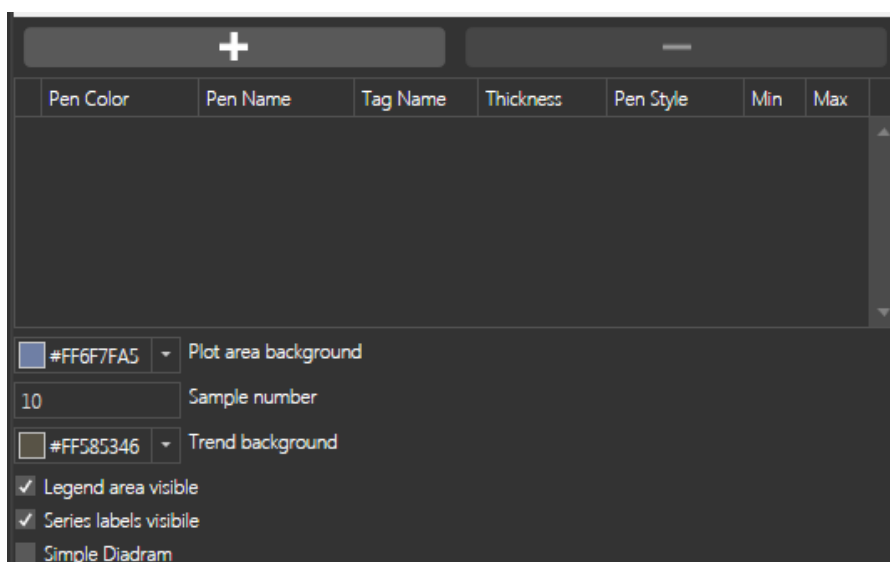


Movicon Next has a window object which can be configured as pleased to create charts in different formats in relation to the Movicon variables.

The charts are built from objects that have different style characteristics compared to the other Movicon objects described so far. These different style types permit more sophisticated and powerful configurations.



The chart object is used for representing the values of different tags by connecting them to pens for which style properties can be configured using the additional property editor.



## Chart Object Properties

### Chart XY Scale Style

**X Value Title:** Sets the title for the X axis.

**Y Value Title:** Sets title for the Y axis.

**XY Title Font Color:** Sets the object's title color.

**Axis Label Foreground:** Sets the color of the axis labels.

**XY Title Font Style:** Sets title's font type.

**XY Axis Font Style:** Sets the font type for the axis labels

**Logarithmic Y Scale:** Enables/disables the logarithmic scale on the Y axis.

**Logarithmic Base Y Scale:** Sets the base for the logarithmic scale of the Y axis (applicable only when "Logarithmic Y Scale" has been activated).

**Scale Max. Value:** Sets the maximum value of the Y axis.

**Scale Min. Value:** Sets the minimum value of the Y axis.

**Automatic Scale:** When enabled, the pens will be automatically scaled in the chart.

**Use Engin. Unit for Min/Max Values:** Enables/disables the displaying of the scale's MIN and MAX values by inheriting them from the EUs relating to the TAGs associated to the pens.

### Chart Style

**Enable animation:** Enables/disables the initial curve drawing animation.

**Animation Mode:** This is used to set the animation mode for the object.

**Mirror Height:** This is used you to set the dimension of the mirror for the object.

**Show Label values on Plot:** Enables/disables the visibility of labels showing reference values on the curve.

**Resolve Label Overlapping Mode:** This is used to set the behaviour type to be maintained while labels overlap.

**Toolbar Foreground:** This is used to set the toolbar foreground.

**Plot Area Background Color:** This is used to set the curve area's background colour.

**Trend Area Background Color:** This is used to set the Trend area's background colour.

**Toolbar Background:** Allows you to set the toolbar Background color.

**Rotated:** Enables/disables the object's rotation.

**Enable 3D:** This is used to set the 3D visualization.

**Simple Diagram Chart Style:** Enables/Disables the curve diagram visualization.

**Zoom 3D Percent:** This is used to set the zoom fact for 3D visualization.

**Pie Chart:** Enables visualization of a simple diagram showing data in pie chart.

**Auto Hide Toolbar:** Enables/disables the function to automatically hide object's top toolbar.

### Chart Advanced

**Linked Pen Name:** This is used to set the name for the pens associated to the chart.

### Chart Legend Style

**Legend Area Visible:** Enables/disables the Legend area within the object.

**Legend Margin:** This is used to set the Legend's margin's value.

**Legend Horizontal Position:** This is used to set the legend area's horizontal position.

**Legend Vertical Position:** This is used to set the legend area's vertical position.

**Legend Max Height:** This is used to set the legend's maximum height.

### Chart Execution

**Sample Number:** This is used to set the number of samples to be represented on the curve.

**Push Tag:** Inserts the "Push" command's tag's name. When setting this variable to '1', the actual value will be inserted in the chart.

**Reset Tag:** Inserts the "Reset" command's tag's name. When setting this variable to the '1' value will eliminate the chart's buffer.

**Zoom Tag:** Inserts the "Zoom" command's tag's name. Setting this variable to "1" will activate the zoom.

**Enables Runtime Changes:** Enables/disables the object's runtime changes.

### Style

**Border Brush:** This is used to set the container's border colour.

**Border Thickness:** This is used to set the container's border thickness.

**Back Color:** This is used to set the container's background colour.

**Stroke Color:** This is used to set the color of the numbers in the object's foreground.

**Disable Anti-aliasing:** Enables/disables the Anti-aliasing pixel correction function.

**Margin:** This is used to set the object's margins.

### Fonts

**Font Family:** Sets font model to be used.

**Font Size:** Sets the font's size within the object.

**Font Style:** Sets the font's style.

**Font Weight:** Sets the thickness or thinness of how the font should be displayed ( *light*, *extralight*, *normal* etc...).

### Layout

See paragraph on "Common ToolBox object and drawing properties"

### Behaviour

See topic: "Objects Executions Properties".

### General

see topic "Common ToolBox Object and Drawing Properties".

### Visibility

**Transparency:** Sets the object's Transparency level.

**Visibility:** enables the object's visibility.

**Visible on Client:** makes object visible on the webclient

**Visibility Level:** permits you to set the mask to determine the object's visibility.

**Zoom Level Visibility X:** permits you to set the Zoom factor on the x axis after which the object will become visible on screen.

**Zoom Level Visibility Y:** permits you to set the Zoom factor on the y axis after which the object will become visible on screen.

### **Tag**

See paragraph on "Common ToolBox object and drawing properties"

### **User Access**

See topic on "Draws and Object User Management" concerning objects and drawings that define the security parameters based on the project's User and Password management.

### **Expressions**

See paragraph "Managing Expression" concerning objects and drawings.

### **Preserve Template Changes**

**Preserve:** see paragraph "Preserve local Draws and Objects properties" concerning original library objects.

## **Additional Smart Tag Properties**

**Pen Color:** used for selecting the pen's color.

**Pen Name:** This field is used to insert a name for the pen with which it will be displayed in the legend. It is also possible to insert a String ID to manage language change at runtime.

**Tag X:** This field is used to select the tag to associate to the X axis.

**Tag Y:** This field is used to select the tag to associate to the Y axis.

**Thickness:** This field is used to set the pen's line thickness.

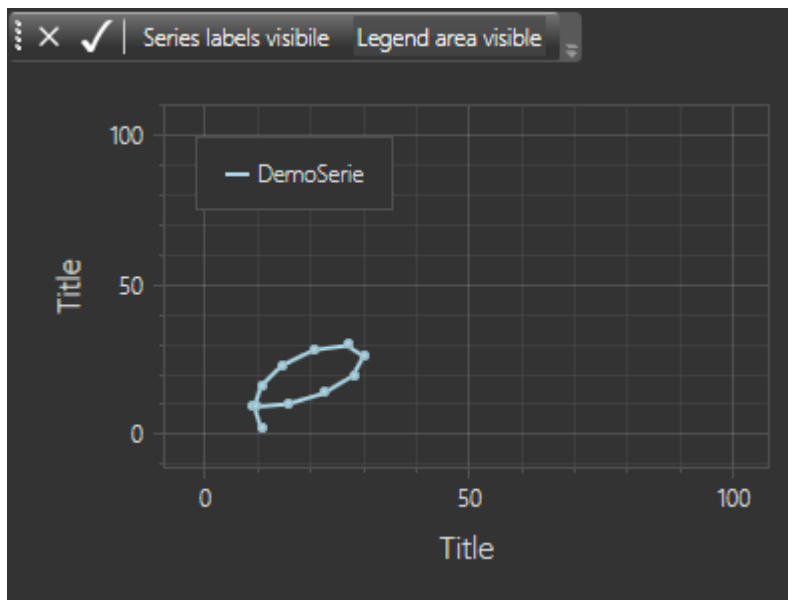
**Pen Style:** This field is used to select the style to display the pen with.

**Min, Max:** These two values define the pen's minimum and maximum display limits. In cases where an Engineering Unit has been associated to the Tag associated to the pen, these two fields will be ignored and the Minimum and Maximum values will be taken directly from the Engineering Unit.

## **5.50. Chart XY**

Chart XY is one of many graphical objects provided by Movicon.NExT to represent TAG values which have been programmed in the project.

Each point, called PENS, presented in the Chart XY graphics has a X coordinate associated to the value of a TAG and a Y coordinate associated to the value of another TAG.

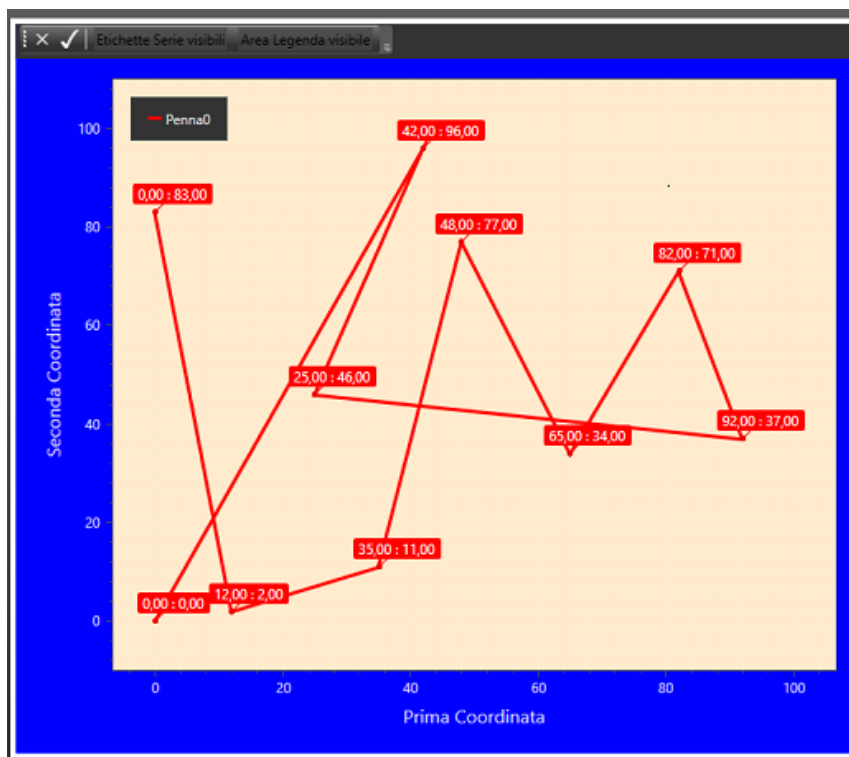


When the values of the 2 tags change, the point will move in the chart until the Force Value command in the object's Toolbar is used to fix its positions.

When this is done several times and when the values of associated tags change, the chart will eventually populate with points until the maximum value, set by the programmer the configuration phase, has been reached (Sample Number parameter).

The following 4 commands in the ChartXY's ToolBar are used to:

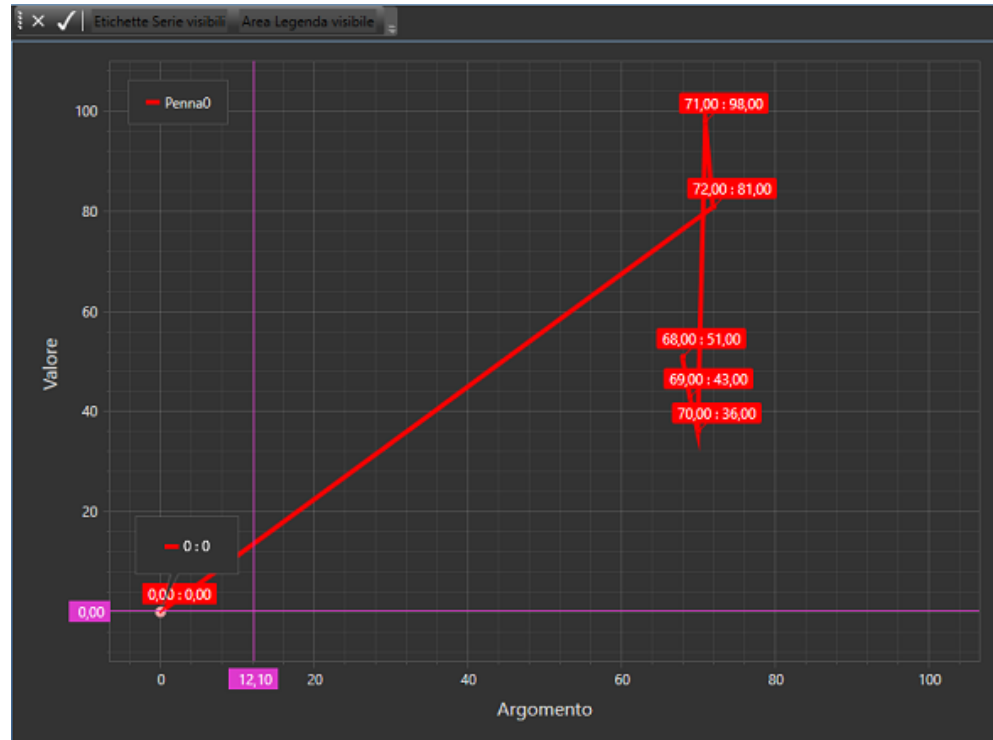
- Reset the displayed samples
- Run samples
- Display labels containing values of the sampled points
- Display the legend containing the names of displayed pens



Example of a Chart XY in runtime displaying only one pen

Several pens can be displayed on a Chart XY at the same time. The maximum limit is determined by the user's ability to read the chart.

Arrays can also be presented on the Chart XY: numeric array values can be associated to the x and y coordinates of a pen and played on the chart xy. When arrays are used, it is not necessary to populate the chart with successive samples of the values associated to the pens as all the values contained in the arrays are presented instantaneously.



Example of a Chart XY showing value arrays

## Chart XY Object Properties

### Chart XY Scale Style

**X Value Title:** This is used for setting the title of the x axis.

**Y Value Title:** This is used for setting the title of the y axis.

**XY Title Font Color:** This is used for setting the color of the object's title.

**Axis Label Foreground:** This is used for setting the color for the axis' labels

**XY Title Font Style:** This is used for setting the object's title font.

**XY Axis Font Style:** This is used for setting the font for the object's Axis' font.

**Logarithmic Y Scale:** This is used for enabling/disabling the logarithmic scale on the Y-axis.

**Logarithmic Base Y Scale:** This is used for setting the value of the y axis in logarithmic scale (can only be activated with the y axis logarithmic scale = true)

**Scale Max. Value:** This is used for setting the maximum value for the Y scale.

**Scale Min. Value:** This is used for setting the minimum value for the Y scale.

**Use Engineering Unit:** Activates the use of the UI values for the Y Scale's Min. and Max. values.

**Automatic Scale:** This is used to enable the automatic scale for the X and Y axis. In this case the scale of the two axis will automatically adapt to the maximum value obtained by the pen.

### Chart XY Style

**Enable Animation:** This is used for enabling the object's graphical animation.

**Animation Mode:** This is used for defining the object's animation type.

**Show Label values on plot:** Enables/disables labels to show on the chart's plotted curve at the side of the various samples.

**Resolve Label Overlapping Mode:** Used for setting the overlapping value label resolution.

**Toolbar Foreground:** Used for setting the toolbar color.

**Plot Area Background Color:** Used for setting the background color for the chart area.

**Trend Area Background Color:** Used for setting the background color the trend area.

**ToolBar Background:** Used for setting the background color for the toolbar area.

**Rotated:** Enables/disables the object's oriented rotation.

**Auto Hide Toolbar:** Enables/disables the autonomous hide toolbar function.

### Chart XY Advanced

**Linked Pen Name:** Used for assigning the name of the pen associated to the chart.

### Chart XY Legend Style

**Legend Area Visible:** When enabled, makes the Legend Area visible in the object.

**Legend Margin:** This is used for setting the legend area margin values.

**Legend Horizontal Position:** This is used for setting the Legend Area's horizontal position.

**Legend Vertical Position:** This is used for setting the Legend Area's vertical position.

### Chart XY Execution

**Sample Number:** This is used for setting the number of samples to show on the chart.

**Push Tag:** This is used for associating a Tag to insert a new value on the chart.

**Reset Tag:** this is used for setting the tag that will be used for the reset command of sample values on the chart.

**Enable Runtime Changes:** Used for modifying object's aspect and style in runtime.

### Style

**Border Brush:** This is used for setting the container's border colour.

**Border Thickness:** This is used for setting the container's border thickness.

**Back Color:** This is used for setting the container's background colour.

**Stroke Color:** Sets the color of the numbers in the object's foreground.

**Disable Anti-aliasing:** Enables/disables the Anti-aliasing pixel correction function.

**Margin:** Sets the object's margins.

### Fonts

**Font Family:** Sets font model to be used.

**Font Size:** Sets the font's size within the object.

**Font Style:** Sets the font's style.

**Font Weight:** Sets the thickness or thinness of how the font should be displayed ( *light*, *extralight*, *normal* etc...).



## Layout

**Horizontal Content Alignment:** used for setting the horizontal alignment of the object's contents.

**Vertical Content alignment:** used for setting the vertical alignment of the object's contents.

**Width:** used for changing the object's width.

**Height:** used for changing the object's height.

**Horizontal Alignment:** used for changing the object's horizontal alignment.

**Vertical Alignment:** used for changing the object's vertical alignment.

**Lock Object Position:** Blocks the object in its position on screen.

## Behaviour

See topic: "Objects Executions Properties".

## General

See topic on "Proprietà comuni oggetti e disegni ToolBox".

## Visibility

**Transparency:** Sets the object's Transparency level.

**Visibility:** enables the object's visibility.

**Visible on Client:** makes object visible on the webclient

**Visibility Level:** permits you to set the mask to determine the object's visibility.

**Zoom Level Visibility X:** permits you to set the Zoom factor on the x axis after which the object will become visible on screen.

**Zoom Level Visibility Y:** permits you to set the Zoom factor on the y axis after which the object will become visible on screen.

## User Access

See topic on "Draws and Object User Management" concerning objects and drawings that define the security parameters based on the project's User and Password management.

## Expressions

See topic on "Managing Expression" concerning objects and drawings.

## Preserve Template Changes

**Preserve:** see paragraph "Preserve local Draws and Objects properties" concerning original library objects.

## Additional Smart Tag Properties

**Pen Color:** indicate the pen's color.

**Point Label Color:** sets the selected point label color.

**Pen Name:** This field is used to insert a name for the pen with which it will be displayed in the legend. It is also possible to insert a String ID to manage language change at runtime.

**Tag X:** This field is used to select the tag to associate to the X axis.

**Tag Y:** This field is used to select the tag to associate to the Y axis.

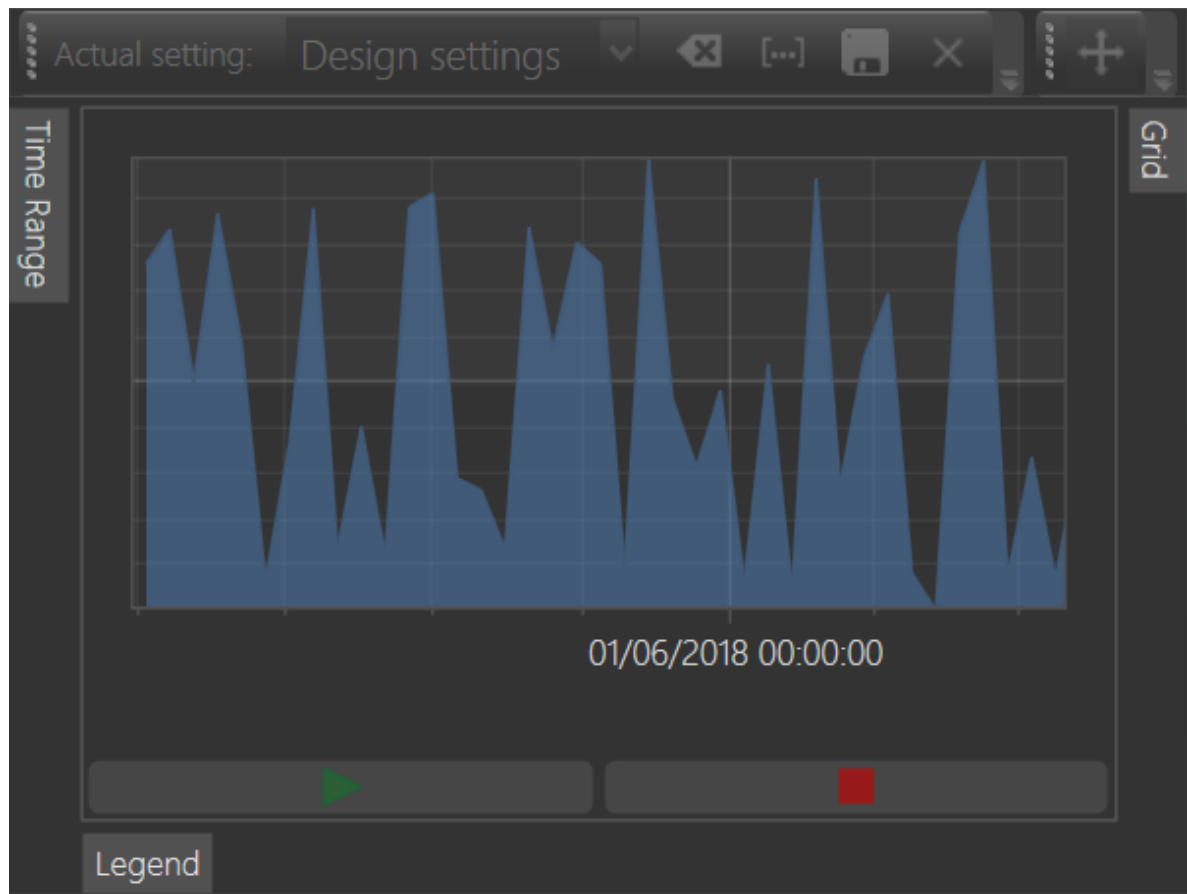
**Thickness:** This field is used to set the pen's line thickness.

**Min, Max:** These two values define the pen's minimum and maximum display limits. In cases where an Engineering Unit has been associated to the Tag associated to the pen, these Minimum and Maximum values will be taken directly from the Engineering Unit ignoring these two fields.

## 5.51. Data Analysis RT

The Data Analysis can be used to display both real-time and Historian data.

It has two "RunStop" and "OnlyStop" operating modes. When selecting the first mode, the values will be displayed in real-time mode. Selecting the second one will allow you to carry out an in-depth analysis on Historian data.



The Data Analysis object can be defined with a list of pens to represent data graphically both in runtime (made possible by using the **smart tags** property bar) and in development mode using the appropriate buttons displayed in the object's 'legend' window.



The following shortcuts can be used to zoom a portion of the graph:

- keep the SHIFT pressed down, select the area with the mouse and Drag&Drop until you reached the enlarged zoom size desired
- use SHIFT+CLICK to zoom the cursor position
- use ALT+CLICK to restore zoom back to previous setting.



If you zoom in with the mouse wheel, it will only work along the axis of the abscissas.

Instead the pinch-to-zoom used in touch screens works along both the axis of the abscissas and the axis of the ordinates.

The Data Analysis also provides tools for representing the analysis and comparisons of data deriving from the DB of the associated Historian. These tools are made available to the user in runtime in the Toolbar at the top of the object.



The buttons are described below in the order they are shown on the Toolbar:

- **Designer:** permits access to the chart designer to customize the window's graphical appearance.
- **Refresh:** Reloads data based on the current time (currently set).
- **Show or Hide cross hair cursor:** Shows or hides the cross hair cursor which appears in correspondence to the sampled values.
- **Show or Hide labels:** Shows or hides the fixed labels that appear in correspondence to the sampled values.
- **Expand or Collapse:** Removes the toolbar by expanding the graph in the screen area.
- **Previous/Next:** Displays the previous or next page relating to the time range selected in the box at the side.
- **Show all Value:** loads all the sampled data.
- **Show Last Minute Value:** Sets the reference base time (minutes) for representing data in graphs.
- **Show Last Hour Value:** Sets the reference base time (hours) for representing data in graphs.
- **Show Last Day Value:** Sets the reference base time (days) for representing data in graphs.
- **Show Last Week Value:** Sets the reference base time (weeks) for representing data in graphs.
- **Show Last Month Value:** Sets the reference base time (months) for representing data in graphs.
- **Show Last Year Value:** Sets the reference base time (years) for representing data in graphs.

## Data Analysis properties

### Data Analysis Style

**Point Precision:** Used to set the number of precision digits of points in the DA chart.

**Diagram Background color:** This is used to set the Diagram's background color.

**Plot Background color:** This is used to set the Plot's background color.

**Axis Label Foreground:** This is used to select the color for the foreground of Axis labels.

**Legend Area Foreground:** Used to set the Legend's foreground color.

**Show Toolbar Show HideButtons:** Activates/Deactivates button toolbars.

**Rotated:** Rotates the data analysis graph area from horizontal to vertical orientation.

**Collapsed Width:** Used to set the object's width when collapsed to "toolbar"

**Collapsed Height:** Used to set the object's height when collapsed to "toolbar"

**Auto-Hide ToolBar:** Enables or disables the automatic toolbar hide function.

**Max Label Points:** Used to set the maximum label points on the object.

**Max Resolve Overlapping Label Points:** Used for setting the maximum overlapping label points.

**Show Toolbar AdvSettings:** Activates/Deactivates advanced settings/buttons on toolbar.

**Show Palette Tab:** Enables/disables the toolbar that contains the "Palette" object.

**Show Toolbar Time Controls:** Enables/Disables the TimeControl options on Toolbar.

**Show Toolbar Max Records:** Enables/Disables the record button options on Toolbar.

**Show Toolbar Compare:** Enables/Disables compare button options on Toolbar

**General Background color:** This is used to set the general background color.

**Toolbar Background:** Allows you to set the toolbar Background color.

**Toolbar Foreground:** Allows you to set the toolbar foreground.

**Use Absolute Ranges:** When selected shows absolute or relative range data.

**LinkedPenName:** This is used to set the name of the linked pen.

**Runtime Mode:** This is used to enable the DA to function in runtime mode.

**Date/Time Format:** This is used to set a string to change the object's Date/Time style. For further information on custom format string type, please refer to the topic on Strings.

**Filter Type:** This is used to select the type of filter to use for extracting data.

**Show Statistic Lines:** Enables/Disables the visibility of the statistic line in the graph (Min, Max, Average).

**Always Expanded:** Expand the data analysis graph area to fit all the screen window area making the toolbars and other areas invisible.

**Disable Zoom Behaviour:** This is used to disable the possibility to apply the zoom in the Data Analysis at runtime.

### Database Settings

**Connection String:** Sets the connection string to the DB (leave blank to use default connection string).

**Max Records:** Sets the maximum number of records to be extracted from the DB.

**Use aggregation:** Enables/disables the data aggregation extracted from DB.

**Max Aggregation Factor:** Sets the factor of the data aggregation extracted from DB.

### Data Analysis Scale Style

**Logarithmic Base Y-Scale:** This is used to set the logarithmic scale's base value (functions only when previous option has been activated).

**Logarithmic Y-Scale:** Enables the logarithmic scale for the Y axis.

**Scale Padding Factor:** This is used to set the padding factor or space between the borders and the scales.

**General Automatic Scaling:** This option enables the scale management at general level. The scale will be the same one for all pens and will assume the minimum value of the different pens' minimums and the maximum value of the different pens' maximums. Each pen will then be traced according to the general minimal and maximum values.

### Data Analysis Grid Style

**x/y Minor Grid line color:** Used to select the color of the minor grid lines on the x/y axes.

**x/y Minor Grid Line visible:** Used to display the minor grid lines on the x/y axes.

**x/y Grid Line color:** Used to select the color of the major grid lines on the x/y axes.

**x/y Grid Line visible:** Used to display the major grid lines on the x/y axes.

**x/y Auto Grid:** Activates the use of the automatic grid scaling on the x/y axes.

**Axis Label color:** Changes the colour of the labels on the x and y axes.

**Max. Errors Before Aborting:** This is used to set the maximum number of data acquisition attempts that the database can make when in Deadlock.

You will need to increase the value of this property in cases in which Deadlock errors occur in the points acquisition phase, keeping in mind this tends to increase due to the increase of the following factors: the number of samples set in the DataAnalysis, the size of the table from which data is read, the requested time range and the number of pens.

### Style

**Border Brush:** This is used for setting the container's border colour.

**Border Thickness:** This is used for setting the container's border thickness.

**Foreground Color:** This is used for setting the colour of the letter and number characters used in the object.

**Background color:** This is used for setting the container's background colour.

**Font Size:** This is used for setting the font size used within the object.

**Disable Anti-aliasing:** Enables/disables the Anti-aliasing pixel correction function.

**Margin:** Sets object's margins

### Fonts

**Font Family:** Used to set the font model to be used.

**Font Size:** Used to set the font's size within the object.

**Font Style:** Used to set the font's style.

**Font Weight:** Used to set the thickness or thinness of how the font should be displayed (*light, extralight, normal etc...*).

### Layout

See topic: "Common ToolBox object and drawing properties"

## Behaviour

**Runtime Configurable:** Enables/disables the editing of the object at runtime (a pop-up window will appear to allow user to edit object).

See topic: "Drawing and Object Behaviour Properties".

## General

**Use Absolute Ranges:** When enabled, this property allows you to define the time range within which to analyse data during runtime. The time range can be set with a fixed time (e.g. one year is  $Y = 1/1$  to  $31/12$ ) or in relation to a particular date (e.g.  $y =$  current date until the next 364 days).

**Point Precision:** This property is used to establish the number of decimal figures to display in the tooltip that shows the points' values when the cursor is moved along the curve. This setting also has effect on values showing in the curve's labels when enabled.

**Operating Mode:** This property is used to select and enable the "RunStop" or quella "OnlyStop" operating mode.

**Filter Type:** Sets a temporal filter to load data from the database. The options are:

- All
- Minute
- Hour
- Day
- Week
- Month
- Year

This setting can be changed using script code in runtime by means of the Data Analysis object's "FilterType" property.

see the topic "Common ToolBox Object and Drawing Properties".

## Visibility

**Transparency:** Used to set the object's Transparency level.

**Visibility:** Used to set enable the object's visibility.

**Visible on Client:** Used to make object visible on the webclient

**Visibility Level:** Used to set the mask to determine the object's visibility.

**Zoom Level Visibility X:** Used to set the Zoom factor on the x axis after which the object will become visible on screen.

**Zoom Level Visibility Y:** Used to set the Zoom factor on the y axis after which the object will become visible on screen.

## Tag

See topic: "Common ToolBox object and drawing properties"

## User Management

See topic on "Draws and Object User Management" concerning objects and drawings that define the security parameters based on the project's User and Password management.

## Expressions

See topic: "Managing Expression" concerning objects and drawings.

## Preserve Template Changes

**Preserve:** See topic: "Preserve local Draws and Objects properties" concerning original library objects.

## Data Analysis Data Time Options

**Current Value Label Background:** This is used to set the labels' background color.

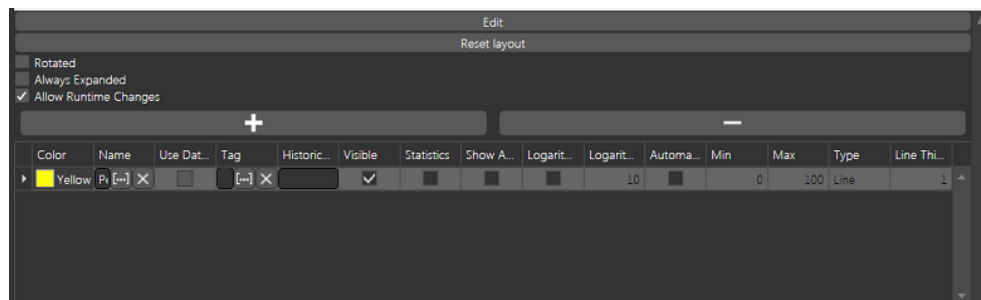
**Current Value Label Foreground:** This is used to set the labels' foreground color.

**Record Only if Good Quality:** when enabled, recording will only take place in the event of good communication quality when the tag in question is connected to the field by means of a Communication Driver.

**Shows Time Frame:** This is used to set the chart's sampling time range

**Record Every:** This is used to set the sampling time.

## Additional Properties (Smart Tags)



The **Edit** button enables you to preview how the object will look during runtime.

When you press the "+" button, a pen relating to the curve to be plotted on the chart will be added. Once added it can then be set according to the columns proposed from left to right:

**Color:** Indicates the pen's color.

**Name:** Used to insert a name for the pen to be displayed with in the legend. An String ID can also be inserted to manage language change during runtime.

**Use DataLogger:** When this box is disabled, it will be possible to select a tag of one of the pens associated with a Historian. When enabled, it will be possible to select a tag of one of the pens associated to a DataLogger. This is done by selecting a DataLogger column.

**Variable:** This field is used to select the tag to associate to the pen (or the DataLogger column in cases in which the tag is associated to a DataLogger and not a Historian).

**Historian/DataLogger:** Shows the name of the Historian or DataLogger to which the pen's tag is associated.

**Use Source Time Stamp:** When enabling this, the chart of the pen will be traced using the Tag's SourceTimeStamp column and not the RecordDateTime column of the record's recording. In cases where the pen is connected to a DataLogger, you must enable the SourceTimeStamp column for its Tag. The SourceTimeStamp column is always recorded with the UTC time for both the Historian and the Dataloggers.

**Localize Source Time Stamp:** This can only be enabled when the "Use Source Time Stamp" had been checked. When both boxes are checked, the pen's chart will be traced using the Source Time Stamp column but by applying local time.

**Use Aggregated Tables:** When the Data logger "Use Aggregated Tables" property is activated, the aggregated tables will be displayed as the data analysis's data source.

**Show Min Pen:** Shows the aggregated table's minimum value data curve.

**Show Avg Pen:** Shows the aggregated table's average value data curve.

**Show Max Pen:** Shows the aggregated table's max value data curve.

**Visible:** This box is used to make the pen visible or invisible (visible for default).

**Statistics:** This box is used to display the statistical curves relating to the pen. In addition to the pen's behaviour curves, another two curves will be displayed showing the minimum and maximum values obtained by the tag.

**Show Axis:** When enabling this box, the pen's custom scale will display on the right of the chart showing the pen's minimum and maximum values.

**Add Virtual Points:** When set to True, it allows the displayed curve to be extended by adding virtual samples. This is handy when needing to display curves associated to variables sampled on variations. The use of this property is recommended in combination with the 'Step Line' pen type.

**Logarithmic Scale:** This is used to enable the Logarithmic Scale.

**Logarithmic Base:** This is used to define which logarithmic base to apply to the logarithmic scale (if selected).

**Automatic Scale:** Enabling this box will display the pen's scale, if enabled, in automatic mode and not with the fixed Minimum and Maximum values.

**Min, Max:** these two values define the pen's minimum and maximum display limits. In cases where the Tag, associated to the pen, has been associated with an Engineering Unit, the Minimum and Maximum values will be taken directly from the Engineering Unit and these two fields will be ignored.

**Type:** This field is used to select the style with which to display the pen.

**Line Thickness:** This field is used to set the pen's line thickness.

## Runtime Legend Fields

**Visible:** This option box allows you to set the pen's visibility of not (set to visible for default).

**Statistics:** This box permits you to display the statistic curves relating to the pen. In addition to the pen's behaviour curve, another two curves will be displayed showing the maximum and minimum values obtained by the variable.

**Add Virtual Points:** When set to True, it allows the displayed curve to be extended by adding virtual samples. This is handy when needing to display curves associated to variables sampled on variations. The use of this property is recommended in combination with the 'Step Line' pen type.

**Color:** Indicates the pen's color.

**Tag:** This field is used to select the variable to associated to the pen (or the DataLogger column in cases when the variable is associated to a DataLogger and not a Historian).

**Engineering Unit:** Enables the use of engineering units.

**Historian/DataLogger:** Shows the name of the Historian or DataLogger to which the pen's variable is associated.



**Name:** This field is used to enter the name with which the pen will be shown in the legend. It is also possible to enter a String ID to manage language changes in runtime.

**Total Points:** Indicates the maximum number of points to be loaded in the object's buffer.

**Total Compressed Points:** Indicates the number of compressed points when the 'Enable data aggregation' property has been activated.

**Compression Ratio:** Indicates the data compression ratio when the 'Enable data aggregation' property has been activated (e.g. with a compression ratio of 10, one record out of every 10 records on the DB will be displayed).

**Min/Max:** These two values define the minimum and maximum pen display limits. In cases when the Tag associated to the pen has an Engineering Unit, the Minimum and Maximum values will be taken directly from the Engineering Unit and these two fields will be ignored.

**Minimum/Maximum statistic value :** Indicates the minimum or maximum statistic value for data loaded in the object's buffer.

**Average:** Average value of data loaded in the object's buffer.

**Median:** Indicates the distribution total's middle value of data loaded in the object's buffer.

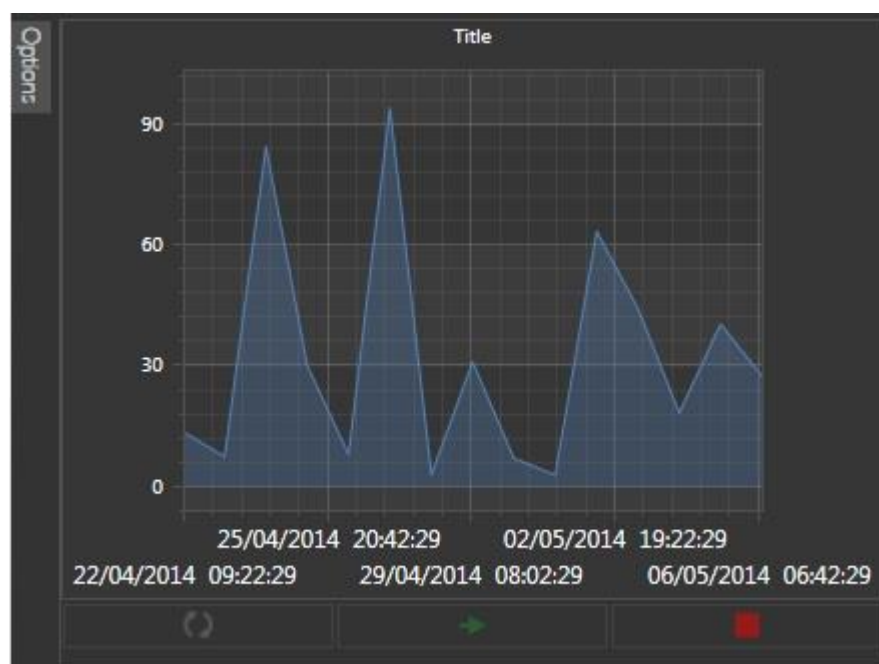
**Variance:** Indicates the variance value on the total of data loaded in the object's buffer.

**Standard Deviation:** Indicates the Standard Deviation value on the total of data loaded in the object's buffer.

## 5.52. Historical Trend

This object is used exclusively for representing historical data (in pause mode) and enables users to set filters for displaying data within the desired time ranges.

It is therefore essential that the tag associated to the object be linked to a historical resource. This object receives data directly from the Server and does not have direct access to the Data Base unlike the Data Analysis and Real Time Trends objects.



This object provides the user with the Fetch, Next and Stop buttons when functioning in runtime mode. These buttons allow the user to browse the data ranges displayed within the graph. The amount of data to be displayed can be set in the "Enable Max Return Values" property found in the toolbar Option on the left. Take care when doing this as the oldest data is always loaded according to the defined time range during the data loading phase. When the number of data samples for the time range exceeds the set limit you can use the Next buttons to scroll back and forth. For example, if you set a limit of 25 samples and a time range = 1 hour, but a 100 samples are recorded within the hour instead, the Trend will display the first 25. However, when you use the Next key, the next 25 samples will be loaded and so forth until the last available recorded samples have been reached.



As a difference to the RealTime Trend, this object also has an 'Options' toolbar that can be used for setting data selections and filters.



The Historical Trend object can only display one pen only which can also be configured by dragging and dropping the desired variable in the object.

## Historical Server Trend Properties

### Trend Style

**View Mode:** changes the data display mode (chart, grid).

**Touch Keyboard:** Enables object interactivity using the touch screen system.

**Graphic Command Button:** Replaces button texts with intuitive icons.

**Show Option Tab in Runtime:** Enables/disables the runtime option bar.

**Auto-Hide Option Tab:** Enables/disables the auto-hide of the object's ToolBar.

**Always Expanded:** Keeps the area within the area maximized hiding the other areas such as the ToolBar.

**Collapsed Height:** Sets the object's collapsed height when reduced from the toolbar.

**Collapsed Width:** Sets the object's collapsed width when reduced from the toolbar.

**Trend Title Font Style:** Used to set the trend title's font type.

**Trend Title Font Color:** Used to set the trend title's font color.

**Button Font Settings:** Sets the object's main font.

**Button Font Color:** Sets the object's font color.

**Toolbar Background:** Allow you to set the toolbar Background color.

**Toolbar Foreground:** Allow you to set the toolbar foreground.

**Allow runtime change:** Allow the object to be edited during runtime using a popup window.

### Trend Plot Style

**Show Labels:** Shows a label for each curve point on the chart.

**Show Plot Markers:** Shows a marker for each curve point on the chart.

**Plot Marker Size:** Sets the size of the markers on the chart.

**Plot Transparency:** Sets the transparency of the curve area on the chart.

**Plot Background Color:** Sets the background color for the Plot area.

**Show Crosshair Line:** Shows crosshair lines intersecting a value when sampled (graph mode).

**Shows Crosshair Argument Labels:** Shows argument label on with values on cross hair intersection.

**Show Crosshair Value Label:** Refers to the "Crosshair Line" and shows a label with intersecting values.

#### **Trend XY Scale Style**

**Automatic Scale:** Enables/disables the automatic scale of the pen in the chart area.

**Min. Scale Value:** Sets the minimum value for the pen scale

**Max. Scale Value:** Sets the maximum value for the pen scale

**Y-Axis Font Settings:** Set the font type used for the X axis (chart mode).

**X-Axis Font Settings:** Sets the font type used for the Y axis (chart mode).

**Y-Axis Prefix Text:** Used to set insert a specific field before the value of the axis (letter, sign, formulae, etc.).

**Show Y Major Grid line:** Shows the Y axis major grid lines.

**Show Y Minor Grid line:** Shows the Y axis minor grid lines.

**Show X Major Grid line:** Shows the X axis major grid lines.

**Show X Minor Grid line:** Shows the X axis minor grid lines.

#### **Trend Data Time Options**

**Fetch Data On Screen Load:** Forces the screen to display upon completion of loading data in the object.

**Use Start Time:** This is used to enable the setting for start time/date.

**Start Time:** This is used to set a start time/date.

**Use End Time: This is used to enable** Used for enabling the setting for end display time/date.

**End Time:** Used for setting a end display time/date.

**Read Type Mode:** Specifies read mode of data from the database.

**Aggregation Type:** Specifies the aggregation type of data from the database when not all of the data can be displayed in the Trend.

**Max. History Count:** Sets the maximum number of recorded samples.

#### **Style**

**Border Brush:** This is used to set the container's border colour.

**Border Thickness:** This is used to set the container's border thickness.

**Back color:** This is used to set the container's background colour.

**Stroke Color:** This is used to set the color of outlines used in the object.

**Disable Anti-aliasing:** Enables/disables the Anti-aliasing pixel correction function.

**Margin:** Used to set object's margins

#### **Fonts**

**Font Family:** Sets font model to be used.

**Font Size:** Sets the font's size within the object.

**Font Style:** Sets the font's style.

**Font Weight:** Sets the thickness or thinness of how the font should be displayed ( *light*, *extralight*, *normal* etc...).

## Layout

See topic on: "Common ToolBox object and drawing properties"

## Behaviour

**Editable in runtime:** sets the viewer to edit mode at runtime. This option also allows the configuration of the created object to be saved and recalled.

See topic on: "Drawing and Object Behaviour Properties".

## General

**MaxLabelNumber:** This parameter is used to define the maximum number of points. When this number is exceeded, the labels will no longer be shown. For example, if the labels have been enabled for displaying but the number of samples exceeds the one defined in this parameter, they will automatically be disabled for displaying. It would be worth considering that the graphical interface performance slows down as the number of labels to displayed increases.

It is therefore advised not to set this property higher than the one set for default to avoid compromising interface performances or to cause it to freeze altogether while loading data.

see topic on: "Common ToolBox Object and Drawing Properties".

## Visibility

**Transparency:** Sets the object's Transparency level.

**Visibility:** enables the object's visibility.

**Visible on Client:** makes object visible on the webclient

**Visibility Level:** permits you to set the mask to determine the object's visibility.

**Zoom Level Visibility X:** permits you to set the Zoom factor on the x axis after which the object will become visible on screen.

**Zoom Level Visibility Y:** permits you to set the Zoom factor on the y axis after which the object will become visible on screen.

## Tag

See topic on: "Common ToolBox object and drawing properties"

## User Access

See topic on: "Draws and Object User Management" concerning objects and drawings that define the security parameters based on the project's User and Password management.

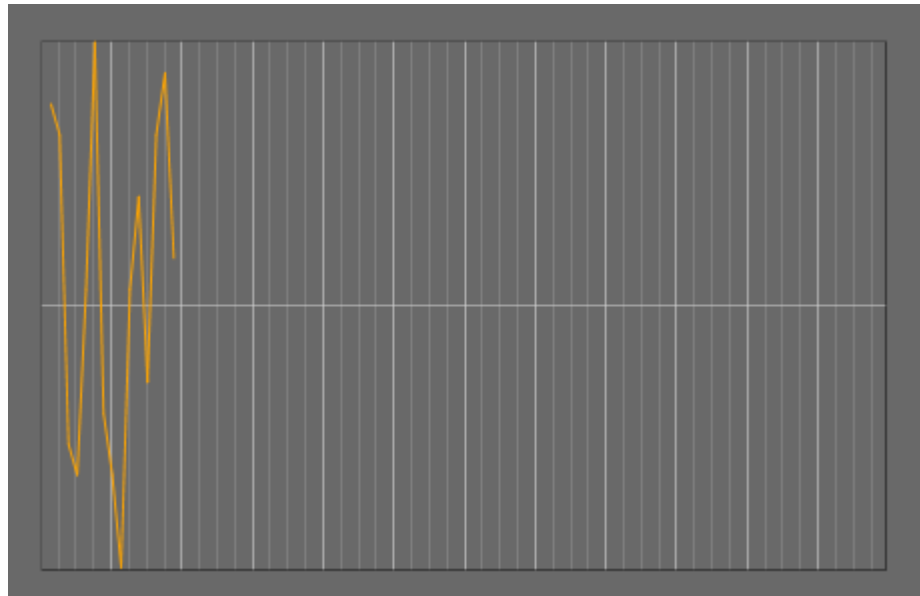
## Expressions

See topic on: "Managing Expression" concerning objects and drawings.

### Preserve Template Changes

**Preserve:** see paragraph "Preserve local Draws and Objects properties" concerning original library objects.

## 5.53. Sparkline Chart



The "Sparkline Chart" object is a graphical chart that displays the values of the current day only.

The DB table from which the data is retrieved must have at least one value column (see "value column name" property), and a date/time column (see "time column name" property)

### SparklineChart properties

#### Style

**Back Color:** This is used to set the container's back colour.

**Stroke Color:** This is used to set the numbers in the object's foreground.

**Border Brush:** This is used to set the container's border colour.

**Margin:** This is used to set the object's margins.

**Disable Anti-aliasing:** Enables/disables the Anti-aliasing pixel correction function.

**Border Thickness:** This is used to set the container's border thickness.

#### Visibility

**Visible on Client:** This is used to make object visible on the webclient

**Visibility Level:** This is used to set the mask to determine the object's visibility.

**Zoom Level Visibility X:** This is used to set the Zoom factor on the x axis after which the object will become visible on screen.

**Zoom Level Visibility Y:** This is used to set the Zoom factor on the y axis after which the object will become visible on screen.

**Visibility:** This is used to enable the object's visibility.

**Transparency:** This is used to set the object's Transparency level.

## **Behaviour**

See topic on: "Drawing and Object Behaviour Properties".

## **Layout**

See topic on: "General Draws and Objects Properties".

## **Tag**

See topic on: "General Draws and Objects Properties".

## **General**

See topic on: "General Draws and Objects Properties".

## **Sparkline DB connection**

**Load Data Every:** Data refresh interval.

**Connection String:** DB connection string.

**Table Name:** Table from where to retrieve data

**Value column name:** Name of the column containing the values to display on chart.

**Time column name:** Name of time axis column

**Max. record num. :** The maximum number of records to be displayed ( default 96, which indicates the number of quarter hours in a day) .

**Where condition:** Additional filter options for selecting the source of data to be displayed in cases when there are many.

## **User Access**

See topic on: "Draws and Objects User Acces Properties" for objects and drawings that define the security parameters according to the project's User and Password management.

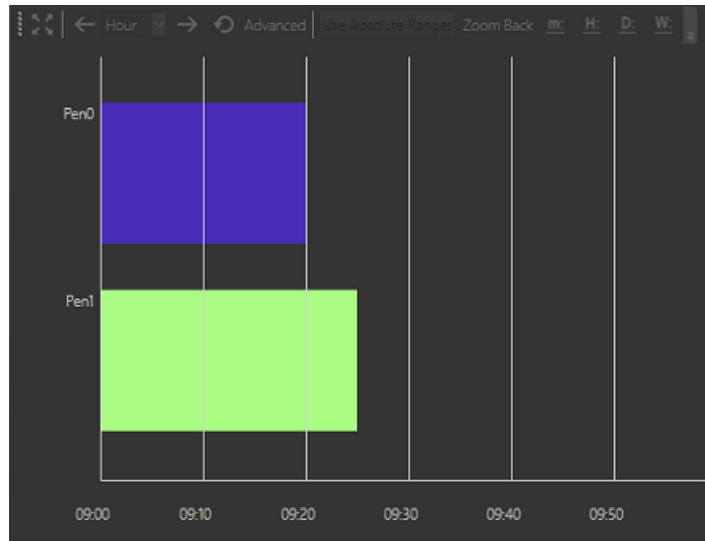
## **Expressions**

See topic on: "Managing Expressions" of objects and drawings.

## **Preserve Template Changes**

See topic on: "Preserve local Draws and Objects Properties" in respect to the original object in the library.

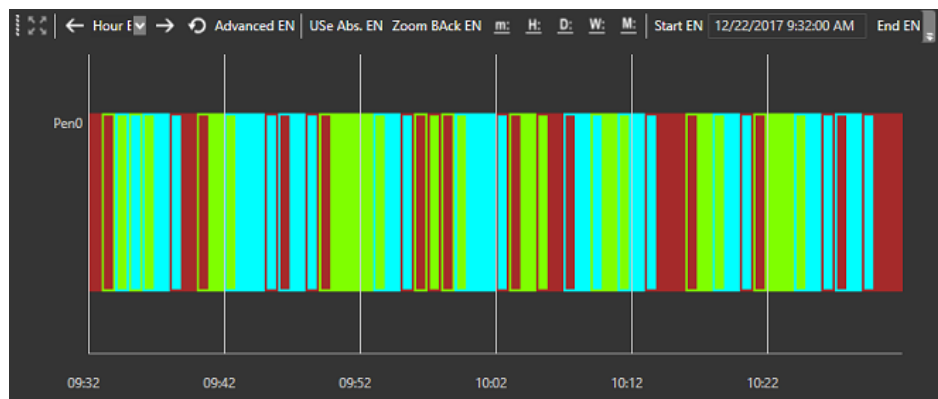
## 5.54. States Chart



The States Chart object is used for representing states (values) of variables associated to a DataLogger or an Historian in order to graphically view, for example, the operational state of a machine.

You can also define threshold colors which the pen will assume according to the variable value associated to a specific time. If the set base times have multiple values, the dominating value will be represented as a back color while the one immediately after (see image below), will be represented in the border color for duration (clicking on the cell which represents inhomogeneous values, the time interval for the relative values will display).

However, you can define as many pens as desired to represent several states at the same time. The chart can also be scrolled to show the historical part as well. .



### States Chart Properties

#### Tag

See topic on: "Common ToolBox object and drawing properties".

#### Style

**Back Color:** This is used to set the container's back colour.

**Stroke Color:** This is used to set the numbers in the object's foreground.

**Border Brush:** This is used to set the container's border colour.

**Margin:** This is used to set the object's margins.

**Disable Anti-aliasing:** This is used to enable/disable the Anti-aliasing pixel correction function.

**Border Thickness:** This is used to enable the container's border thickness.

## Layout

See topic on: "Common ToolBox object and drawing properties".

## User Access

See topic on: "Draws and Object User Management" for objects and drawings that define the security parameters according to the project's Users and Password management.

## Behaviour

See topic on: "Drawing and Object Behaviour Properties".

## Fonts

**Font size:** This is used to set font side within object.

**Font Family:** This is used to set the font type to be used.

**Font Style:** This is used to set the font style desired.

**Font Weight:** This is used to set the font's characteristics ( *light*, *extralight*, *normal* ecc...).

## Visibility

**Visibility Layer Level:** This is used to set the mask to determine the object's visibility.

**Visible on Web Client:** This is used to make object visible on the webclient

**Zoom Level Visibility X:** This is used to set the Zoom factor on the x axis after which the object will become visible on screen.

**Zoom Level Visibility Y:** This is used to set the Zoom factor on the y axis after which the object will become visible on screen.

**Transparency:** This is used to set the object's Transparency level.

**Visibility:** This is used to enable the object's visibility.

## General

See topic on: "Common ToolBox object and drawing properties"

## Preserve Template Changes

See topic on: "Preserv Local Draws and Object Properties" in respect to the original object in the library.

## Expressions

See topic on: "Managing Expressions" for objects and drawings.

## State Chart Style

**Pen Height:** This is used to set the pens' height in pixels.

**Separator Height:** This is used to set the separator height in pixels.

**Auto Hide Toolbar:** This is used to enable the auto hide toolbar function.



**Use Absolute Ranges:** Permits data to be shown between the relative/absolute range.

**Show Row Labels:** This is used to show/hide the labels.

**Show Tooltips:** Allows tooltips for data displayed in the object to be hidden when hovering mouse on the pen. Therefore, the value assumed by the variable will not be shown at that moment.

**Label Foreground Brush:** This is used to set the label and time line colors in the chart.

**Legend Area Visible:** Shows/Hides the legends.

**Toolbar Background:** This is used to set the ToolBar color.

**Auto refresh:** This is used to enable the automatic pen value refresh.

### States Chart Options

**Filter Type:** This is used to set a temporal filter to display data with.

**Num. Samples:** Determines the maximum number of samples to be collected from the DB the moment the pen is displayed.

## Additional Properties (Smart Tags)

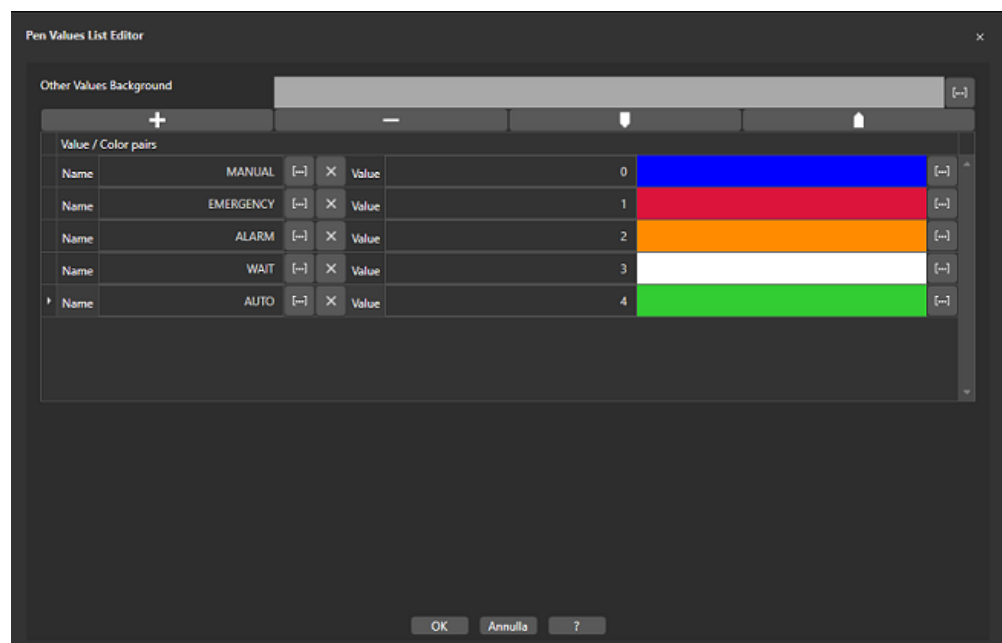
**Pen Nome:** The name to display the pen in the legend with is entered here. It is also possible to insert an ID String to manage language changes at runtime.

**Tag Name:** This field is used to select a variable to associate to the pen (or DataLogger column in cases where the variable is associated to a DataLogger and not a Historian).

**Use DataLogger:** When this is disabled, it will be possible to select the Pen's Tag among those associated to the Historian. Conversely, when enabled it will be possible to select the pen's Tag from those associated to a DataLogger and therefore a DataLogger Column.

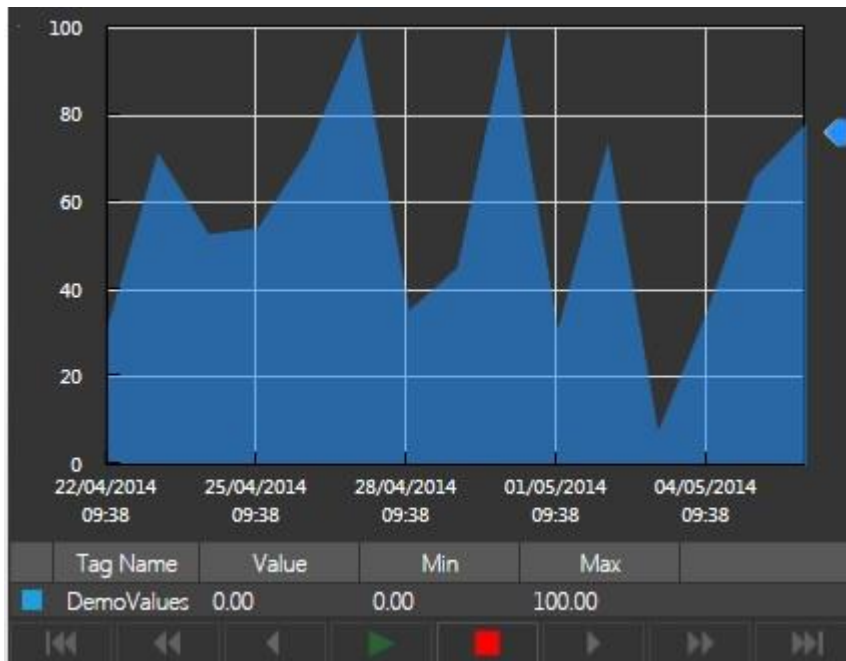
**DataLogger/Historical:** shows the name of the DataLogger to which the pen's variable has been associated.

**Value/Color Pairs:** This is used to set the color to be assigned to the pen based on its value. In cases where the value assumed by the pen has not been defined, the colour used in the 'other value background' property will be used (see screenshot below).



## 5.55. RealTimeTrend

As a difference to the other objects of this type, this object displays the variables linked to the chart's pens in real-time. Therefore, pens that relate to variables not necessarily connected to a historical resource can be configured.



When in pause mode, the realtime trend will stop displaying the values in realtime in order to perform an analysis on the currently displayed data.



When putting the Trend object in stop mode, pens can be selected from the legend by clicking on the corresponding line of the pen desired. This will highlight the corresponding curve on the chart. To deselect the pen, simply click on the pen's line again while pressing down the CTRL key.

The Realtime Trend object provides various navigation buttons for the user to use in runtime to display data. The user can expand the range of displayed data on screen by using the scroll bar on the X and Y axis (eg. the time interval presented on the Trend's X axis can be modified).

### RealTime Trend Plotting Properties

#### Trend Plot Style

**Point Font Settings:** This is used to set the font type for the sample point.

**Plot Area Background Color:** This is used to set the Plot area's background colour.

**Pen Name:** This is used to set a name for the pen.

**Legend Area Text Color:** This is used to change the color of the legend area.

**Legend Area Visible:** Displays/hides the legend area within the object.

**Pen Area Background Color:** This is used to set the background colour of the pen area.

**Pen Area Visible:** This is used to display/hide the pens in the object.

**Show Tag Name:** This is used to display/hide the variable name in the label placed under the cursor.

**Show Pen Name:** This is used to display/hide the pen's name in the label placed under the cursor.

### Trend Style

**Runtime Mode:** This is used to set the Runtime mode type.

**Title:** This is used to set a title for the object.

**Title Font Settings:** This is used to set the Title's font type to be used.

**Title Font Color:** This is used to set the title's font color to be used.

**Trend Background Color:** This is used to set the Trend area's background colour.

**Toolbar Background:** This is used to set the toolbar Background color.

**Toolbar Foreground:** This is used to set the toolbar foreground.

**Allow runtime change:** Allows object to be edited during runtime using a popup window.

**AutoHide Toolbar:** Allows you to enable/disable the autohide funct. for the object's top toolbar

**Always Expanded:** Allows you to enable/disable the expanded view for the object.

### Trend Data time Options

**Num. Samples:** This is used to set the maximum number of samples.

**Record Every:** This is used to set the recording time to be used for updating the chart curve.

**View Time Frame:** This is used to set the time frame to be displayed in the chart.

**Time Format:** This is used to set the Time Format. For further information concerning custom format sting types please refer to the topic on Strings.

**Date Format:** This is used to set the Date Format. For further information concerning custom format sting types please refer to the topic on Strings.

**Connection String:** This is used to set the connection string to the database (if not defined the realtime variable value will be used).

**Record Only when quality good:** Allows you to enable/disable the possibility to record the value only on quality good of the tag.

### Trend XY Scale Style

**Y-Axe Font Settings:** This is used to set the font type used for the X axis (chart mode).

**X-Axe Font Settings:** This is used to set the character type to be used for the Y axis (chart mode).

**Axis Label Text Color:** This is used to set the Axis Labels' text color.

**Axis Stroke Color:** This is used to set the Axis stroke color.

**Axis Stroke Thickness:** This is used to set the thickness dimension for the trend's axis stroke

**Plot Border Brush:** This is used to set the brush color for the plotting area's border.

**Plot Border Thickness:** This is used to set the thickness dimension for the border trend area.

**Grid Color:** This is used to set the grid color.

**Y Min. Grid Line Number:** This is used to set the number of lines in the Grid on the Y axis.

**Y Major Div. Number:** This is used to set the number of major division lines on the Plot area's Y axis. In cases where the 'Automatic Scale' property has been enabled, the value set here will be considered as the number of division intervals to keep to. However, it is not guaranteed that this value will be respected when automatic scaling is performed.

**X Major Div. Number:** This is used to set the number of major division lines on the Plot area's X axis. In cases in which the Automatic Scale property had been enabled, the value set here will be considered as the number of division intervals to keep to. However, it is not guaranteed that this will be respected when automatic scaling is performed.

**Show X Maj.Grid Line:** Displays the X axis major grid lines.

**Show Y Maj.Grid Line:** Displays the Y axis major grid lines.

**Automatic Scale:** When this is enabled, the automatic scale will be used for the pens in the chart.

**Use Engineering Unit Values:** Uses the Engineering Unit associated to the pen's variable to inherit the Y axis.

**General Automatic Scale:** This option enables the scale management at general level. The scale will be the same one for all pens and will assume the minimum value of the different pens' minimums and the maximum value of the different pens' maximums. Each pen will then be traced according to the general minimal and maximum values.

## Style

**Border Brush:** This is used to set the container's border color.

**Border Thickness:** This is used to set the container's border thickness.

**Foreground Color:** This is used to set the color of the letter and number characters used in the object.

**Background color:** This is used to set the container's background color.

**Font Size:** This is used to set the font size used within the object.

**Disable Anti-aliasing:** Enables/disables the Anti-aliasing pixel correction function.

**Margin:** This is used to set the object's margins.

**Transparency:** This is used to set the object's Transparency level.

## Fonts

**Font Family:** Sets font model to be used.

**Font Size:** Sets the font's size within the object.

**Font Style:** Sets the font's style.

**Font Weight:** Sets the thickness or thinness of how the font should be displayed ( *light*, *extralight*, *normal* etc...).

## Layout

See topic on: "Common ToolBox object and drawing properties"

## Behaviour

**Editable in runtime:** sets the viewer to edit mode at runtime. This option also allows the configuration of the created object to be saved and recalled.

See topic on: "Drawing and Object Behaviour Properties".

## General

**Point Precision:** This property is used to establish the number of decimal figures to display in the tooltip that shows the value of the points when the Trend is stopped and the cursor is moved along the curve.

See topic on: "Common ToolBox Object and Drawing Properties".

## Visibility

**Visibility:** enables the object's visibility.

**Visible on Client:** makes object visible on the webclient

**Visibility Level:** permits you to set the mask to determine the object's visibility.

**Zoom Level Visibility X:** permits you to set the Zoom factor on the x axis after which the object will become visible on screen.

**Zoom Level Visibility Y:** permits you to set the Zoom factor on the y axis after which the object will become visible on screen.

## Tag

See topic on: "Common ToolBox object and drawing properties"

## User Management

See topic on: "Draws and Objects User Management" concerning objects and drawings that define the security parameters based on the project's User and Password management.

## Expressions

See topic on: "Managing Expressions" concerning objects and drawings.

## Preserve Template Changes

Preserve: see topic on "preserv local draws and objects properties" in respect to the original library object.

## Additional Properties (Smart Tags)

**Visible:** this box is used to make pen visible or invisible (visible for default).

**Pen Color:** indicates the pen's color.

**Pen Name:** this field is used to insert a name for the pen to be displayed with in the legend. An String ID can also be inserted to manage language change during runtime.

**Tag Name:** this field is used to select the tag to associated to the pen (or DataLogger column in cases where the tag is associated to a DataLogger and not a Historian).

**Array Index:** If the set tag is an Array type this permits each individual element of the Array to be pointed.

**Thickness:** this field is used to set the pen's line thickness.

**Pen Style:** this field is used to select the style with which to display the pen.

**Min/Max:** these two values define the pen's minimum and maximum display limits. In cases where the Tag, associated to the pen, has been associated with an Engineering Unit, the Minimum and Maximum values will be taken directly from the Engineering Unit and these two fields will be ignored.

## Runtime Legend Fields

**Visible:** this option box is used to set the pen to visible or invisible (set visible for default).

**Pen Color:** indicates the color of the pen.

**Pen Name:** the name with which the pen will be displayed in the legend is entered here. It is also possible to enter a String ID to manage language change in runtime.

**Engineering Unit:** enables the use of the engineering unit.

**Value:** the value obtained by the variable associated to the pen in real-time.

**Min/Max:** these two values define the pen's minimum and maximum limits. In cases in which the Tag associated to the pen has been associated with a Engineering Unit, the Minimum and Maximum values will be taken directly form the Engineering Unit and these two fields will be ignored.

**Minimum/Maximum Statistic Value:** indicates the minimum or maximum statistic value for data loaded in the object's buffer.

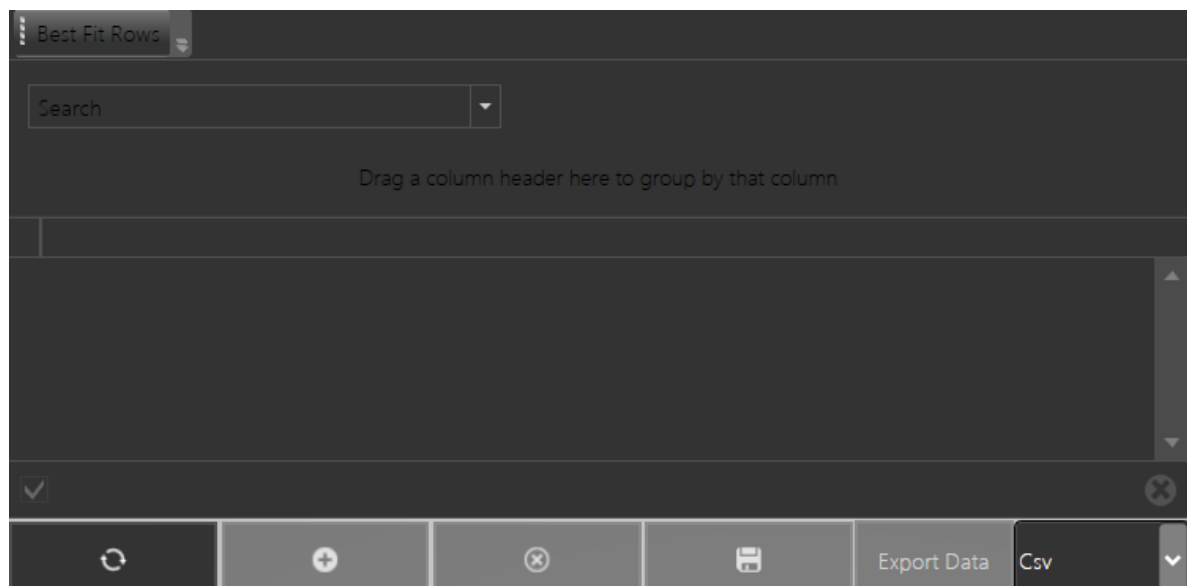
## 5.56. Viewers

## 5.57. DB Grid Control

The "DB connector" Grid object has been designed to create a connection towards a table created in a database that does not necessarily need to be managed by the Next resources.

The database connection is set with a query using the "DataSource" menu item from the smart properties which are accessed by clicking on the spanner symbol.

When in run mode, the grid will display the data from the table.



The viewer also has other smart properties, such as :**"Allow Add New", "Allow Remove", "Allow Edit"**.

These properties can be used to insert a new records by adding rows in the grid, delete rows and the relating records in the DB table and change the values of one or more grid cells.



The Grid object can also be used to display the result of a view.

## DB connector Grid Viewer Properties

### Grid Style

**Row Area Font Settings:** Sets font type for data area.

**Header Font Settings:** Sets font type for column headers.

**Toolbar Background:** Used to define the background color to apply to the Toolbar.

**Toolbar Foreground:** Used to define the foreground color to apply to the Toolbar.

**Graphic Button:** Allows the object's buttons to be displayed as icons instead of text.

**Allow Edit:** Allows the object to be edited in runtime.

**Allow Add New:** Enables the command used for adding new rows in the grid.

**Allow Remove:** Enables the command used for removing the selected rows from the grid.

**Max. Transactions:** Maximum number of transactions to be used for each record.

**Allow PrimaryKey Changing:** Allows you to modify the values relating to the Primary Key column.

**Show Auto Layout Button:** Enables the automatic column layout button placed on the Grid object's top bar.

**Show Command Buttons:** Shows/hides the command buttons.

**Shows Filter box:** Shows/hides the filter box.

**Shows search box:** Shows/hides search box.

**Shows Regroup Box:** Shows/hides the regroup box.

**HeaderBackground:** This is used to set the Grid's Header background color.

**HeaderForeground:** This is used to set the Grid's Header text color.

**RowAreaForeground:** This is used to set the Grid's row text color.

**FocusedRowBackground:** This is used to set the background color of the selected row.

**FocusedRowForeground:** This is used to set the color of the selected row's text.

**ShowCellsBorder:** This is used to show/hide the border of Grid's cells.

**CellBorderColor:** This is used to set the color of the Grid's cells' borders.

**FocusedCellBackground:** This is used to set the background color of the selected cell /cells.

**FocusedCellForeground:** This is used to set the text color of the selected cell/cells.

### Grid Advanced

**Editable:** Allows the object to be edited in Runtime.

## User Access

See topic on "Draws and Object User Management" concerning objects and drawings that define the security parameters based on the project's User and Password management.

## Style

**Border Brush:** This is used for setting the container's border colour.

**Border Thickness:** This is used for setting the container's border thickness.

**Back color:** This is used for setting the container's background colour.

**Stroke Color:** This is used for setting the colour of the letter and number characters used in the object.

**Disable Anti-aliasing:** Enables/disables the Anti-aliasing pixel correction function.

**Margin:** sets the object's margins

## Fonts

**Font Family:** Sets font model to be used.

**Font Size:** Sets the font's size within the object.

**Font Style:** Sets the font's style.

**Font Weight:** Sets the thickness or thinness of how the font should be displayed ( *light*, *extralight*, *normal* etc...).

## Layout

See paragraph on "Common ToolBox object and drawing properties".

## Behaviour

See topic: "Drawing and Object Behaviour Properties".

## General

see topic "Common ToolBox Object and Drawing Properties".

## Visibility

**Transparency:** Sets the object's Transparency level.

**Visibility:** enables the object's visibility.

**Visible on Client:** makes object visible on the webclient

**Visibility Level:** permits you to set the mask to determine the object's visibility.

**Zoom Level Visibility X:** permits you to set the Zoom factor on the x axis after which the object will become visible on screen.

**Zoom Level Visibility Y:** permits you to set the Zoom factor on the y axis after which the object will become visible on screen.

## Tag

See paragraph on "Common ToolBox object and drawing properties".

## Expressions

See paragraph "Managing Expression" concerning objects and drawings.



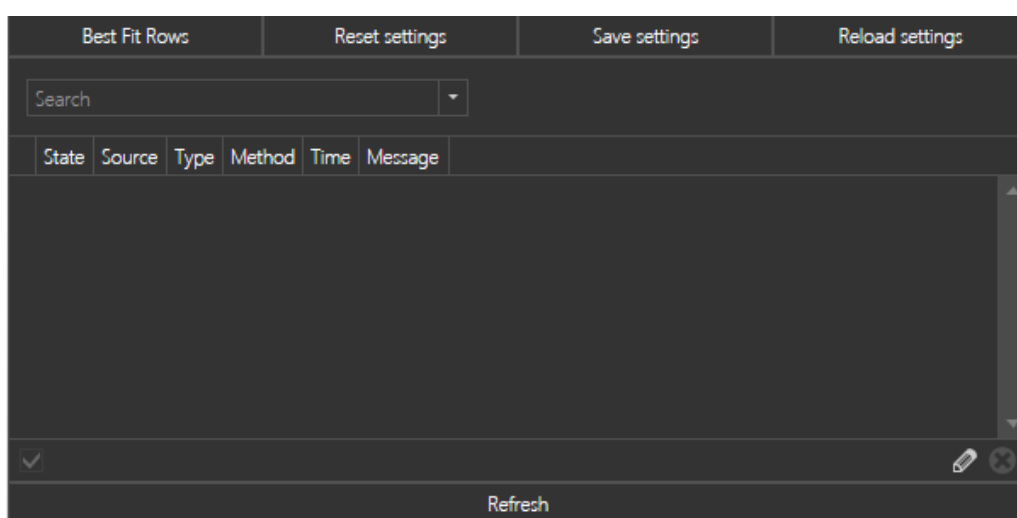
### Preserve Template Changes

**Preserve:** see paragraph "Preserve local Draws and Objects properties" concerning original library objects.

## 5.58. Auditing Trail Validator

The "Auditing Viewer" window is an object used for displaying events that have been notified by the connected OPC server.

The reported event types are tightly linked to the Server's programming.



For example, the AUDIT window may also contain notifications as a consequence of alarm ACK or RESET events.

## Auditing Viewer Properties

### Auditing window Advanced

**Toolbar Background:** Used to define the background color to apply to the Toolbar.

**Toolbar Foreground:** Used to define the foreground color to apply to the Toolbar.

### Behaviour

**Editable in runtime:** enables the viewer object with edit mode so that it can be edited at Runtime. This option also allows you to save and call the created object's configuration.

See topic: "Objects Executions Properties".

### User Access

See topic: "Draws and Object User Management" concerning objects and drawings that define the security parameters based on the project's User and Password management.

## Audit window Style

**Alarm Area font Settings:** Allow you to set the alarm text font.

**Allow Column to be dropped:** Allow to drop column in runtime.

**Header Font Settings:** Allow you to set the column header font.

## Style

**Border Brush:** This is used for setting the container's border colour.

**Border Thickness:** This is used for setting the container's border thickness.

**Foreground Color:** This is used for setting the colour of the letter and number characters used in the object.

**Background color:** This is used for setting the container's background colour.

**Font Size:** This is used for setting the font size used within the object.

**Disable Anti-aliasing:** Enables/disables the Anti-aliasing pixel correction function.

**Margin:** sets object's margins

## Fonts

**Font Family:** Sets font model to be used.

**Font Size:** Sets the font's size within the object.

**Font Style:** Sets the font's style.

**Font Weight:** Sets the thickness or thinness of how the font should be displayed (*light, extralight, normal etc...*).

## Layout

See paragraph on "Common ToolBox object and drawing properties".

## General

see the topic "Common ToolBox Object and Drawing Properties".

## Visibility

**Transparency:** Sets the object's Transparency level.

**Visibility:** enables the object's visibility.

**Visible on Client:** makes object visible on the webclient

**Visibility Level:** permits you to set the mask to determine the object's visibility.

**Zoom Level Visibility X:** permits you to set the Zoom factor on the x axis after which the object will become visible on screen.

**Zoom Level Visibility Y:** permits you to set the Zoom factor on the y axis after which the object will become visible on screen.

## Tag

See paragraph on "Common ToolBox object and drawing properties".

## User Management

See topic on "User Manager Properties" concerning objects and drawings that define the security parameters based on the project's User and Password management.

### Expressions

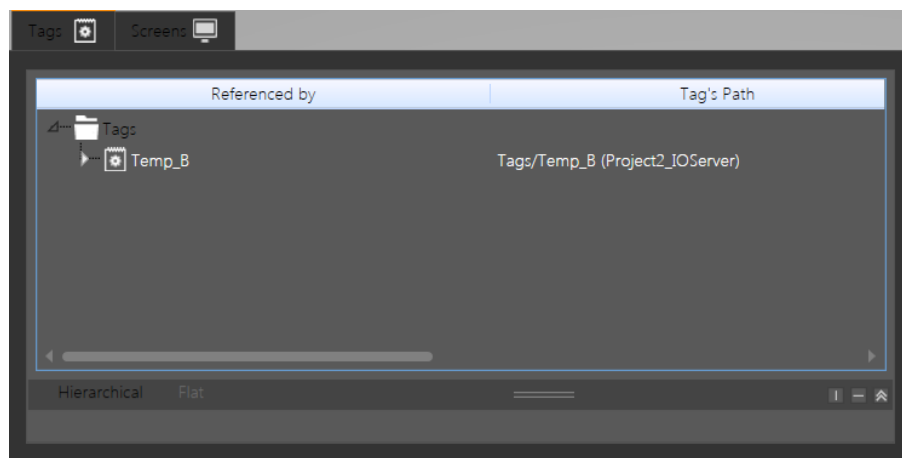
See the "Expressions in objects" topic concerning objects and drawings.

### Preserve Template Changes

See paragraph "Preserve local Draws and Objects properties" concerning original library objects

## 5.59. Cross Reference Viewer

For further information on this object please see the topic relating to "Cross\_Reference"



### Cross Reference Viewer Properties

#### Style

**Border Brush:** This is used for setting the container's border colour.

**Border Thickness:** This is used for setting the container's border thickness.

**Back color:** This is used for setting the container's background colour.

**Stroke Color:** This is used for setting the colour of the letter and number characters used in the object.

**Disable Anti-aliasing:** Enables/disables the Anti-aliasing pixel correction function.

**Margin:** sets the object's margins

#### Fonts

**Font Family:** Sets font model to be used.

**Font Size:** Sets the font's size within the object.

**Font Style:** Sets the font's style.

**Font Weight:** Sets the thickness or thinness of how the font should be displayed ( *light*, *extralight*, *normal* etc...).

#### Layout

See paragraph on "Common ToolBox object and drawing properties"

**Behaviour**

See topic: "Drawing and Object Behaviour Properties".

**General**

see topic "Common ToolBox Object and Drawing Properties".

**Visibility**

**Transparency:** Sets the object's Transparency level.

**Visibility:** enables the object's visibility.

**Visible On Web Client:** makes object visible on the webclient

**Visibility Layer Level:** Use this to set the mask to determine the object's visibility.

**Zoom Level Visibility X:** Use this to set the Zoom factor on the x axis after which the object will become visible on screen.

**Zoom Level Visibility Y:** Use this to set the Zoom factor on the y axis after which the object will become visible on screen.

**Tag**

See paragraph on "Common ToolBox object and drawing properties"

**User Access**

See topic on "Draws and Object User Management" concerning objects and drawings that define the security parameters based on the project's User and Password management.

**Expressions**

See paragraph "Managing Expression" concerning objects and drawings.

**Preserve Template Changes**

**Preserve:** see paragraph "Preserve local Draws and Objects properties" concerning original library objects.

**Miscellaneous**

Permits you to modify the object container properties.

## 5.60. PDF Viewer

The "PDF Viewer" enables associated PDF documents to be viewed in runtime. Note that in this case the pages will be bigger than the screen window layout and, therefore, side scrollbars will be provided to make navigating the various pages easier for the user.



Example of PDF viewer with the Command Bar disabled

## PDF Viewer Properties

### File Settings

**FilePath:** This is used to set the path of the PDF file that will open in runtime.

**Command Bar Style:** This is used to set the style command bar.

### Style

**Border Brush:** This is used to set the container's border colour.

**Border Thickness:** This is used to set the container's border thickness.

**Foreground Color:** This is used to set the colour of the letter and number characters used in the object.

**Background color:** This is used to set the container's background colour.

**Disable Anti-aliasing:** Enables/disables the Anti-aliasing pixel correction function.

**Margin:** This is used to set the object's margins

### Fonts

**Font Family:** This is used to set font model to be used.

**Font Size:** This is used to set the font's size within the object.

**Font Style:** This is used to set the font's style.

**Font Weight:** This is used to set the thickness or thinness of how the font should be displayed ( *light*, *extralight*, *normal* etc...).

### Layout

See topic on: "Common ToolBox object and drawing properties"

### Behaviour

See topic on: "Drawing and Object Behaviour Properties".

### General

see topic on: "Common ToolBox Object and Drawing Properties".

## Visibility

**Transparency:** This is used to set the object's Transparency level.

**Visibility:** This is used to enable the object's visibility.

**Visible on Client:** This is used to make the object visible on the webclient

**Visibility Level:** This is used to set the mask to determine the object's visibility.

**Zoom Level Visibility X:** This is used to set the Zoom factor on the x axis after which the object will become visible on screen.

**Zoom Level Visibility Y:** This is used to set the Zoom factor on the y axis after which the object will become visible on screen.

## Tag

See topic on: "Common ToolBox object and drawing properties"

## User Access

See topic on: "Draws and Object User Management" concerning objects and drawings that define the security parameters based on the project's User and Password management.

## Expressions

See topic on: "Managing Expression" concerning objects and drawings.

## Preserve Template Changes

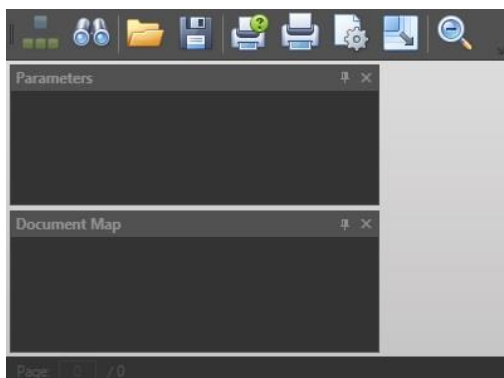
**Preserve:** see topic on: "Preserve local Draws and Objects properties" concerning original library objects.



The "PDF Viewer" management is not supported during Web Visualization. In cases where the object is deployed on the web client, the "Save", "Open", "Print" and "Close" functions will not function and the corresponding buttons will remain disabled.

## 5.61. Report Viewer

The 'Report Viewer' is used for displaying a selected report layout in a specific part of the screen. Scrollbars will be enabled to facilitate navigation when the report layout size is bigger than the Viewer object.



## Report Viewer Properties

### Report settings

**Report name:** Sets the name of the report to be linked to the viewer.

**Parameters:** Used for setting report parameters.

### Style

**Border Brush:** This is used for setting the container's border colour.

**Border Thickness:** This is used for setting the container's border thickness.

**Foreground Color:** This is used for setting the colour of the letter and number characters used in the object.

**Background color:** This is used for setting the container's background colour.

**Disable Anti-aliasing:** Enables/disables the Anti-aliasing pixel correction function.

**Margin:** sets the object's margins

### Fonts

**Font Family:** Sets font model to be used.

**Font Size:** Sets the font's size within the object.

**Font Style:** Sets the font's style.

**Font Weight:** Sets the thickness or thinness of how the font should be displayed ( *light, extralight, normal etc...*).

### Layout

See topic: "Common ToolBox object and drawing properties"

### Behaviour

See topic: "Drawing and Object Behaviour Properties".

### General

see topic "Common ToolBox Object and Drawing Properties".

### Visibility

**Transparency:** Sets the object's Transparency level.

**Visibility:** enables the object's visibility.

**Visible on Client:** makes object visible on the webclient

**Visibility Level:** permits you to set the mask to determine the object's visibility.

**Zoom Level Visibility X:** permits you to set the Zoom factor on the x axis after which the object will become visible on screen.

**Zoom Level Visibility Y:** permits you to set the Zoom factor on the y axis after which the object will become visible on screen.

### Tag

See topic: "Common ToolBox object and drawing properties"

### User Access

See topic: "Draws and Object User Management" concerning objects and drawings that define the security parameters based on the project's User and Password management.

### Expressions

See topic: "Managing Expression" concerning objects and drawings.

### Preserve Template Changes

**Preserve:** See topic: "Preserve local Draws and Objects properties" concerning original library objects.

## 5.62. Web Browser Viewer

The "Web Browser Viewer" is used to display or modify associated web pages in runtime. If the web page is larger than the screen layout, side scroll bars will appear to facilitate navigation for the user.



### Web Browser Viewer Properties

#### WEB Settings

**URL address:** This is used to set the URL address to be opened within the viewer.

#### Style

**Border Brush:** This is used to set the container's border colour.

**Border Thickness:** This is used to set the container's border thickness.

**Foreground Color:** This is used to set the colour of the letter and number characters used in the object.

**Background color:** This is used to set the container's background colour.



**Font Size:** This is used to set the font size used within the object.

**Disable Anti-aliasing:** Enables/disables the Anti-aliasing pixel correction function.

**Margin:** This is used to set the object's margins

## Fonts

**Font Family:** This is used to set the font model to be used.

**Font Size:** This is used to set the font's size within the object.

**Font Style:** This is used to set the font's style.

**Font Weight:** This is used to set the thickness or thinness of how the font should be displayed ( *light*, *extralight*, *normal* etc...).

## Layout

See topic on: "Common ToolBox object and drawing properties"

## Behaviour

See topic on: "Drawing and Object Behaviour Properties".

## General

see topic on: "Common ToolBox Object and Drawing Properties".

## Visibility

**Transparency:** This is used to set the object's Transparency level.

**Visibility:** This is used to enable the object's visibility.

**Visible on Client:** This is used to make the object visible on the webclient

**Visibility Level:** This is used to set the mask to determine the object's visibility.

**Zoom Level Visibility X:** This is used to set the Zoom factor on the x axis after which the object will become visible on screen.

**Zoom Level Visibility Y:** This is used to set the Zoom factor on the y axis after which the object will become visible on screen.

## Tag

See topic on: "Common ToolBox object and drawing properties"

## User Access

See topic on: "Draws and Object User Management" concerning objects and drawings that define the security parameters based on the project's User and Password management.

## Expressions

See topic on: "Managing Expression" concerning objects and drawings.

## Preserve Template Changes

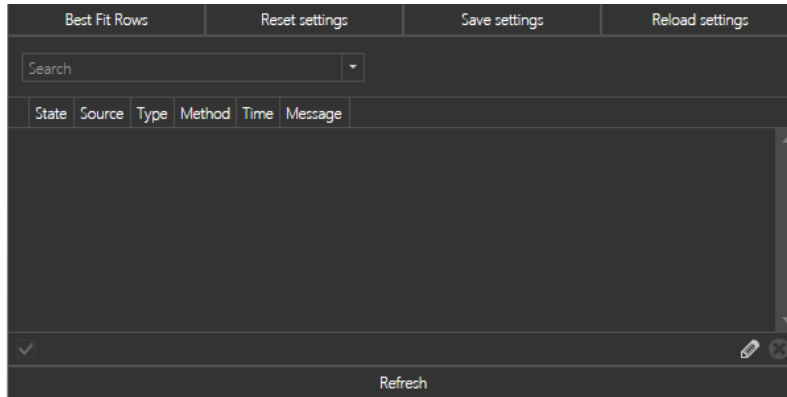
**Preserve:** see paragraph "Preserve local Draws and Objects properties" concerning original library objects.

## 5.63. Historical Data Viewers

## 5.64. Auditing Trail Validator

The "Auditing Viewer" window is an object used for displaying events that have been notified by the connected OPC server.

The reported event types are tightly linked to the Server's programming.



For example, the AUDIT window may also contain notifications as a consequence of alarm ACK or RESET events.

### Auditing Viewer Properties

#### Auditing window Advanced

**Toolbar Background:** Used to define the background color to apply to the Toolbar.

**Toolbar Foreground:** Used to define the foreground color to apply to the Toolbar.

#### Behaviour

**Editable in runtime:** enables the viewer object with edit mode so that it can be edited at Runtime. This option also allows you to save and call the created object's configuration.

See topic: "Objects Executions Properties".

#### User Access

See topic: "Draws and Object User Management" concerning objects and drawings that define the security parameters based on the project's User and Password management.

#### Audit window Style

**Alarm Area font Settings:** Allow you to set the alarm text font.

**Allow Column to be dropped:** Allow to drop column in runtime.

**Header Font Settings:** Allow you to set the column header font.

#### Style

**Border Brush:** This is used for setting the container's border colour.

**Border Thickness:** This is used for setting the container's border thickness.

**Foreground Color:** This is used for setting the colour of the letter and number characters used in the object.

**Background color:** This is used for setting the container's background colour.

**Font Size:** This is used for setting the font size used within the object.

**Disable Anti-aliasing:** Enables/disables the Anti-aliasing pixel correction function.

**Margin:** sets object's margins

## Fonts

**Font Family:** Sets font model to be used.

**Font Size:** Sets the font's size within the object.

**Font Style:** Sets the font's style.

**Font Weight:** Sets the thickness or thinness of how the font should be displayed (*light, extralight, normal etc...*).

## Layout

See paragraph on "Common ToolBox object and drawing properties".

## General

see the topic "Common ToolBox Object and Drawing Properties".

## Visibility

**Transparency:** Sets the object's Transparency level.

**Visibility:** enables the object's visibility.

**Visible on Client:** makes object visible on the webclient

**Visibility Level:** permits you to set the mask to determine the object's visibility.

**Zoom Level Visibility X:** permits you to set the Zoom factor on the x axis after which the object will become visible on screen.

**Zoom Level Visibility Y:** permits you to set the Zoom factor on the y axis after which the object will become visible on screen.

## Tag

See paragraph on "Common ToolBox object and drawing properties".

## User Management

See topic on "User Manager Properties" concerning objects and drawings that define the security parameters based on the project's User and Password management.

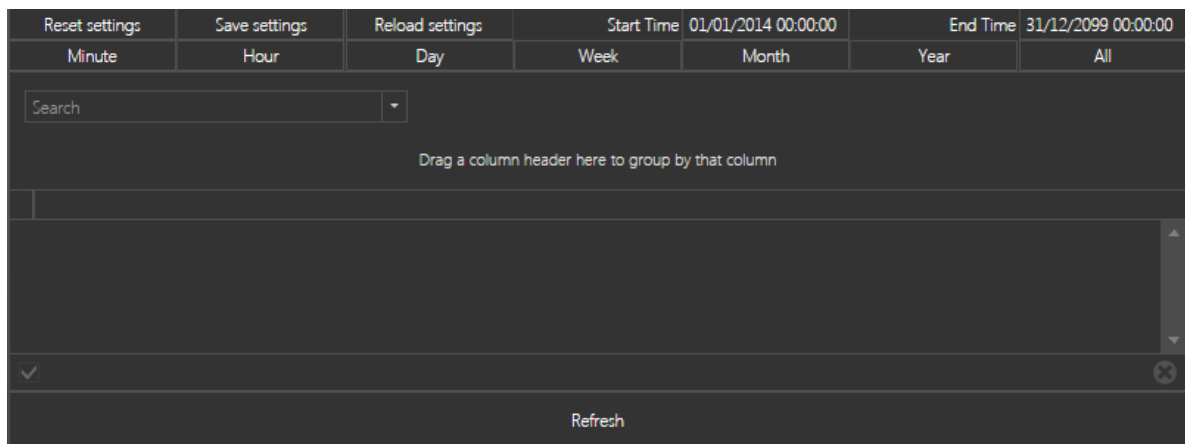
## Expressions

See the "Expressions in objects" topic concerning objects and drawings.

## Preserve Template Changes

See paragraph "Preserve local Draws and Objects properties" concerning original library objects

## 5.65. Data Logger Viewer



The Platform.Next Data Logger Viewer is a configurable object designed for displaying data recorded by the project's Data Loggers.

The Data Logger Viewer is available from the Toolbox 'Viewers' Group.

Any desired number of Data Logger Viewer objects can be inserted in various screens within the project as needed.

### Data Logger Viewer properties

#### Datalogger Viewer Style

**Row Font Settings:** Sets the font type for row contents.

**Header Font Settings:** Sets the font type for Header.

**Toolbar Foreground:** This is used for setting the toolbar's Foreground colour.

**Toolbar Background:** This is used for setting the toolbar's background colour.

#### Data Logger Viewer Option

**Connection String:** Used to set the connection string to referenced DB.

**Data Logger Name:** Used to set the name of the datalogger to be viewed.

**Filter Type:** Used to set a temporal filter for loading data from the database

**Max Rows:** Used to set the maximum number of rows to be retrieved and shown in the viewer.

#### Style

**Border Brush:** This is used for setting the container's border colour.

**Border Thickness:** This is used for setting the container's border thickness.

**Foreground Color:** This is used for setting the colour of the letter and number characters used in the object.

**Background color:** This is used for setting the container's background colour.

**Font Size:** This is used for setting the font size used within the object.

**Disable Anti-aliasing:** Enables/disables the Anti-aliasing pixel correction function.

**Margin:** Used to set the object's margins

## Fonts

**Font Family:** Used to set the font model to be used.

**Font Size:** Used to set the font's size within the object.

**Font Style:** Used to set the font's style.

**Font Weight:** Used to set the thickness or thinness of how the font should be displayed ( *light, extralight, normal* etc...).

## Layout

See paragraph on "Common ToolBox object and drawing properties"

## Behaviour

See paragraph on "Drawing and Object Behaviour Properties".

## General

see topic "Common ToolBox Object and Drawing Properties".

## Visibility

**Transparency:** Used to set the object's Transparency level.

**Visibility:** Used to set to enable the object's visibility.

**Visible on Client:** Used to set to make the object visible on the webclient

**Visibility Level:** Used to set the mask to determine the object's visibility.

**Zoom Level Visibility X:** Used to set the Zoom factor on the x axis after which the object will become visible on screen.

**Zoom Level Visibility Y:** Used to set the Zoom factor on the y axis after which the object will become visible on screen.

## Tag

See topic on: "Common ToolBox object and drawing properties"

## User Access

See topic on: "Draws and Object User Management" concerning objects and drawings that define the security parameters based on the project's User and Password management.

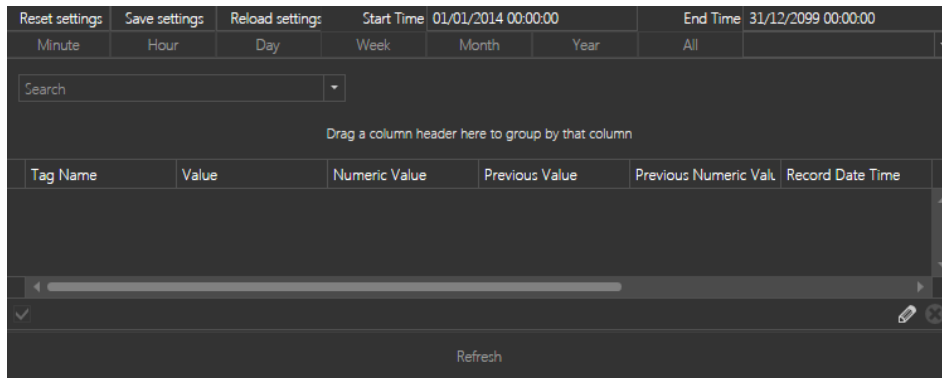
## Expressions

See topic on: "Managing Expression" concerning objects and drawings.

## Preserve Template Changes

**Preserve:** see paragraph "Preserve local Draws and Objects properties" concerning original library objects.

## 5.66. Historical Viewer



The "Historical Viewer" object is available in the "Viewers" group from the ToolBox (as a difference to the "Historical Data Viewers", this object loads data from the "UFUAAuditDataItem" table that contains data recorded by the "Historical Prototype", being historical variable records).

Several "Historical Viewers" can be inserted in a certain number of project screens as required.

### Historical Viewer Properties

#### Tag

See topic: "General Draws and object properties".

#### Historical Viewer Options

**Connection String:** Sets the string to connect to the referenced DB.

**Historical Name:** Sets the name of the Historical to display.

**Max Rows:** Maximum number of rows to be displayed in the object.

**Filter Type:** Sets a time range filter for loading data from the database.

#### Style

**Back Color:** This is used to set the container's back color.

**Stroke Color:** Sets the stroke color to be used for the numbers inside the container.

**Border Color:** This is used to set the container's border color.

**Border Thickness:** This is used to set the container's border thickness.

**Disable Anti-Aliasing:** This is used to Enable/Disable the Antialiasing function.

**Margin:** This is used to set the object's margins.

#### Layout

See topic: "General Draws and object properties".

#### User Access

See topic: "Draws and Objects User Access Properties" that defines the security parameters to use according to the project's User and Passwords management.

## Behaviour

See topic: "Draws and Objects Behaviour Properties".

## Fonts

**Font Family:** This is used to set the font model to use.

**Font Size:** This is used to set the font size of characters within the object.

**Font Style:** This is used to set the font style.

**Font Type:** This is used to set the font type to use ( *light, extralight, normal ecc...*).

## General

See topic: "General Draws and object properties".

## Expressions

See project: "Managing Expressions in Draws and Objects".

## Visibility

**Transparency:** This is used to set the object's transparency level.

**Visibility:** This is used to enable the object's visibility.

**Visible on WebClient:** This makes the object visible on WebClient.

**Visibility Layer Level:** This is used to set the mask which determines the object's visibility.

**Zoom Level Visibility X:** This is used to set the Zoom level on the X axis after which the object will become visible on screen.

**Zoom Level Visibility Y:** This is used to set the Zoom level on the Y axis after which the object will become visible on screen.

## Voice

**Speech Command:** This is used to set the object with a "speech command" to be used during runtime.

## Preserve Template Changes

**Preserve:** see topic "preserve properties of local Draws and Objects" in respect to the original object from the library.

## Historical Viewer Style

**Toolbar Foreground:** This is used to define the toolbar's color.

**Toolbar Background:** This is used to define the toolbar's back color.

**Header Font Settings:** This is used to set the Header's font type.

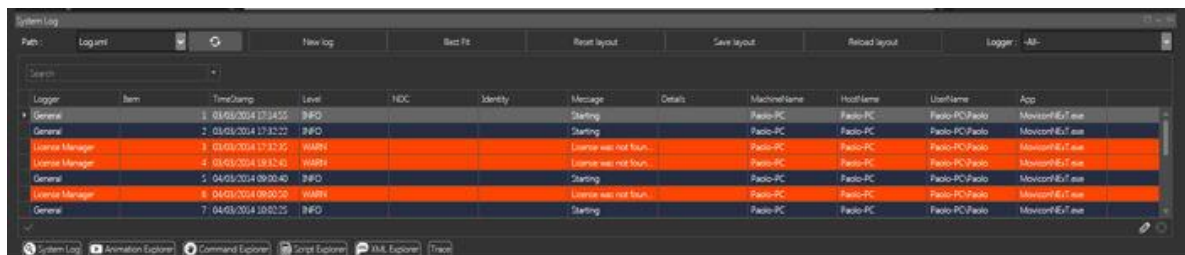
**Row Font Settings:** This is used to set the font type for the row contents and data columns.

## 5.67. System Log Window

The **System Log** window that reports information and error messages generated during project editing or runtime is located on the bottom border of the Movicon.NexT workspace.

This window is accessible during design engineering mode by using the Tabs located on the bottom border. It can also be opened during project Runtime with the "**SHIFT+F2**" keys.

In addition the "system log viewer" object is also available from the toolbox for inserting on screen.



The data shown in the "System log" window is saved on file. This file is created in:

C:\ProgramData\Progea\Movicon.NExT.Log\Log.xml.

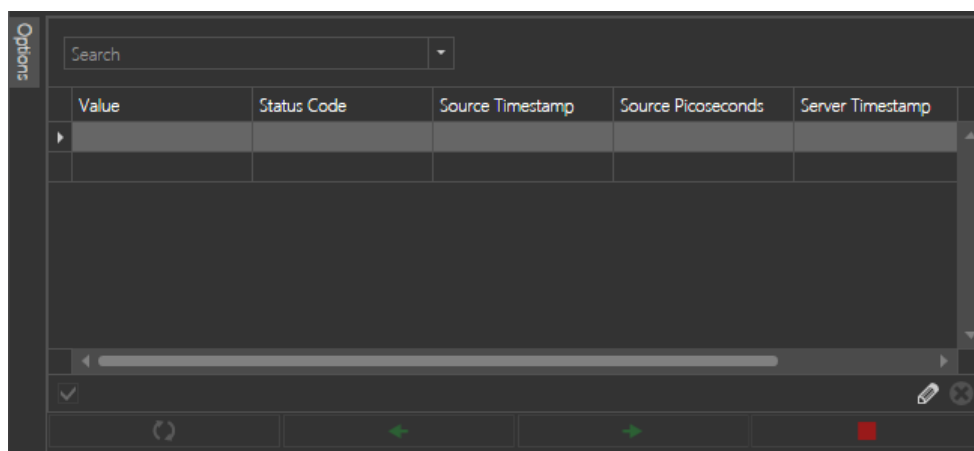
The information saved in the Log file is displayed as follows:

<b>Logger</b>	Module from which message derives, for example Server if from Server side or General if a project message.
<b>Item</b>	Generated message is shown by incremental ordinal number of occurrence.
<b>TimeStamp</b>	Date and time when message was generated.
<b>Level</b>	Indicates the type of message, INFO if it is a simple informational message, ERROR if it is an error
<b>NDC</b>	Work in progress
<b>Identity</b>	identified the user who executed the process that generated the error.
<b>Message</b>	Error message text.
<b>MachineName</b>	Name of PC in which message was collected.
<b>HostName</b>	Name of PC in which message was generated.
<b>UserName</b>	Name of Windows user which Movicon started up with.
<b>App</b>	Application which generated error (MoviconNExT.exe, MovNExTServer.exe...).

In addition to messages the System Log window also has some fields that can be used for filtering events for the "Logger" column or for searching texts.



## 5.68. Historical Grid



Data recorded in the relative DB can be viewed in table or chart format by associating the viewer with a TAG linked to a Historical Prototype. The graphical aspect of the table can be customized, e.g. to change size and/or column order, by using the "EDIT" command from the property window.

## Trend Viewer Properties

Please refer to the topic on "Historical Trend Server " for further information.

## 5.69. Miscellaneous Viewers

## 5.70. Calculator

The Movicon NExT calculator object has all the functions of a scientific calculator and, like all the other viewer objects, can be inserted on screen and configured through its various properties.



## Calculator Properties

### Style

**Border Brush:** This is used for setting the container's border colour.

**Border Thickness:** This is used for setting the container's border thickness.

**Foreground Color:** This is used for setting the colour of the letter and number characters used in the object.

**Background color:** This is used for setting the container's background colour.

**Font Size:** This is used for setting the font size used within the object.

**Disable Anti-aliasing:** Enables/disables the Anti-aliasing pixel correction function.

**Margin:** sets the object's margins.

### Fonts

**Font Family:** Sets font model to be used.

**Font Size:** Sets the font's size within the object.

**Font Style:** Sets the font's style.

**Font Weight:** Sets the Font's displayed thickness or thinness (*light, extralight, normal etc...*).

### Layout

See paragraph on "Common ToolBox object and drawing properties"

### Behaviour

See topic: "Objects Executions Properties".

### General

See topic: "Common ToolBox Object and Drawing Properties".

### Visibility

**Transparency:** Sets the object's Transparency level.

**Visibility:** Enables the object's visibility.

**Visible on Client:** Makes object visible on the webclient

**Visibility Level:** Permits you to set the mask to determine the object's visibility.

**Zoom Level Visibility X:** Permits you to set the Zoom factor on the x axis after which the object will become visible on screen.

**Zoom Level Visibility Y:** permits you to set the Zoom factor on the y axis after which the object will become visible on screen.

### Tag

See topic "Common ToolBox object and drawing properties"

### User Management

See topic: "Draws and Object User Management" concerning objects and drawings that define the security parameters based on the project's User and Password management.

### Expressions

See topic: "Managing Expression" concerning objects and drawings.

### Preserve Template Changes

**Preserve:** see paragraph "Preserve local Draws and Objects properties" concerning original library objects.

## 5.71. Media Element Control

The 'Multimedia Viewer' is used to display multimedia, such as video, audio and images, in runtime. This object has a toolbar which can be activated/deactivated through its properties. This toolbar can be used by the user to interact with object such as to control sound volume and reproduction speed.



### Multimedia Viewer Properties

#### Media settings

**Source File:** This is used to insert the completed path of the file to be reproduced.

**Loading Command:** This is used to set the command to be executed when object is loaded.

## Style

**Border Brush:** This is used to set the container's border colour.

**Border Thickness:** This is used to set the container's border thickness.

**Foreground Color:** This is used to set the colour of the letter and number characters used in the object.

**Background color:** This is used to set the container's background colour.

**Font Size:** This is used to set the font size used within the object.

**Disable Anti-aliasing:** Enables/disables the Anti-aliasing pixel correction function.

**Margin:** This is used to set the object's margins

## Fonts

**Font Family:** This is used to set the font model to be used.

**Font Size:** This is used to set the font's size within the object.

**Font Style:** This is used to set the font's style.

**Font Weight:** This is used to set the thickness or thinness of how the font should be displayed ( *light, extralight, normal etc...*).

## Layout

See topic on: "Common ToolBox object and drawing properties".

## Behaviour

See topic on: "Drawing and Object Behaviour Properties".

## General

See topic on: "Common ToolBox Object and Drawing Properties".

## Visibility

**Transparency:** This is used to set the object's Transparency level.

**Visibility:** This is used to enable the object's visibility.

**Visible on Client:** Enable this to make object visible on the webclient

**Visibility Level:** This is used to set the mask to determine the object's visibility.

**Zoom Level Visibility X:** This is used to set the Zoom factor on the x axis after which the object will become visible on screen.

**Zoom Level Visibility Y:** This is used to set the Zoom factor on the y axis after which the object will become visible on screen.

## Tag

See topic on: "Common ToolBox object and drawing properties".

## User Access

See topic on: "Draws and Object User Management" concerning objects and drawings that define the security parameters based on the project's User and Password management.

### Expressions

See topic on: "Managing Expression" concerning objects and drawings.

### Preserve Template Changes

**Preserve:** see topic on: "Preserve local Draws and Objects properties" concerning original library objects.

## 5.72. Media Element Control

The 'Multimedia Viewer' is used to display multimedia, such as video, audio and images, in runtime. This object has a toolbar which can be activated/deactivated through its properties. This toolbar can be used by the user to interact with object such as to control sound volume and reproduction speed.



### Multimedia Viewer Properties

#### Media settings

**Source File:** This is used to insert the completed path of the file to be reproduced.

**Loading Command:** This is used to set the command to be executed when object is loaded.

#### Style

**Border Brush:** This is used to set the container's border colour.

**Border Thickness:** This is used to set the container's border thickness.

**Foreground Color:** This is used to set the colour of the letter and number characters used in the object.

**Background color:** This is used to set the container's background colour.

**Font Size:** This is used to set the font size used within the object.

**Disable Anti-aliasing:** Enables/disables the Anti-aliasing pixel correction function.

**Margin:** This is used to set the object's margins

## Fonts

**Font Family:** This is used to set the font model to be used.

**Font Size:** This is used to set the font's size within the object.

**Font Style:** This is used to set the font's style.

**Font Weight:** This is used to set the thickness or thinness of how the font should be displayed ( *light, extralight, normal etc...*).

## Layout

See topic on: "Common ToolBox object and drawing properties".

## Behaviour

See topic on: "Drawing and Object Behaviour Properties".

## General

See topic on: "Common ToolBox Object and Drawing Properties".

## Visibility

**Transparency:** This is used to set the object's Transparency level.

**Visibility:** This is used to enable the object's visibility.

**Visible on Client:** Enable this to make object visible on the webclient

**Visibility Level:** This is used to set the mask to determine the object's visibility.

**Zoom Level Visibility X:** This is used to set the Zoom factor on the x axis after which the object will become visible on screen.

**Zoom Level Visibility Y:** This is used to set the Zoom factor on the y axis after which the object will become visible on screen.

## Tag

See topic on: "Common ToolBox object and drawing properties".

## User Access

See topic on: "Draws and Object User Management" concerning objects and drawings that define the security parameters based on the project's User and Password management.

## Expressions

See topic on: "Managing Expression" concerning objects and drawings.

## Preserve Template Changes

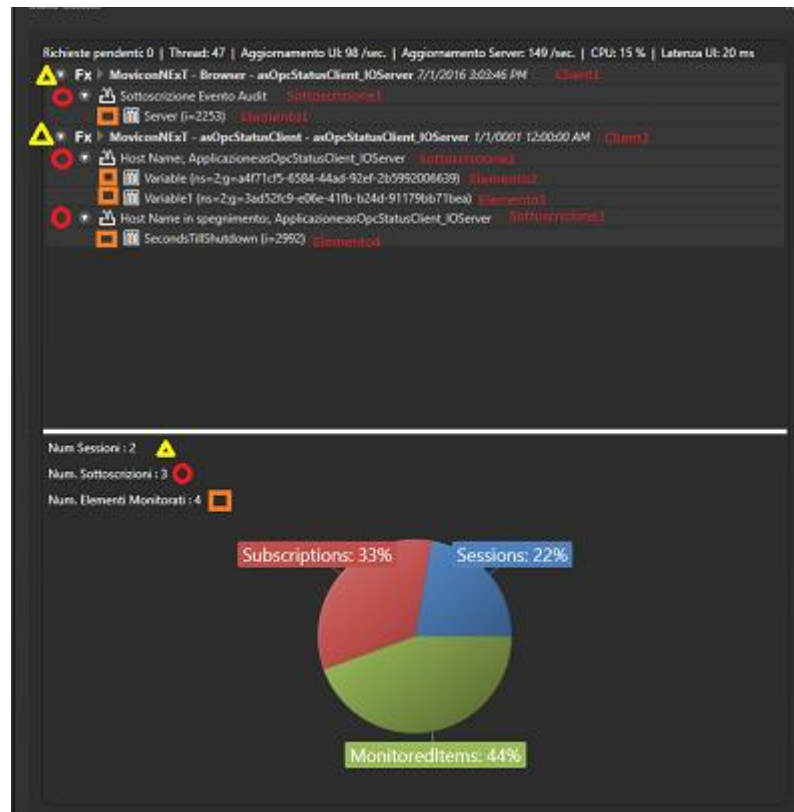
**Preserve:** see topic on: "Preserve local Draws and Objects properties" concerning original library objects.

## 5.73. OPC Client Status Viewer

This object is found in the Viewers Miscellaneous category and can be invoked when starting up the project in runtime from development mode with the "Shift + F1" combo keys. .

This window shows the connected OPC Client status.

Those clients that are connected to the local machine are highlighted in the window started up on the client.



This image shows the "MoviconNExT – Browser..." which is the opc browser window opened from development mode, while the "MoviconNExT – asOpcStatus – asOpcStatusClient\_IOServer" is the project's actual screen.

The sessions are the connections made by local clients, in this case there are two sessions in progress indicated by the yellow triangle.

The number of subscriptions refers to the subscribed group information and are highlighted with the red circle.

The monitored items are the subnodes relating to the variables put into use with additional subscribed information such as "SecondsTillShutdown" that indicates the remaining seconds before a redundant server or 'Server' shuts down.

## OPC Client Status Viewer Properties

### Style

**Border Brush:** This is used to set the container's border colour.

**Border Thickness:** This is used to set the container's border thickness.

**Back color:** This is used to set the container's background colour.

**Stroke Color:** This is used to set the colour of the letter and number characters used in the object.

**Disable Anti-aliasing:** Enables/disables the Anti-aliasing pixel correction function.

**Margin:** This is used to set the object's margins

## Fonts

**Font Family:** This is used to set the font model to be used.

**Font Size:** This is used to set the font's size within the object.

**Font Style:** This is used to set the font's style.

**Font Weight:** This is used to set the thickness or thinness of how the font should be displayed ( *light, extralight, normal* etc...).

## Layout

See topic on: "Common ToolBox object and drawing properties"

## Behaviour

See topic on: "Drawing and Object Behaviour Properties".

## General

See topic on: "Common ToolBox Object and Drawing Properties".

## Visibility

**Transparency:** Sets the object's Transparency level.

**Visibility:** enables the object's visibility.

**Visible On Web Client:** makes object visible on the webclient

**Visibility Layer Level:** permits you to set the mask to determine the object's visibility.

**Zoom Level Visibility X:** permits you to set the Zoom factor on the x axis after which the object will become visible on screen.

**Zoom Level Visibility Y:** permits you to set the Zoom factor on the y axis after which the object will become visible on screen.

## Tag

See topic on: "Common ToolBox object and drawing properties"

## User Access

See topic on: "Draws and Object User Management" concerning objects and drawings that define the security parameters based on the project's User and Password management.

## Expressions

See topic on: "Managing Expression" concerning objects and drawings.

## Preserve Template Changes

**Preserve:** See topic on: "Preserve local Draws and Objects properties" concerning original library objects.



## 5.74. Spreadsheet Viewer

The Spreadsheet Viewer object is used for displaying or modifying associated excel spreadsheets. If the spreadsheet is larger than the viewer screen, side scroll bars will display to facilitate user navigation.

	A	B	C
1	BREAKEVEN ANALYSIS		
2			
3			
4			
5	SALES		
6	Sales price per unit		\$150.00
7	Sales volume per period (units)		1,000
8	Total sales		\$150,000.00
9			
10	VARIABLE COSTS		
11	Commission per unit		\$4.00
12	Direct material per unit		\$60.00
13	Shipping per unit		\$5.00
14	Supplier per unit		\$2.00
15	Other variable costs per unit		\$9.60
16	Variable costs per unit		\$100.60
17	Total variable costs		\$100,600.00
18			
19	Unit contribution margin		49.40
20	Gross margin		\$49,400.00
21			
22	FIXED COSTS PER PERIOD		
23	Administrative costs		\$15,000.00
24	Insurance		\$12,000.00
25			

### SpreadSheet Viewer Properties

#### Spread Sheet Settings

**Show Ribbons:** Displays the ribbons for editing spreadsheets in the window.

**Document Path:** Sets the path from which document is to be loaded.

**Read-Only:** Sets the document to read only to impede user from modifying it.

#### Style

**Border Brush:** This is used to set the container's border colour.

**Border Thickness:** This is used to set the container's border thickness.

**Stroke Color:** This is used to set the border color.

**Disable Anti-aliasing:** Enables/disables the Anti-aliasing pixel correction function.

**Margin:** This is used to set the object's margins

## Appearance

**Background color:** This is used to set the container's background colour.

## Fonts

**Font Family:** Sets font model to be used.

**Font Size:** Sets the font's size within the object.

**Font Style:** Sets the font's style.

**Font Weight:** Sets the thickness or thinness of how the font should be displayed ( *light*, *extralight*, *normal* etc...).

## Layout

See topic on: "Common ToolBox object and drawing properties"

## Behaviour

See topic on: "Drawing and Object Behaviour Properties".

## General

See topic on: "Common ToolBox Object and Drawing Properties".

## Visibility

**Transparency:** This is used to set the object's Transparency level.

**Visibility:** This is used to enable the object's visibility.

**Visible on Client:** This is used to make the object visible on the webclient

**Visibility Level:** This is used to set the mask to determine the object's visibility.

**Zoom Level Visibility X:** This is used to set the Zoom factor on the x axis after which the object will become visible on screen.

**Zoom Level Visibility Y:** This is used to set the Zoom factor on the y axis after which the object will become visible on screen.

## Tag

See topic on: "Common ToolBox object and drawing properties"

## User Access

See topic on: "Draws and Object User Management" concerning objects and drawings that define the security parameters based on the project's User and Password management.

## Expressions

See topic on: "Managing Expression" concerning objects and drawings.

## Preserve Template Changes

**Preserve:** see paragraph "Preserve local Draws and Objects properties" concerning original library objects.



There are some restrictions as regards to using the object in web client. When clicking on the control, it will navigate through several spreadsheets. This navigation is cyclical. Therefore, when the last spreadsheet is reached, it will return back to the first one.

## 5.75. Stream Media Element

The "Stream Media Element" and "Media Element Control" objects are used to stream and view videos and film clips at runtime. The toolbar that this object is associated to can be activated/deactivated in its properties. This object comes with functions that the user can use to control the volume and reproduction speed as well as other things.

### Stream Media Element object properties

#### Stream Media Element Settings

**Streaming URL:** The complete URL of the media to be viewed.

**Force TCP :** This property is used to force a TCP connection.

**Stream User:** User who will connect to the service in streaming.

**Stream Password:** Password of user who will connect to the service in streaming.

**Is High Priority (default = false):** when activated, this property will consider the StreamMediaElement object as high priority and as a result will consume more resources but show a smoother video.

**Is Audio Disabled (default = false):** when activated, this will disable the streaming's audio source if present.

#### Media Element Settings

**Source File:** The complete path of the file to be reproduced is entered in this property.

**Loading Command:** This is used to set the command to execute when the object is loaded.

#### Tag

Please refer to topic on "General Draws and Object Properties".

#### Style

**Back Color:** This is used to set the container's back color.

**Stroke Color:** This is used to set the stroke color used within container.

**Border Color:** This is used to set the container border color.

**Border Thickness:** This is used to set the container border thickness.

**Disable Anti-Aliasing:** This is used to enable/disable the Antialiasing function.

**Margin:** This is used to set the object's margins.

#### Layout

Please refer to topic on: "General Draws and Object Properties".

#### User Access

Please refer to topic on: "Draws and Objects User Access Properties" that define the security parameters according to the project's Users and Passwords management.

## Behaviour

Please refer to topic on: "Draws and Object Behaviour Properties".

## Fonts

**Font Family:** This is used to set the Font type to be used.

**Font Size:** This is used to set the font size to use within the object.

**Font Style:** This is used to set the font style.

**Font Weight:** This is used to set the font boldness to be used ( *light, extralight, normal* etc...).

## General

Please refer to topic on: "General Draws and Object Properties".

## Preserve Template Changes

Preserve: please refer to topic on: "Preserve local Draws and Object properties" in respect to the original object in the library.

## Expressions

Please refer to topic on: "Managing Expression in Draws and Objects".

## Visibility

**Visibility Layer Level:** This is used to set the object's visibility layer level.

**Zoom Level Visibility X:** This is used to set the Zoom level on the X axis after which the object will become visible on screen.

**Zoom Level Visibility Y:** This is used to set the Zoom level on the Y axis after which the object will become visible on screen.

**Transparency:** This is used to set the object's Transparency level.

**Visibility:** This is used to enable the object's visibility.

**Visible On Web Client:** This is used to make the object visible on web Client.

## 6. Tag Assignment

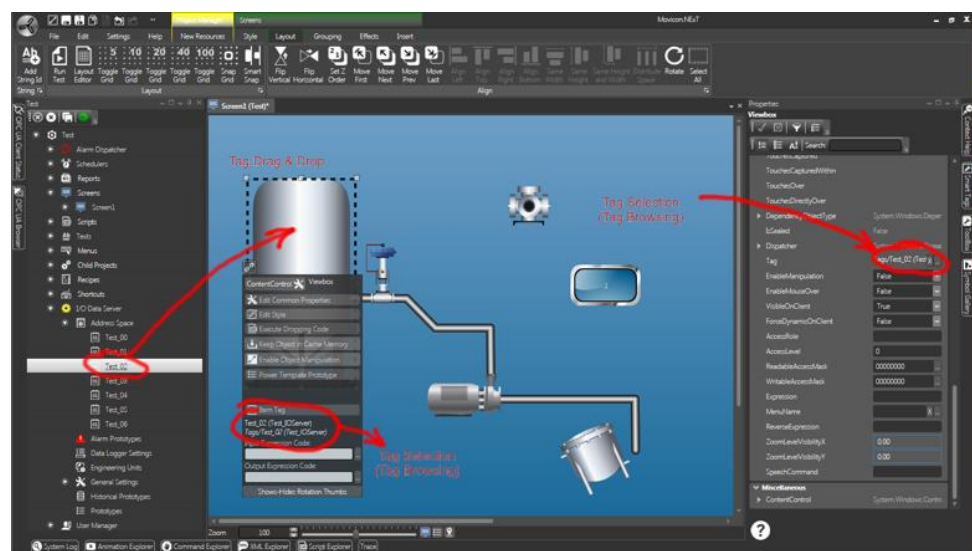
### 6.1. Assigning Tags to Objects

Normally a graphical screen object can be static or dynamic type. During runtime the dynamic object has the task of representing or controlling project Tag values.

For a graphical object to be dynamic it must have one or more Tags associated to its dynamic animations or control functions.

#### 6.1.1. The Reference Tag

Each graphical element can be associated with a Reference Tag. This tag will be referred to by each associated functionality unless another Tag has been defined for a specific object functionality.



Reference Tag can be assigned in different ways. The value of the Tag assigned to an object will be managed in the object's associated dynamic functionalities unless specified otherwise by other Tags.

#### Dragging Tags from the Tag List

A Tag can be assigned to an object by simply dragging it from the list to the object using the Drag & Drop techniques.

In order to do this proceed as follows:

1. Open the Address Space of the I/O Data Server module in the Workspace and display the Tag list located in the tree structure.
2. Select the Tag desired from this list and drag it slowly within the screen and drop it on the desired graphical object. This object will automatically be selected to receive the Tag whose properties can then be set.

## Selecting a Tag using the Tag Browser Window

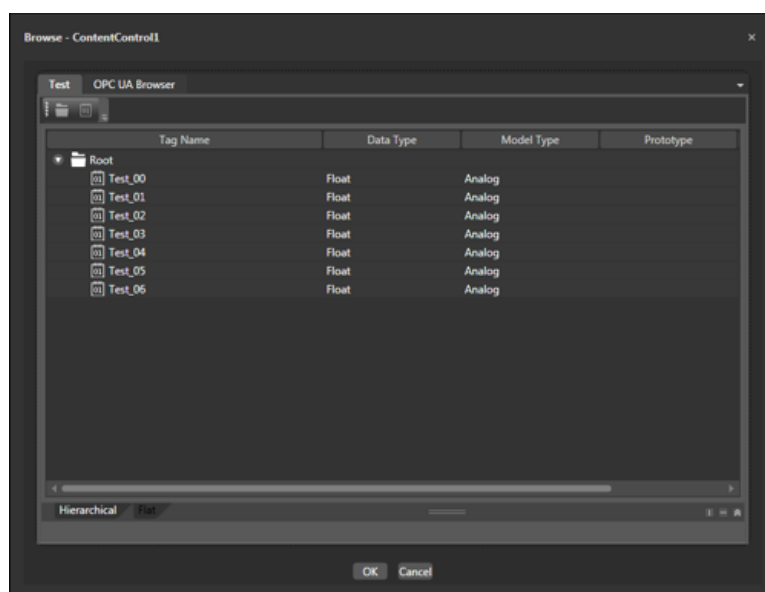
The Tag Browser window can also be used for selecting a Tag to associate to an object. The Tag Browser window can be accessed through the object's tooltip. To select a Tag using the Tag Browser window proceed as follows:

1. Select the object desired and click on the small button to activate the Object Menu.
2. Select the "Item Tag" from the command tooltip to access the Tag Browser window.
3. The Tag Browser window can also be opened by right clicking object to open a textual menu and then clicking on the "Data Context Editor" item.



*The Tag Browser can be accessed from the object command tooltip menu.*

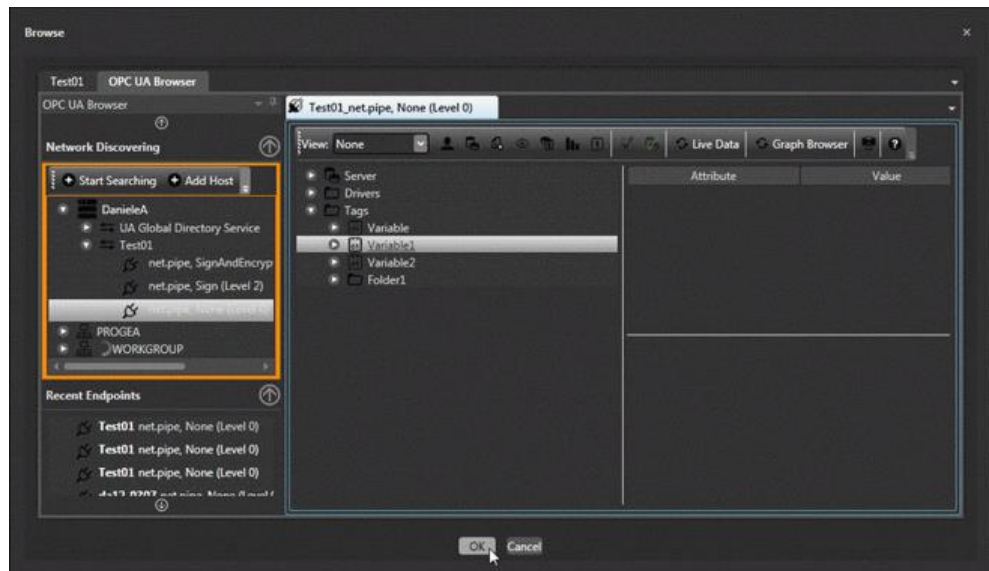
4. Select the Tag desired from the Tag Browser window and confirm with OK to associate the Tag with the object.



*An object Tag Browser window*

## Selecting an Item directly from a third party OPC UA Server

The "**OPC UA Browser**" tab can also be selected from the Tag Browser to access the list of OPC UA Servers available. This window can also be used for selecting the Tag desired and confirming with 'OK' to assign it to the object. This operation will only work when the project Server is launched beforehand.



*The OPC UA Server can also be accessed from the Tag Browser window.*

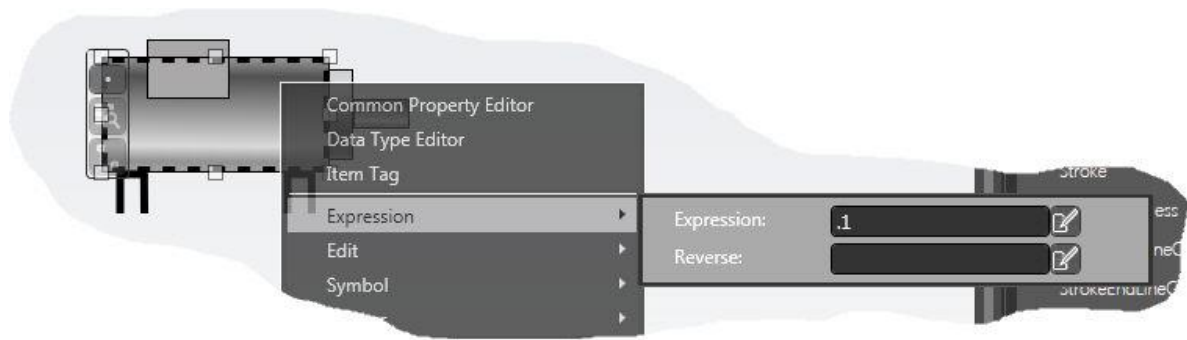
## 6.2. Pointing to Variable Bits and Array Variable Elements

Movicon supports the option to specify and assign single bits of byte, word double type variables or single elements of data Array variables when assigning Tag variables to symbols or object.

To specify the bit of a Tag variable assigned to a symbol or an object, you will need to use the "**Input Expression Code**" field which can be accessed with a right mouse click or using the object's command menu.



The bit numbers (starting with 0) to be assigned must be entered in the **expression** field after a dot. For example bit 1 should be inserted as `<.1>`.



Like with pointing to a variable bit, you can also assign the element of an Array Tag variable. Also in this case, after having associated the Array variable to the object, you must insert the element number to point to in the "Input Expression Code" field enclosing the number between squared brackets.

For example to point to the first Array element, being the 0 element, you must insert the "[0]" syntax.

## 6.3. Expressions in Objects

It is possible to insert expressions or formulae in the object's 'Expressions' and 'Inverse Expressions' property fields (or Animation properties). This possibility enables the system to calculate the set expression and display the result as an assigned object value.



**IMPORTANT:** Be reminded that "Excel 2013" is the syntax type to use for **expressions**, which is very different from the VB.NET syntax used in scripts. The advantage of using Excel expressions is that the script resources are not loaded for objects that use them making a significant improvement to performances.

For example, to execute the OR of two variables, the expression will be OR ([VAR0001],[VAR0002]). The wizard will in any case help you compose the expression desired.

If you wish to the value in the display resulting from an expression between a Tag and a constant calculation, you will need to assign the Tag to the object as a Contextual Tag and then insert the calculation expression in the Display's 'Expression' property. From this point onwards, the Display will no longer display the Tag value but the resulting expression's value at runtime.



**IMPORTANT:** when inserting an Expression in an object, you will need to assign one of the Tags managed in the expression as the object's contextual Tag.

In cases in which the object, for example the Display object, should be editable, it will be necessary to insert the 'Inverse Expression' by means of which the value's bidirectionality can be managed. When the control is also editable, it is necessary to retrieve the value with which to set the variable associated to the object when the control is edited.



For example, associating the "TagName1" variable to a Display object and then inserting the following formula in the "Expression" field:

$$=[x] + [\text{TagName2}] + 10$$

Where the placeholder "[x]" represents the value of the contextual Tag associated to the control. The square brackets, however, are used to indicate that the content is a Movicon.NExT variable. .

At this point, if the dynamic value of the "TagName1" associated to the Display object is 1, and the value of the "TagName2" is 2, the displayed value will be the result of the expression, that is to say:

$$1 + 2 + 10 = 13$$

If the Display were also editable, it would be necessary to apply the inverse formula to set the variable according to the inserted value. In this case, the 'Inverse Expression' field would be edited with the following formula:

$$=[x] - [\text{TagName2}] - 10$$

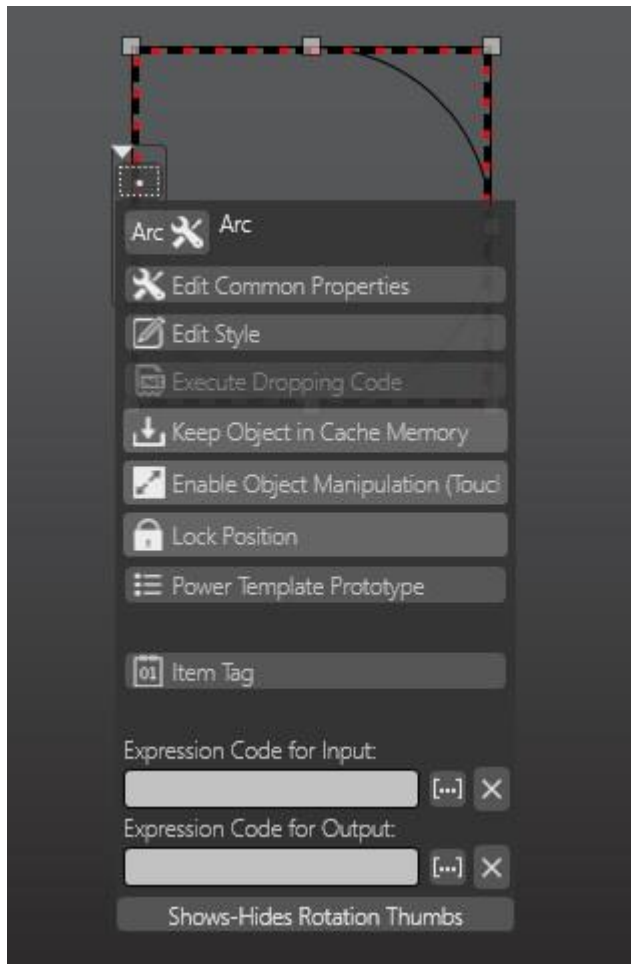
Where, in this case, the placeholder [x] represents the value inserted in the display object (and not the value of the Contextual Tag) and the result of the formula is then inserted in the contextual Tag associated to the object, for example "TagName1". At this point, if the value inserted in the Display 20 and the value of the 'TagName' were 2, the 'TagName1' would be set with the resulting value of the expression:

$$20 - 2 - 10 = 8$$


The inverse expression is valued only when the direct Expression has been inserted. Otherwise the Inverse Expression will be ignored.

To realize the above described example, please proceed as follows:

1. Open a screen and insert an object such as the Display object.
2. Assign the "TagName1" variable to the Display to use as the Display's Contextual Tag.
3. Insert the expression with the  $=[x] + [\text{TagName2}] + 10$  syntax in the Display's 'Expression' property.
4. Insert the expression with the  $=[x] - [\text{TagName2}] - 10$  syntax in the Display's 'Inverse Expression' property.
5. The Display will display the result of the expression at project runtime.



*Object Tooltip commands are used for inserting expressions or formulae for displaying data.*



Warning! A localization problem in converting double values currently exists. To remedy this problem for the time being you will need to set the English language in your local workstation in order to test expressions.

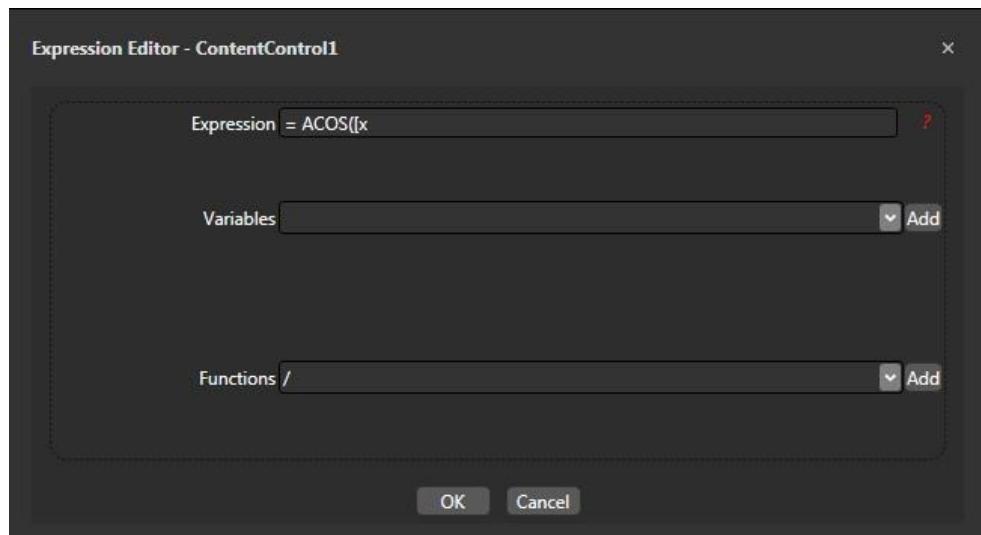


Attention! The expression manager does not recognize variable names as "Case Sensitive". Therefore the expression might not get valued correctly in cases where more than one variable have the same name but are distinguished by uppercase or lowercase letters (e.g. "TankLevel" and "tanklevel").

## Expression Syntax

Mathematical syntax types are used for expressions (eg. Ms Excel) to enable the combination of mathematical functions with project variables.

In the window used for setting the expressions you can access the below Expression Editor by clicking the edit icon on the right. This editor window will help you compose complex expressions.



Variable or formulae and functions can be added to the expression field by selecting them from the drop-down list indicated by the down pointed arrows on the right. After selecting the ones desired use the "Add" buttons to add them to Expression field above.



Warning! If a decimal value is inserted in the expression (eg 1.3 etc.) it will be necessary to use the dot (".") as decimal separator, independently from the Windows settings (eg: = [x] + [TagName2] + 1.3).

## 6.4. Function\_Reference\_Section

FUNCTION	Syntax	Description
<b>ABS</b>	<b>ABS(number)</b>	Returns the absolute value of a number. The absolute value of a non-negative number is the number itself. The absolute value of a negative number is -1 times the number.
<b>ACOS</b>	<b>ACOS(number)</b>	Returns the inverse cosine of a number. Inverse cosine is also referred to as arccosine. The arccosine is the angle whose cosine is the given number. The returned angle is given in radians in the range of 0 to pi.
<b>ACOSH</b>	<b>ACOSH(number)</b>	Returns the inverse hyperbolic cosine of a number. The number must be greater than or equal to 1. The inverse hyperbolic cosine is the

		value whose hyperbolic cosine is the given number.
<b>AND</b>	<b>AND(logical1, logical2, ...)</b>	Returns <b>True</b> if all the arguments have a logical value of True and returns False if at least one argument is False.
<b>ASIN</b>	<b>ASIN(number)</b>	Returns the inverse sine of a number. Inverse sine is also referred to as arcsine. The arcsine is the angle whose sine is the given number. The returned angle is given in radians in the range from -pi/2 to +pi/2.
<b>ASINH</b>	<b>ASINH(number)</b>	Returns the inverse hyperbolic sine of a number. The inverse hyperbolic sine is the value whose hyperbolic sine is the given number, so ASINH(SINH(number)) equals number.
<b>ATAN</b>	<b>ATAN(number)</b>	Returns the inverse tangent of a number. Inverse tangent is also known as arctangent. The arctangent is the angle whose tangent is a number. The returned angle is given in radians in the range from -pi/2 to +pi/2.
<b>ATAN2</b>	<b>ATAN2(x_num,y_num)</b>	Returns the inverse tangent of the specified x and y co-ordinates. The arctangent is the angle from the x-axis to a line containing the origin (0, 0) and the point (x_num, y_num). The angle is given in radians between -pi and pi, excluding -pi.
<b>ATANH</b>	<b>ATANH(number)</b>	Returns the inverse hyperbolic tangent of a number. Number must be strictly between -1 and 1. The inverse hyperbolic tangent is the value whose hyperbolic tangent is number, so ATANH(TANH(number))

		equals the given number.
<b>AVEDEV</b>	<b>AVEDEV(number1, number2, ...)</b>	Returns the average of the absolute mean deviations of data points. AVEDEV is a measure of the variability in a data set.
<b>AVERAGE</b>	<b>AVERAGE(number1, number2, ...)</b>	Returns the average (arithmetic mean) of the arguments.
<b>AVERAGEA</b>	<b>AVERAGEA(value1, value2,...)</b>	Calculates the average (arithmetic mean) of the values in the list of arguments. In addition to numbers and text logical values such as True and False are also included in the calculation.
<b>AVG</b>	<b>AVG(number1, number2,...)</b>	Returns the average (arithmetic mean) of the arguments.
<b>BINOMDIST</b>	<b>BINOMDIST(number_s, trials, probability_s, cumulative)</b>	Returns the individual term binomial distribution probability.
<b>CEILING</b>	<b>CEILING(number, significance)</b>	Returns number rounded up, away from zero, to the nearest multiple of significance. For example, if you want to avoid using pennies in your prices and your product is priced at \$4.82, use the formula =CEILING(4.82,0.05) to round prices up to the nearest nickel.
<b>CHAR</b>	<b>Char(number)</b>	The <b>Char</b> function returns the character whose number code is defined in the parameter.
<b>CHIDIST</b>	<b>CHIDIST(x, degrees_freedom)</b>	Returns the one-tailed probability of the chi-squared distribution. The $\chi^2$ distribution is associated with a $\chi^2$ test. Use the $\chi^2$ test to compare observed and expected values.
<b>CHIINV</b>	<b>CHIINV(probability, degrees_freedom)</b>	Returns the inverse of the one-tailed probability of the chi-squared ( $\chi^2$ ) distribution. If

		<p>probability = CHIDIST(x,...), then CHIIINV(probability,...) = x. Use this function to compare observed results with expected ones in order to decide whether your original hypothesis is valid.</p>
<b>CHITEST</b>	<b>CHITEST(actual_range, expected_range)</b>	<p>Returns the test for independence. CHITEST returns the value from the chi-squared (c2) distribution for the statistic and the appropriate degrees of freedom.</p>
<b>CHOOSE</b>	<b>Choose(index, valuearray)</b>	<p>The <b>Choose</b> function returns the value from a range of values on a specific index.</p>
<b>COLUMN</b>	<b>Column(range)</b>	<p>The <b>Column</b> function returns the column index of the provided column in range.</p>
<b>COMBIN</b>	<b>COMBIN(number, number_chosen)</b>	<p>Returns the number of combinations for a given number of items. Use COMBIN to determine the total possible number of groups for a given number of items.</p>
<b>CONCATENATE</b>	<b>CONCATENATE (text1, text2,...)</b>	<p>Joins several text strings into one text string.</p>
<b>CONFIDENCE</b>	<b>CONFIDENCE(alpha,standard_dev,size)</b>	<p>Returns a value that you can use to construct a confidence interval about a population mean. The confidence interval is a range of values. In your sample, mean x is at the center of this range and the range is <math>x \pm \text{CONFIDENCE}</math>. For example, if x is the sample mean of delivery times for products ordered through the mail, <math>x \pm \text{CONFIDENCE}</math> is a range of population means.</p>
<b>CORREL</b>	<b>CORREL(array1, array2)</b>	<p>Returns the correlation coefficient of the array1 and array2 cell ranges.</p>

<b>COS</b>	<b>COS(number)</b>	Returns the cosine of the given angle.
<b>COSH</b>	<b>COSH(number)</b>	Returns the hyperbolic cosine of a number.
<b>COUNT</b>	<b>COUNT(value1, value2,...)</b>	Counts the number of items in a list that contains numbers.
<b>COUNTA</b>	<b>COUNTA(value1, value2,...)</b>	Counts the number of cells that are not empty.
<b>COUNTBLANK</b>	<b>COUNTBLANK(range)</b>	Counts empty cells in a specified range of cells.
<b>COUNTIF</b>	<b>COUNTIF(range, criteria)</b>	Counts the number of cells within a range that meet the given criteria.
<b>COVAR</b>	<b>COVAR(array1, array2)</b>	Returns covariance, the average of the products of deviations for each data point pair.
<b>CRITBINOM</b>	<b>CRITBINOM(trials, probability_s, alpha)</b>	Returns the smallest value for which, the cumulative binomial distribution is greater than or equal to a criterion value.
<b>DATE</b>	<b>DATE(year, month, day)</b>	Returns the sequential serial number that represents a particular date.
<b>DATEVALUE</b>	<b>DATEVALUE(date_text)</b>	Returns the serial number of the date represented by the date_text.
<b>DAY</b>	<b>DAY(serial_number)</b>	Returns the day of a date, represented by a serial number. The day is given as an integer ranging from 1 to 31.
<b>DAYS360</b>	<b>DAYS360(start_date, end_date, method)</b>	Returns the number of days between two dates based on a 360-day year (twelve 30-day months) which, is used in some accounting calculations.
<b>DB</b>	<b>DB(cost, salvage, life, period, month)</b>	Returns the depreciation of an asset for a specified period using the fixed-declining balance method.

<b>DDB</b>	<b>DDB(cost, salvage, life, period, factor)</b>	Returns the depreciation of an asset for a specified period using the double-declining balance method or some other method you specify.
<b>DEGREES</b>	<b>DEGREES(angle)</b>	Converts radians into degrees.
<b>DEVSQ</b>	<b>DEVSQ(number1, number2,...)</b>	Returns the sum of squares of deviations of data points from their sample mean.
<b>DOLLAR</b>	<b>Dollar (number, decimal_places)</b>	The <b>Dollar</b> function converts a number to text, using a currency format. The format used is \$#,##0.00_);(\$#,##0.00).
<b>EVEN</b>	<b>EVEN(number)</b>	Returns the number rounded up to the nearest even integer.
<b>EXACT</b>	<b>Exact(value1, value2)</b>	The <b>Exact</b> function compares two values ignoring the styles and returns the boolean value as true or false.
<b>EXP</b>	<b>EXP(number)</b>	Returns e raised to the power of the given number.
<b>EXPONDIST</b>	<b>EXPONDIST(x, lambda, cumulative)</b>	Returns the exponential distribution.
<b>FACT</b>	<b>FACT(number)</b>	Returns the factorial of a number. The factorial of a number is the product of all positive integers <= the given number.
<b>FALSE</b>	<b>False(stringvalue)</b>	The <b>False</b> function returns the logical value for the false.
<b>FDIST</b>	<b>FDIST(x, degrees_freedom1, degrees_freedom2)</b>	Returns the F probability distribution.
<b>FIND</b>	<b>Find(lookfor, lookin, start)</b>	The <b>Find</b> function finds a portion of a string from a particular text and returns the location of the string.
<b>FINV</b>	<b>FINV(probability,deg_freedom1,deg_freedom2)</b>	The <b>Finv</b> function returns the inverse of the F probability distribution. If p =



		FDIST(x,...), then FINV(p,...) = x. Using F distribution, you can compare the degree of variability for two data sets.
<b>FISHER</b>	<b>FISHER(x)</b>	Returns the Fisher transformation at x. This transformation produces a function that is normally distributed rather than skewed.
<b>FISHERINV</b>	<b>FISHERINV(y)</b>	Returns the inverse of the Fisher transformation. If y = FISHER(x), then FISHERINV(y) = x.
<b>FIXED</b>	<b>Fixed ( number, decimal_places, no_commas )</b>	The <b>Fixed</b> function rounds off to a specified number of decimal places and returns the value in text format.
<b>FLOOR</b>	<b>FLOOR(number, significance)</b>	Rounds off the given number down, toward zero, to the nearest multiple of significance.
<b>FORECAST</b>	<b>FORECAST(x, known_ys, known_xs)</b>	Calculates a future value by using existing values using a linear regression. The predicted value is a y- value for a given x- value.
<b>FV</b>	<b>FV( interest_rate, number_payments, payment, PV, Type )</b>	The <b>FV</b> function returns the future value of an investment, based on an interest rate and a constant payment schedule.
<b>GAMMADIST</b>	<b>GAMMADIST(x, alpha,beta, cumulative)</b>	Returns the gamma distribution.
<b>GAMMAINV</b>	<b>Gammainv(p, alpha, beta)</b>	The <b>Gammainv</b> function returns the inverse function for the GAMMADIST function.
<b>GAMMAINV</b>	<b>GAMMAINV(probability, alpha, beta)</b>	Returns the inverse of the gamma cumulative distribution. If p = GAMMADIST(x,...), then GAMMAINV(p,...) = x.
<b>GEOMEAN</b>	<b>GEOMEAN(number1, number2,...)</b>	Returns the geometric mean of an array or range of positive data.

<b>GROWTH</b>	<b>=GROWTH(known_y's, [known_x's], [new_x's],</b>	This feature enables you to calculate predicted exponential growth using existing data. This calculates and returns an array of values used for the regression analysis. Growth enables you to perform a regression analysis.
<b>HARMEAN</b>	<b>HARMEAN(number1, number2,...)</b>	Returns the harmonic mean of a data set. The harmonic mean is the reciprocal of the arithmetic mean of reciprocals.
<b>HLOOKUP</b>	<b>HLOOKUP(lookup_value, table_array, row_index_num, range_lookup)</b>	Searches for a value in the top row of the array of values and then returns a value in the same column from a row you specify in the array. Use HLOOKUP when your comparison values are located in a row across the top of a table of data and you want to look down a specified number of rows. Use VLOOKUP when your comparison values are located in a column to the left of the data you want to find.
<b>HOURL</b>	<b>HOURL(serial_number)</b>	Returns the hour of a time value. The hour is given as an integer, ranging from 0 (12:00 A.M.) to 23 (11:00 P.M.).
<b>HYPGEOMDIST</b>	<b>Hypgeomdist(sample, numberofsample, population, numberofpopulation)</b>	The <b>Hypgeomdist</b> function returns the hypergeometric distribution.
<b>HYPEGEOMDIST</b>	<b>HYPGEOMDIST(sample_s, number_sample, population_s, number_population)</b>	Returns the hypergeometric distribution. HYPGEOMDIST returns the probability of a given number of sample successes, given the sample size, population successes and population size.
<b>IF</b>	<b>IF(logical_test, value_if_true, value_if_false)</b>	Returns one value if a condition you specify

		evaluates to True and another value if it evaluates to False. Use IF to conduct conditional tests on values and formulas.
<b>INDEX</b>	<b>Index(range,row,col)</b>	The <b>Index</b> function returns the exact value from the provided row index and column index from a specific range.
<b>INDIRECT</b>	<b>Indirect(content)</b>	The <b>Indirect</b> function returns the reference as a string instead of providing the content or range within it.
<b>INT</b>	<b>INT(number)</b>	Rounds a number down to the nearest integer.
<b>INTERCEPT</b>	<b>INTERCEPT(known_y's, known_x's)</b>	Calculates the point at which, the least squares fit line will intersect the y-axis.
<b>IPMT</b>	<b>IPMT(rate, per, nper, pv, fv, type)</b>	Returns the interest payment for a given period for an investment based on periodic, constant payments and a constant interest rate.
<b>IRR</b>	<b>IRR(values, guess)</b>	Returns the internal rate of return for a series of cash flows represented by the numbers in values. The cash flows must occur at regular intervals such as monthly or annually.
<b>ISBLANK</b>	<b>IsBlank( value )</b>	The <b>IsBlank</b> function checks for blank or null values.
<b>ISERR</b>	<b>IsErr( value )</b>	value is the value that you want to test. If the value is an error value (except #N/A), this function will return TRUE/FALSE to indicate whether a value is an error.
<b>ISERROR</b>	<b>ISERROR(value)</b>	Returns True if the value is a string that starts with a #.
<b>ISLOGICAL</b>	<b>IsLogical( value )</b>	The <b>IsLogical</b> function checks whether a value is a logical value and returns a TRUE or FALSE.

<b>ISNA</b>	<b>IsNA(value)</b>	The <b>IsNA</b> function returns a boolean value after determining that the provided value is a #NA error value.
<b>ISNONTEXT</b>	<b>IsNonText(text)</b>	The <b>IsNonText</b> function returns the boolean value after determining that the provided value is not a string.
<b>ISNUMBER</b>	<b>ISNUMBER(value)</b>	Returns True if the value parses as a numeric value.
<b>ISPMT</b>	<b>ISPMT(rate, per, nper, pv)</b>	Calculates the interest paid during a specific period of an investment.
<b>ISTEXT</b>	<b>IsText(text)</b>	The <b>IsText</b> function returns a boolean value after determining that the provided value is a string.
<b>KURT</b>	<b>KURT(number1, number2, ...)</b>	Returns the kurtosis of a data set. Kurtosis characterizes the relative peakedness or flatness of a distribution compared with the normal distribution. Positive kurtosis indicates a relatively peaked distribution. Negative kurtosis indicates a relatively flat distribution.
<b>LARGE</b>	<b>LARGE(array, k)</b>	Returns the k-th largest value in a data set.
<b>LEFT</b>	<b>LEFT(text, num_chars)</b>	LEFT returns the first character or characters in a text string, based on the number of characters you specify.
<b>LN</b>	<b>LN(number)</b>	Returns the natural logarithm of a number. Natural logarithms are based on the constant e (2.718281828459...).
<b>LEN</b>	<b>Len(text)</b>	LEN returns the length of a text string, including spaces.

<b>LOG</b>	<b>LOG(number, base)</b>	Returns the logarithm of a number to the base that you specify.
<b>LOG10</b>	<b>LOG10(number)</b>	Returns the base-10 logarithm of a number.
<b>LOGEST</b>	<b>=LOGEST(known_y's, [known_x's], [const], [stats])</b>	This feature enables you to calculate predicted exponential growth using existing data. This calculates and returns an array of values used for the regression analysis. Logest calculates and returns an array of values that is used in regression analysis.
<b>LOGINV</b>	<b>LOGINV(probability, mean, standard_dev)</b>	Returns the inverse of the lognormal cumulative distribution function of x, where $\ln(x)$ is normally distributed with parameters mean and standard_dev. If $p = \text{LOGNORMDIST}(x, \dots)$ , then $\text{LOGINV}(p, \dots) = x$ .
<b>LOGNORMDIST</b>	<b>LOGNORMDIST(x, mean, standard_dev)</b>	Returns the cumulative lognormal distribution of x, where $\ln(x)$ is normally distributed with parameters mean and standard_dev.
<b>LOWER</b>	<b>Lower( text )</b>	The <b>Lower</b> function converts all characters in the specified text string to lowercase. Characters in the string that are not letters, are not changed.
<b>MATCH</b>	<b>Match( value, array, match_type )</b>	The <b>Match</b> function searches for a specified value in an array and returns the relative position of that item.
<b>MAX</b>	<b>MAX(number1, number2, ...)</b>	Returns the largest value in a set of values.
<b>MAXA</b>	<b>MAXA(value1, value2, ...)</b>	Returns the largest value in a list of arguments. Text and logical values such as True and False are compared as well as numbers.
<b>MEDIAN</b>	<b>MEDIAN(number1, number2, ...)</b>	Returns the median of the given numbers. The

		median is the number in the middle of a set of numbers; that is, half the numbers have values that are greater than the median and half have values that are less.
<b>MID</b>	<b>MID(text, start_position, num_chars)</b>	MID returns a text segment of a character string. The parameters specify the starting position and the number of characters.
<b>MIN</b>	<b>MIN(number1, number2, ...)</b>	Returns the smallest number in a set of values.
<b>MINA</b>	<b>MINA(value1, value2, ...)</b>	Returns the smallest value in the list of arguments. Text and logical values such as True and False are compared as well as numbers.
<b>MINUTE</b>	<b>MINUTE(serial_number)</b>	Returns the minutes of a time value. The minute is given as an integer, ranging from 0 to 59.
<b>MIRR</b>	<b>MIRR(values, finance_rate, reinvest_rate)</b>	Returns the modified internal rate of return for a series of periodic cash flows.
<b>MOD</b>	<b>MOD(number, divisor)</b>	Returns the remainder after the number is divided by a divisor. The result has the same sign as the divisor.
<b>MODE</b>	<b>MODE(number1, number2, ...)</b>	Returns the most frequently occurring or repetitive, value in an array or range of data.
<b>MONTH</b>	<b>MONTH(serial_number)</b>	Returns the month of a date represented by a serial number. The month is given as an integer, ranging from 1 (January) to 12 (December).
<b>NEGBINOMDIST</b>	<b>NEGBINOMDIST(number_f, number_s, probability_s)</b>	Returns the negative binomial distribution. NEGBINOMDIST returns the probability that there will be number_f failures before the number_s-th

		success, when the constant probability of a success is probability_s.
<b>NORMDIST</b>	<b>NORMDIST(x, mean, standard_dev, cumulative)</b>	Returns the normal distribution for the specified mean and standard deviation.
<b>NORMINV</b>	<b>NORMINV(probability, mean, standard_dev)</b>	Returns the inverse of the normal cumulative distribution for the specified mean and standard deviation.
<b>NORMSDIST</b>	<b>NormsDist(value)</b>	The <b>NormsDist</b> function returns the probability that the observed value of a standard normal random variable will be less than or equal to the parameter.
<b>NORMSINV</b>	<b>NormsInv(value)</b>	The <b>NormsInv</b> function returns the standard normal random variable that has <i>Mean</i> 0 and <i>Standard Deviation</i> 1
<b>NOT</b>	<b>NOT(logical)</b>	Reverses the value of its argument.
<b>NOW</b>	<b>NOW( )</b>	Returns the serial number of the current date and time.
<b>NPER</b>	<b>NPER(rate, pmt, pv, fv, type)</b>	Returns the number of periods for an investment based on periodic, constant payments and a constant interest rate.
<b>NPV</b>	<b>NPV(rate, value1, value2, ...)</b>	Calculates the net present value of an investment by using a discount rate and a series of future payments (negative values) and income (positive values).
<b>ODD</b>	<b>ODD(number)</b>	Returns the number rounded up to the nearest odd integer.
<b>OFFSET</b>	<b>Offset( range, rows, columns, height, width )</b>	The <b>Offset</b> function returns a reference to a range that is offset a number of rows and columns from any given range or cell.

<b>OR</b>	<b>OR(logical1, logical2, ...)</b>	Returns True if any argument is True; returns False if all arguments are <b>False</b> .
<b>PEARSON</b>	<b>PEARSON(array1, array2)</b>	Returns the Pearson product moment correlation coefficient, <i>r</i> , a dimensionless index that ranges from -1.0 to 1.0 inclusive and reflects the extent of a linear relationship between two data sets.
<b>PERCENTILE</b>	<b>PERCENTILE(array, k)</b>	Returns the <i>k</i> -th percentile of values in a range.
<b>PERCENTRANK</b>	<b>PERCENTRANK(array, x, significance)</b>	Returns the rank of a value in a data set as a percentage of the data set.
<b>PERMUT</b>	<b>Permut(n, k)</b>	The <b>Permut</b> function returns the number of permutations of <i>n</i> items taken at <i>k</i> time.
<b>PI</b>	<b>PI( )</b>	Returns the number 3.14159265358979, the mathematical constant pi, accurate to 15 digits.
<b>PMT</b>	<b>PMT(rate, nper, pv, fv, type)</b>	Calculates the payment for a loan based on constant payments and a constant interest rate.
<b>POISSON</b>	<b>POISSON(x, mean, cumulative)</b>	Returns the Poisson distribution.
<b>POW</b>	<b>POW(number, power)</b>	The <b>Pow</b> function returns the result of a number raised to a power.
<b>POWER</b>	<b>POWER(number, power)</b>	Returns the result of a number raised to a power.
<b>PPMT</b>	<b>PPMT(rate, per, nper, pv, fv, type)</b>	Returns the payment on the principal for a given period, for an investment based on periodic, constant payments and a constant interest rate.
<b>PROB</b>	<b>PROB(x_range, prob_range, lower_limit, upper_limit)</b>	Returns the probability whose values are in a range that is between two limits. If upper_limit is not supplied, returns



		the probability that values in x_range are equal to lower_limit.
<b>PRODUCT</b>	<b>PRODUCT(number1, number2, ...)</b>	Multiplies all the numbers given as arguments and returns the product.
<b>PV</b>	<b>PV(rate, nper, pmt, fv, type)</b>	Returns the present value of an investment. The present value is the total amount that a series of future payments is worth now.
<b>QUARTILE</b>	<b>QUARTILE(array, quart)</b>	Returns the quartile of a data set.
<b>RADIANS</b>	<b>RADIANS(angle)</b>	Converts degrees to radians.
<b>RAND</b>	<b>RAND( )</b>	Returns an evenly distributed random number greater than or equal to 0 and less than 1.
<b>RANK</b>	<b>RANK(number, ref, order)</b>	Returns the rank of a number in a list of numbers. The rank of a number is its size relative to other values in a list. (If you were to sort the list, the rank of the number would be its position.)
<b>RATE</b>	<b>RATE(nper, pmt, pv, fv, type, guess)</b>	Returns the interest rate per period of an annuity. RATE is calculated by iteration and may not converge to a unique solution.
<b>RIGHT</b>	<b>RIGHT(text, num_chars)</b>	RIGHT returns the last character or characters in a text string, based on the number of characters you specify.
<b>ROUND</b>	<b>ROUND(number, num_digits)</b>	Rounds a number to a specified number of digits.
<b>ROUNDDOWN</b>	<b>ROUNDDOWN(number, num_digits)</b>	Rounds a number down towards zero.
<b>ROUNDUP</b>	<b>ROUNDUP(number, num_digits)</b>	Rounds a number up away from 0 (zero).
<b>RSQ</b>	<b>RSQ(known_y's, known_x's)</b>	Returns the square of the Pearson product moment correlation

		coefficient through data points in known_y's and known_x's.
<b>SECOND</b>	<b>SECOND(serial_number)</b>	Returns the seconds of a time value. The second is given as an integer in the range 0 (zero) to 59.
<b>SIGN</b>	<b>SIGN(number)</b>	Determines the sign of a number. Returns 1 if the number is positive, zero (0) if the number is 0 and -1 if the number is negative.
<b>SIN</b>	<b>SIN(number)</b>	Returns the sine of the given angle.
<b>SINH</b>	<b>Sinh(value)</b>	The <b>SinH</b> function computes the hyperbolic sine of the argument.
<b>SKEW</b>	<b>SKEW(number1, number2, ...)</b>	Returns the skewness of a distribution. Skewness characterizes the degree of asymmetry of a distribution around its mean.
<b>SLN</b>	<b>SLN(cost, salvage, life)</b>	Returns the straight-line depreciation of an asset for one period.
<b>SLOPE</b>	<b>SLOPE(known_y's, known_x's)</b>	Returns the slope of the linear regression line through data points in known_y's and known_x's. The slope is the rate of change along the regression line.
<b>SMALL</b>	<b>SMALL(array, k)</b>	Returns the k-th smallest value in a data set.
<b>SQRT</b>	<b>SQRT(number)</b>	Returns a positive square root.
<b>STANDARDIZE</b>	<b>STANDARDIZE(x, mean, standard_dev)</b>	Returns a normalized value from a distribution characterized by mean and standard_dev.
<b>STDEV</b>	<b>STDEV(number1, number2, ...)</b>	Estimates the standard deviation based on a sample. The standard deviation is a measure of how widely values

		are dispersed from the average value (the mean).
<b>STDEVA</b>	<b>STDEVA(value1, value2 , ...)</b>	Estimates standard deviation based on a sample. The standard deviation is a measure of how widely values are dispersed from the average value (the mean). Text and logical values such as True and False are also included in the calculation.
<b>STDEVP</b>	<b>STDEVP(number1, number2, ...)</b>	Calculates standard deviation based on the entire population given as arguments.
<b>STDEVPA</b>	<b>STDEVPA(value1, value2, ...)</b>	Calculates the standard deviation based on the entire population given as arguments, including text and logical values.
<b>STEYX</b>	<b>STEYX(known_y's, known_x's)</b>	Returns the standard error of the predicted y-value for each x in the regression.
<b>SUBSTITUTE</b>	<b>SUBSTITUTE(text, old_text, new_text, instance_num)</b>	Substitutes new_text for old_text in a text string. Use SUBSTITUTE when you want to replace specific text in a text string; use REPLACE when you want to replace any text that occurs in a specific location in a text string.
<b>SUM</b>	<b>Sum( number1, number2, ... number_n )</b>	The <b>Sum</b> function adds all numbers in a range of cells and returns the result.
<b>SUMIF</b>	<b>SumIf( range, criteria, sum_range )</b>	<b>SumIf</b> function adds the specified range of cells by a given criteria.
<b>SUMPRODUCT</b>	<b>SUMPRODUCT(array1, array2, array3, ...)</b>	Multiplies corresponding components in the given arrays and returns the sum of those products.
<b>SUMSQ</b>	<b>SUMSQ(number1, number2, ...)</b>	Returns the sum of the squares of the arguments.
<b>SUMXMY2</b>	<b>SumXmY2( array1, array2 )</b>	The <b>SumXmY2</b> function calculates the

		sum of the squares of the differences between the corresponding items in the arrays and returns the sum as results.
<b>SUMX2MY2</b>	<b>SUMX2MY2(array_x, array_y)</b>	Returns the sum of the difference of squares of corresponding values in two arrays.
<b>SUMX2PY2</b>	<b>SUMX2PY2(array_x, array_y)</b>	Returns the sum of the sum of squares of corresponding values in two arrays. The sum of the sum of squares is a common term in many statistical calculations.
<b>SYD</b>	<b>SYD(cost, salvage, life, per)</b>	Returns the sum-of-years' digits depreciation of an asset for a specified period.
<b>TAN</b>	<b>TAN(number)</b>	Returns the tangent of a number.
<b>TANH</b>	<b>TANH(number)</b>	Returns the hyperbolic tangent of a number.
<b>TEXT</b>	<b>TEXT(value, format_text)</b>	Converts a value to text in a specific number format.
<b>TIME</b>	<b>TIME(hour, minute, second)</b>	Returns the decimal number for a particular time. The decimal number returned by TIME is a value ranging from 0 (zero) to 0.99999999, representing the times from 0:00:00 (12:00:00 A.M.) to 23:59:59 (11:59:59 P.M.).
<b>TIMEVALUE</b>	<b>TIMEVALUE(time_text)</b>	Returns the decimal number of the time represented by a text string. The decimal number is a value ranging from 0 (zero) to 0.99999999, representing the times from 0:00:00 (12:00:00 A.M.) to 23:59:59 (11:59:59 P.M.).
<b>TODAY</b>	<b>TODAY( )</b>	Returns the serial number of the current date. The serial number is the number of days since Jan 1, 1900.

<b>TRIM</b>	<b>Trim( text )</b>	The Trim function returns a text value with the leading and trailing spaces removed.
<b>TRIMMEAN</b>	<b>TRIMMEAN(array, percent)</b>	Returns the mean of the interior of a data set. TRIMMEAN calculates the mean taken by excluding a percentage of data points from the top and bottom tails of a data set.
<b>TRUE</b>	<b>True(stringvalue)</b>	The <b>True</b> function returns the logical value for True.
<b>TRUNC</b>	<b>TRUNC(number, num_digits)</b>	Truncates a number to an integer by removing the fractional part of the number.
<b>UPPER</b>	<b>Upper( text )</b>	The <b>Upper</b> function converts all characters in a text string to uppercase.
<b>VALUE</b>	<b>Value(range)</b>	The <b>Value</b> function computes the date or a string that contains the number, and converts it into number format.
<b>VAR</b>	<b>Var( number1, number2, ... number_n )</b>	The Var function returns the variance of a population based on sample of numbers.
<b>VARA</b>	<b>VarA( value1, value2, ... value_n )</b>	The <b>VarA</b> function returns the variance of a population based on a sample of numbers, text, and logical values (ie: TRUE or FALSE).
<b>VARP</b>	<b>VarP(listofvalues)</b>	The VarP function returns population variance of the listed values.
<b>VARPA</b>	<b>VARPA(value1, value2, ...)</b>	Calculates variance based on the entire population. In addition to numbers and text, logical values such as True and False are also included in the calculation.
<b>VDB</b>	<b>VDB(cost, salvage, life, start_period, end_period, factor, no_switch)</b>	Returns the depreciation of an asset for any period you specify, including

		partial periods, using the double-declining balance method or some other method you specify. VDB stands for variable declining balance.
<b>VLOOKUP</b>	<b>VLOOKUP(lookup_value, table_array, col_index_num, range_lookup)</b>	Searches for a value in the left most column of a table and then returns a value in the same row from a column you specify in the table. Use VLOOKUP instead of HLOOKUP when your comparison values are located in a column to the left of the data you want to find.
<b>WEEKDAY</b>	<b>WEEKDAY(serial_number,return_type)</b>	Returns the day of the week corresponding to a date. The day is given as an integer, ranging from 1 (Sunday) to 7 (Saturday) by default.
<b>WEIBULL</b>	<b>WEIBULL(x,alpha,beta,cumulative)</b>	The <b>Weibull</b> function returns the Weibull distribution. This distribution is used in reliability analysis, such as calculating a device's mean time to failure.
<b>XIRR</b>	<b>Xirr(cashflow, datelist, value)</b>	The <b>Xirr</b> function computes the internal rate of return for a schedule of possibly non-periodic cash flows.
<b>YEAR</b>	<b>YEAR(serial_number)</b>	Returns the year corresponding to a date. The year is returned as an integer in the range 1900-9999.
<b>ZTEST</b>	<b>ZTEST(array, u0, sigma)</b>	Returns the one-tailed probability-value of a z-test.

For further information about function usability please refer to the Microsoft Office Excel guide.

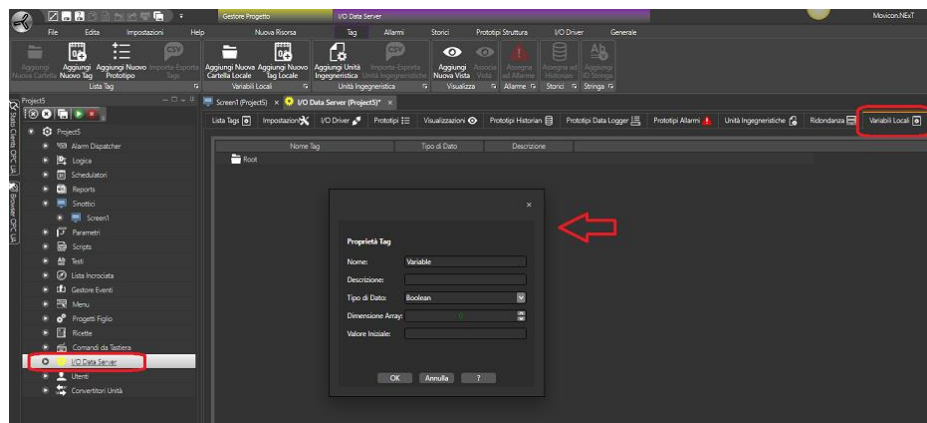
## 6.5. Local Client Tags

The local project tags are tags that the programmer can create within the project and use in screen controls or script. The Local Tags are not deployed in the Server's address space and are only visible at Client level.

The Local Tags, such as Server Tags, can be created with names defined by the programmer and of different types grouped in folders at several nested levels. However, local tags have certain limits in respect to the Server Tags such as:

- They can be defined with a "Data Type" but not as "Model Type"
- They cannot be exchanged with the field using a communication driver
- They cannot be assigned with engineering units
- They cannot be retentive
- They cannot be associated with alarms, historical or Data Logger prototypes
- They cannot be managed with user access levels
- They cannot be Structure, Digital or Enum type but only simple or Array types

The Local Variables can be created using an appropriate editor that can be accessed from the "Local Variables" tab once opening the "I/O Data Server" resource. They can also be created directly from the Tag Browser window which is also used for assigning variables to objects.



It is possible to identify whether the variable associated from a control is Local or Server type by the notation shown within the brackets after the tag's name in the title of the object's "General Properties" window or in the tooltip which appears when clicking on the first small button at the top of the object's menu. The notation within the brackets will show "TemporaryVariable" when the variable is a local type or the Server's AppName, for example "Project1\_IOServer", when a Server Tag.

## 6.6. Client system tags

Movicon provides users with a set of predefined tags that are used to obtain system information or for simulation curves. The System Tags are not published in the Server's address space and are only visible at Client level only.

The System Tags can be selected and assigned to project controls using the "Tag Browse Window".

Each System Tag has a specific function as described in the table below:

Variable Name	Data Type	Description
Blink2500ms	Integer	This variable blinks between "0" and "1" with a frequency of 2.5 seconds.
Blink5s	Integer	This variable blinks between "0" and "1" with a frequency of 5 seconds.
Blink10s	Integer	This variable blinks between "0" and "1" with a frequency of 10 seconds.
Blink30s	Integer	This variable blinks between "0" and "1" with a frequency of 30 seconds.
Blink200ms	Integer	This variable blinks between "0" and "1" with a frequency of 200 milliseconds.
Blink500ms	Integer	This variable blinks between "0" and "1" with a frequency of 500 milliseconds.
Blink1s	Integer	This variable blinks between "0" and "1" with a frequency of 1 second.
Sine	Float	This variable generates a simulation of a sine of an angle sine that varies from 0 to 360 degrees, with values ranging from "1" to "-1".
Square	Integer	This variable generates a simulation of a square wave with values ranging from "1" to "-1".
Triangle	Float	This variable generates a simulation of a triangle wave with values ranging from "1" to "-1".
Sawtooth	Float	This variable generates a simulation of a sawtooth wave with values ranging from "1" to "-1".
Pulse	Integer	This variable generates an impulse each second at value "1" to then return to value "0".
WhiteNoise	Float	This variable simulates a white noise at a value range between "1" and "-1".
GaussNoise	Float	This variable simulates a Gauss Noise at a value range between "1" and "-1".
DigitalNoise	Integer	This variable simulates a digital noise at a value range between "1" and "1".
LastHostNameUsed	String	This variable reports the name of the last Server to which the Client connected. The Client may also be connected to other Servers at the same time. This variable only shows the name of the Server subscribed to in the last connection.
CurrentTime	String	This variable reports the system's time.
CurrentDate	String	This variable reports the system's date.
CurrentLongDate	String	This variable reports the system's time and data. (e.g. Wednesday, January 27, 2016).



CurrentUser	String	This variable contains the name of the active user. If no user is logged on, the variable's value will be null.
MouseMove	Boolean	Indicates mouse is moving when active.
ActiveScreen	String	Name of the last active screen.
PI	Float	This variable reports the PC's performance index.
GPI	Float	This variable reports the PC's graphics performance index.
NLP	Integer	This variable reports the number of the PC's logic processors.
NP	Integer	This variable reports the number of the PC's physical processors.
CurrentRole	String	This variable contains the currently active user's role value. If no user is logged in, this value will be null.
CurrentAccessLevel	Integer	This variable contains the currently active user's access level value. If no user is logged in, this value will be '0'.
CurrentAccessMask	Integer	This variable contains the currently active user's access mask value. If no user is logged in, this value will be '0'.



System variables are read only.



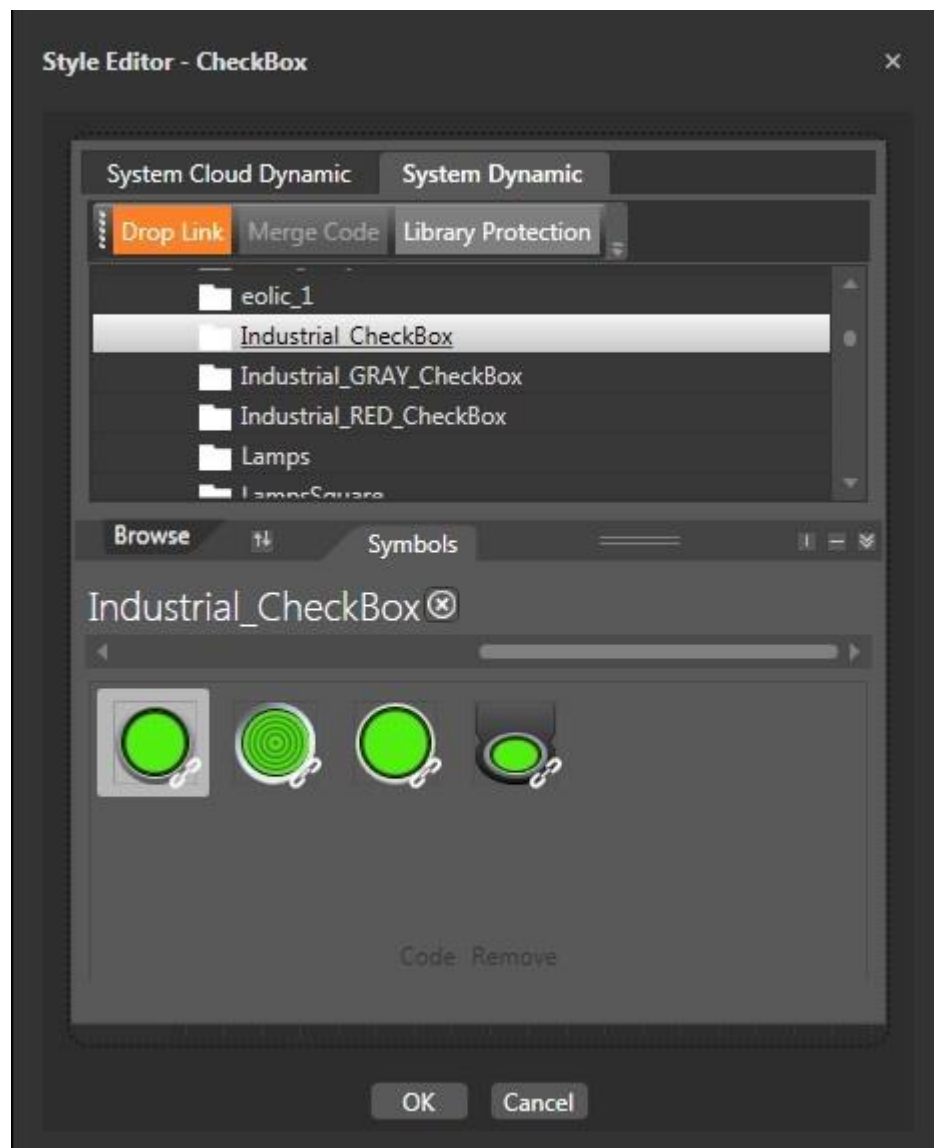
## 7. Styles

### 7.1. Using Styles

The Movicon.NExT graphics technology uses the intrinsic features of WPF to allow the uncoupling of each object's graphical part from its logic or mechanical functional part. This feature is expressed through the use of "Styles".

After having inserted and configured the dynamic properties of a graphical symbol from the library, or an object from the toolbox, its "style" can be changed at any given moment afterwards. This is done by using the "**Edit Style**" command from the "**Style**" group's **Ribbon**.

This command will display the selection window where a new style can be assigned to the selected object by picking one from the style library.



By using this method the object will dynamically function in the same way as before but with a new look.



It is also possible to access the style properties through the object's command menu and selecting the "Edit Style" field.

## 7.2. Managing Styles in Movicon Next

Each XAML object can be styled using the 'Style' file. The Style file reference is defined in the XAML object's file. The style XAML object can also be inserted in the Movicon Toolbox or added to the Objects in the Symbol Gallery.

In cases where you wish to use the styled XAML object in the Toolbox, you will need to perform these operations in order to do so:

1. Insert the XAML object's file in the  
C:\ProgramData\Progea\Movicon.NExT\Toolbox\Screen folder.
2. Insert the style .xaml file in the "Resources " folder of the Movicon NeXT installation folder (for example: C:\ProgramData\Progea\Movicon.NExT\Resources).

An example of a reference to the Style file reported in the XAML object's file is:

*E.g.: <TextBox Width="120" Height="35" Style="{DynamicResource TextBoxStyle1}" />*

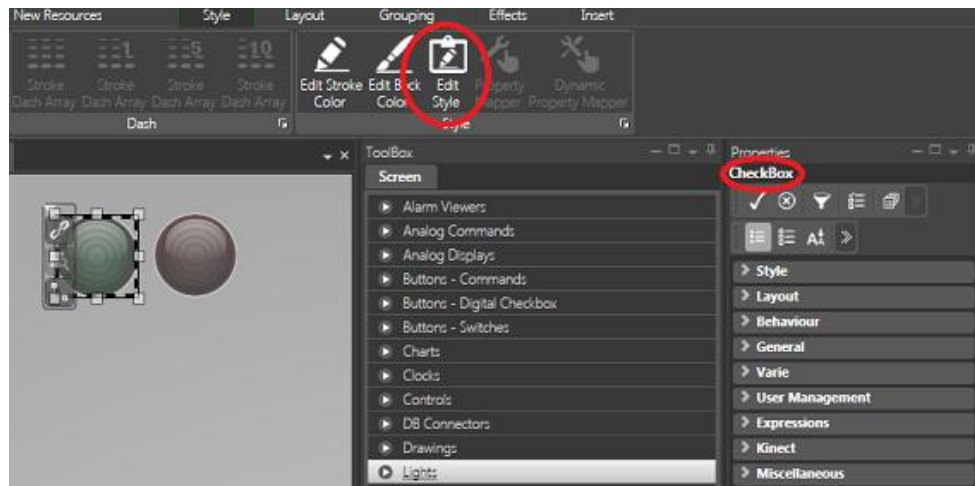
*where the style is defined "TextBoxStyle1" and the file with the style resource is inserted in C:\ProgramData\Progea\Movicon.NExT\Resources*

The "Macro-Categories" for which it is possible to edit the styles by means of using Movicon Next with the appropriate tools provided in Ribbons are:

- Buttons
- CheckBoxes
- ProgressBars
- Sliders



For further info on editing styles in Movicon please refer to the chapter on "Using Styles"



The figure above shows a 'Light' object taken from the Toolbox (belonging to the CheckBox category) with its style being modified using the ribbon indicated.

The use of Styles in Movicon Next can be found in the Symbol Gallery's Digital and Analog tabs. The types of styles found in these windows are:

- **CheckBox** combined with the digital folder (e.g. objects that can change color)
- **ProgressBar** combined with the Analog folder (e.g. objects with filling animation)

The use of Styles in the Toolbox can be found in various types applied to all the object Macro-Categories:

#### 7.1. Buttons

1. CheckBoxes
2. ProgressBars
3. Sliders



When applying styles to objects from the **Symbol Gallery**, Movicon will compare the structure of the C:\ProgramData\Progea\Movicon.NExT\Styles folder with the corresponding C:\ProgramData\Progea\Movicon.NExT\Symbols folder . If the sub folders in the corresponding paths have the same name, Movicon will use that style.

As regards to objects from the **"ToolBox"**, the style is applied by using a 'reference' system "reference" in the object's XAML. For example: **File Reference** (Name Object.XAML.data). Within the path we will have (FilePathObject base@FilePathStyle?StyleName):  
C:\ProgramData\Progea\Movicon.NExT\Styles  
\Button.default@C:\ProgramData\Progea\Movicon.NExT\Styles\Button\Custom  
\EsStyleButton.xaml?EsButtonStyle1



At the moment only some categories enable you to edit their styles despite the fact that potentially all WPF objects should be edited in this way.,

## Add New Styles

It is therefore possible to add new style files directly to the MoviconNExT style installation folder C:\ProgramData\Progea\Movicon.NExT\Styles\. While inserting a style, Movicon searches for .sgdefault file corresponding to the folder in which the Style file resides. The styles defined in the Movicon library are:

- **Buttons** C:\ProgramData\Progea\Movicon.NExT\Styles\Button
- **CheckBoxes** C:\ProgramData\Progea\Movicon.NExT\Styles\CheckBox
- **ProgressBars** C:\ProgramData\Progea\Movicon.NExT\Styles\ProgressBar
- **Sliders** C:\ProgramData\Progea\Movicon.NExT\Styles\Slider

As regards to the ToolBox, the objects that refer to specific styles are characterized by these three files:

1. Image File to make object appear on the toolbox list (ObjectName.png)
2. Xaml File to define object (ObjectName.xaml) Eg: <TextBoxWidth="120"Height="35"Style="{DynamicResource TextBoxStyle1}" />
3. Reference File to reference the object's style (ObjectName.xaml.data)
  - FilePathObject base@FilePathStyle?StileName
  - Eg.:C:\ProgramData\Progea\Movicon.NExT\Styles\Button.default@C:\ProgramData\Progea\Movicon.NExT\Styles\Button\Custom\EsStyleButton.xaml?EsButtonStyle1

## Rules for Creating new Styles

There are a variety of editors that can be used to create Style files starting with the most complex (e.g. Blend), to the less performing ones (e.g. Notepad). However, no matter which one is used it is important that you create the style according to the rules below in order to be supported in MoviconNExT.

### The styles must be included in a ResourceDictionary:

```
<ResourceDictionary
xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
xmlns:d="http://schemas.microsoft.com/expression/blend/2008" mc:Ignorable="d"
xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"> </ResourceDictionary>
```



For further details about creating styles please refer to the Microsoft guide.

A simple example of creating a style would be:

```
<Style x:Key="EsButtonStyle2" TargetType="{x:Type Button}">
<Setter Property="Template">
<Setter.Value>
<ControlTemplate TargetType="{x:Type Button}">
    <Grid x:Name="grid">
    <Viewbox Stretch="Fill">
        <Grid x:Name="modernButton1_tond" Height="90.704"
Width="90.708">
            <Rectangle Tag="Background" x:Name="Back"
Fill="Red" Opacity="1"/>
        </Grid>
    </Viewbox>
    </Grid>
</ControlTemplate>
</Setter.Value>
</Setter>
</Style>
```

```

        <TextBlock Tag="Title" MaxWidth="70" TextWrapping="Wrap"
VerticalAlignment="{Binding VerticalContentAlignment, RelativeSource={RelativeSource
TemplatedParent}}"
HorizontalAlignment="{Binding HorizontalContentAlignment,
RelativeSource={RelativeSource TemplatedParent}}"
Text="{Binding Content, RelativeSource={RelativeSource TemplatedParent}}"
d:IsHidden="True"/>
    </Grid>
</ControlTemplate>
</Setter.Value>
</Setter>
</Style>

```



The Tag="Background" is used to enable objects to be checked in order to apply a background, when being edited from the Style Ribbon (Edit Back Color). The TextBlock can be used for displaying text above the control so that it can then be modified in design mode from the CommonProperties window.

- **Keep lines of code to be inserted to a minimum in order for Movicon to load quickly.** The objects with complex Paths should be optimized (some editors, such as Blend, allow complex paths to be grouped together by using the 'combinepath' command).
- **In order to discriminate style behaviour in cases where the project is being run on a low performing system (such as the WebClient), you can define style behaviours using Triggers instead of VisualStates.** VisualStates establish the object's behaviour at a determined state without offering the possibility to discriminate values of those properties exposed. Alternatively, Triggers offer the possibility to activate determined StoryBoards only when certain conditions occur. Movicon provides a DependencyProperty type that determines whether the object has been loaded in a low performing system or not. The exact property used for this is called **ScreenDocument.RunningOnSlowPCMachine**. In order to reference this property you will need to add the "Namespace" to the ResourceDictionary reference:

```
xmlns:mynsp="clr-namespace:ScreenSettings;assembly=ScreenSettings"
```

Once the ScreenSettings library has been referenced, the trigger can be used in the following way:

```

<Condition
Property="mynsp:ScreenDocument.RunningOnSlowPCMachine" Value="False"/>

```

```

<Condition
Property="mynsp:ScreenDocument.RunningOnSlowPCMachine" Value="True"/>

```

The storyboards to be activated in the Triggers can be defined within the style file's resource session.

#### Example:

```

<Style x:Key="EsButtonStyle3" TargetType="{x:Type Button}">
<Setter Property="Template">
<Setter.Value>

```

```

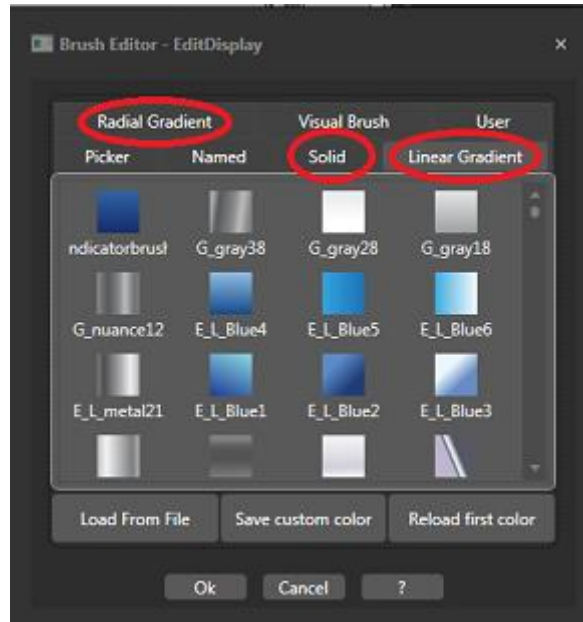
<ControlTemplate TargetType="{x:Type Button}">
<ControlTemplate.Resources>
<Storyboard x:Key="MouseOver">
    <DoubleAnimationUsingKeyFrames
        Storyboard.TargetProperty="(UIElement.Opacity)"
        Storyboard.TargetName="Back">
        <EasingDoubleKeyFrame KeyTime="0" Value="0.75"/>
    </DoubleAnimationUsingKeyFrames>
</Storyboard>
<Storyboard x:Key="NotMouseOver">
    <DoubleAnimationUsingKeyFrames
        Storyboard.TargetProperty="(UIElement.Opacity)"
        Storyboard.TargetName="Back">
        <EasingDoubleKeyFrame KeyTime="0" Value="1"/>
    </DoubleAnimationUsingKeyFrames>
</Storyboard>
</ControlTemplate.Resources>
<ControlTemplate.Triggers>
<MultiTrigger>
    <MultiTrigger.Conditions>
        <Condition Property="IsMouseOver" Value="True"/>
    </MultiTrigger.Conditions>
    <MultiTrigger.EnterActions>
        <BeginStoryboard x:Name="MouseOverStoryboard"
            Storyboard="{StaticResource MouseOver}"/>
    </MultiTrigger.EnterActions>
    <MultiTrigger.ExitActions>
        <StopStoryboard BeginStoryboardName="MouseOverStoryboard"/>
        <BeginStoryboard Storyboard="{StaticResource NotMouseOver}"/>
    </MultiTrigger.ExitActions>
</MultiTrigger>
</ControlTemplate.Triggers>
<Grid x:Name="grid">
    <Viewbox x:Name="viewbox" Stretch="Fill">
        <Viewbox.Effect>
            <DropShadowEffect Opacity="0"/>
        </Viewbox.Effect>
        <Grid x:Name="modernButton1_tond" Height="90.704"
            Width="90.708">
            <Rectangle Tag="Background" x:Name="Back"
                Fill="{DynamicResource E_G_Arcs}" Opacity="1"/>
        </Grid>
    </Viewbox>
    <TextBlock Tag="Title" MaxWidth="70" TextWrapping="Wrap"
        VerticalAlignment="{Binding VerticalContentAlignment, RelativeSource={RelativeSource
        TemplatedParent}}"
        HorizontalAlignment="{Binding HorizontalContentAlignment,
        RelativeSource={RelativeSource TemplatedParent}}"
        Text="{Binding Content, RelativeSource={RelativeSource TemplatedParent}}"
        d:IsHidden="True"/>
    </Grid>
</ControlTemplate>
</Setter.Value>
</Setter>

```



</Style>

- **Use colors (Brush) dynamically**, without defining them in static mode within the control. The most simple way to do this is to use those colors that have been already loaded by MoviconNExT. The list and name of these colors can be found in the Brush settings window that can be accessed from the Style Ribbon for setting BackColors.



*The predefined Movicon colors that can be used dynamically are found in the windows in red*

To use these colors simply use this command:

{DynamicResource COLOR\_NAME} (eg.: {DynamicResource E\_G\_Arcs}).

You can also add new colors to the resource file loaded by MoviconNExT in the C:\ProgramData\Progea\Movicon.NExT\Resources folder:

- LinearGradient (LinearGradientBrush)
- RadialGradient (RadialGradientBrush)
- SolidColor (SolidColorBrush)
- VisualBrush (VisualBrush)
- FillingBrush (DrawingBrush)



Changing the colors in this folder will change the aspect of the entire symbol library.

- **Ultimately, it is always best to use unique keys for Style definitions and Resources such as the Brushed.** The best method to use is the Guid (**G**lobally **U**nique **I**dentifier) as a key. Different tool types can be used for creating GUIDs that permit unique codes to be generated randomly (based on based on data such as date and time of the machine with which the code is generated).

## 7.2. Managing Styles in Movicon.Next from external files

The Font style of the entire application or just some of its screens can be defined using an appropriate file as described below:

### The application's general style file

When creating a **default.style** file in the application's Documents folder (for further details see the example below) you will be able to edit the valid style definitions for the whole application.

### Specific screen style file

When creating a **<screenname>.xaml.style** in the **Screen** folder, you will be able to edit the valid style definitions just for the <screenname> screen only. Style files that are created in this folder will be loaded into the general style folder and their settings will have priority over those in the file default.style file.

**Style file syntax:** the style file syntax is the one used for each xaml resource file.

For example:

```
<ResourceDictionary
xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
```

```
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml">
```

```
<Style TargetType="{x:Type TextBox}" x:Key="MyTextBoxStyle">
```

```
    <Setter Property="FontFamily" Value="Constantia"/>
```

```
    <Setter Property="FontWeight" Value="Bold"/>
```

```
    <Setter Property="FontStyle" Value="Italic"/>
```

```
    <Setter Property="FontSize" Value="48"/>
```

```
    <Setter Property="Foreground" Value="#FFBDEC12"/>
```

</Style>

</ResourceDictionary>

## Associating styles to controls

To associate controls with custom styles, simply insert the style reference directly in each control to consequently create a custom library.

However the custom library with the specific control styles defined in one project will not be recognized by subsequent new projects when run. For example, if you define a specific style in a 'Custom' Toolbox control in one project and then later create a new project using the same control, this control will function with its default style when the new project is run and not the one it was customized with. This is because the new project, in which the control is being used in runtime, is not the one in which the custom library containing the specific style was created in. Therefore associated custom control styles are specific to the project in which they were created and cannot be used automatically by other projects.

In this phase the steps to use the style defined in the files are as follows:

1. Create a folder in the toolbox for the controls to be associated with styles (e.g.: C:\ProgramData\Progea\Movicon.NExT.3.1\Toolbox\Screen\CustomControls)
2. Copy the controls to be styled
3. Edit the controls by adding the style reference (eg. Style="{DynamicResource MyAlarmWindowStyle}")
4. Create the style file to be applied, in the default.style file for example, with the '<screennamescreen>.xml.style' name.

An example of a style definition for the **TextBox**:

### Style File

```
<?xml version="1.0" encoding="utf-8"?>
```

```
<ResourceDictionary
```

```
xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
```

```
xmlns:s="clr-
```

```
namespace:Microsoft.Surface.Presentation.Controls;assembly=Microsoft.Surface.Presentation"
```

```
xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml">
```

```

<Style TargetType="{x:Type TextBox}" x:Key="MyTextBoxStyle">

  <Setter Property="FontFamily" Value="Gill Sans" />

  <Setter Property="FontWeight" Value="Bold" />

  <Setter Property="FontStyle" Value="Italic" />

  <Setter Property="FontSize" Value="10" />

  <Setter Property="Foreground" Value="#FFBDEC12" />

  <Setter Property="Background" Value="#FFFAEBD7" />

</Style>

</ResourceDictionary>

```

## Text Control

```

<?xml version="1.0" encoding="utf-8"?>

<TextBox xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml" Width="100" Height="30"
MinWidth="5" MinHeight="5" BorderThickness="0" Background="{x:Null}"
IsReadOnly="True" IsHitTestVisible="False" Text="Text" Style="{DynamicResource
MyTextBoxStyle}" />

```

Example of a style definition for the **AlarmWindow**:

## Style File

```

<?xml version="1.0" encoding="utf-8"?>

<ResourceDictionary

  xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

  xmlns:s="clr-
namespace:Microsoft.Surface.Presentation.Controls;assembly=Microsoft.Surface.Presentation"

  xmlns:aw="clr-namespace:AlarmWindow;assembly=AlarmWindow"

  xmlns:converters="clr-namespace:Converters;assembly=Converters"

```

```
xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml">

    <converters:FontSettings x:Key="rowFontSettings">

        <converters:FontSettings.FontFamily>

            <FontFamily>Gill Sans</FontFamily>

        </converters:FontSettings.FontFamily>

        <converters:FontSettings.FontSize>

            10

        </converters:FontSettings.FontSize>

        <converters:FontSettings.FontStyle>

            Normal

        </converters:FontSettings.FontStyle>

        <converters:FontSettings.FontWeight>

            Normal

        </converters:FontSettings.FontWeight>

    </converters:FontSettings>

    <converters:FontSettings x:Key="headerFontSettings">

        <converters:FontSettings.FontFamily>

            <FontFamily>SketchFlow Print</FontFamily>

        </converters:FontSettings.FontFamily>

        <converters:FontSettings.FontSize>

            25

        </converters:FontSettings.FontSize>

        <converters:FontSettings.FontStyle>

            Normal

        </converters:FontSettings.FontStyle>

        <converters:FontSettings.FontWeight>

            Normal
```

```

        </converters:FontSettings.FontWeight>

    </converters:FontSettings>

    <Style TargetType="{x:Type aw:GridAlarmWindow}" x:Key="MyAlarmWindowStyle">

        <Setter Property="AlarmAreaFontSettings" Value="{StaticResource rowFontSettings}"
    />

        <Setter Property="HeaderFontSettings" Value="{StaticResource headerFontSettings}"
    />

    </Style>

</ResourceDictionary>

```

## AlarmWindowControl

```

<?xml version="1.0" encoding="utf-8"?>

<aw:GridAlarmWindow
xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml" xmlns:aw="clr-
namespace:AlarmWindow;assembly=AlarmWindow" Width="600" Height="300"
Tag="ProblematicXamlWriter" Style="{DynamicResource MyAlarmWindowStyle}">

    <aw:GridAlarmWindow.Resources>

        <Style TargetType="{x:Type ListView}" x:Key="AlarmDataSource"> </Style>

    </aw:GridAlarmWindow.Resources>

</aw:GridAlarmWindow>

```



In order to setup references to controls in the style definition phase, they must be inserted in the controls' name spaces. For example, the alarm window name space is:

```
xmlns:aw="clr-namespace:AlarmWindow;assembly=AlarmWindow"
```

Once this has been done, the target type in the style will be:

```
TargetType="{x:Type aw:GridAlarmWindow}"
```

## 8. Dynamics Animation

### 8.1. Dynamic Animation Settings

The Movicon Object and Symbol Dynamic Animations, which concerns the association of variables (Tags) to determine graphical animations during Runtime, can be set through the "Dynamic Animation Settings" or "Animation Explorer" window.

The name of the reference tag can be defined for each Dynamic Animation using the "**Tag**" property. In cases where Tags have not been defined for each animation, the **reference Tag** associated to the object will be used (see topic regarding "Assigning Tags to Objects").

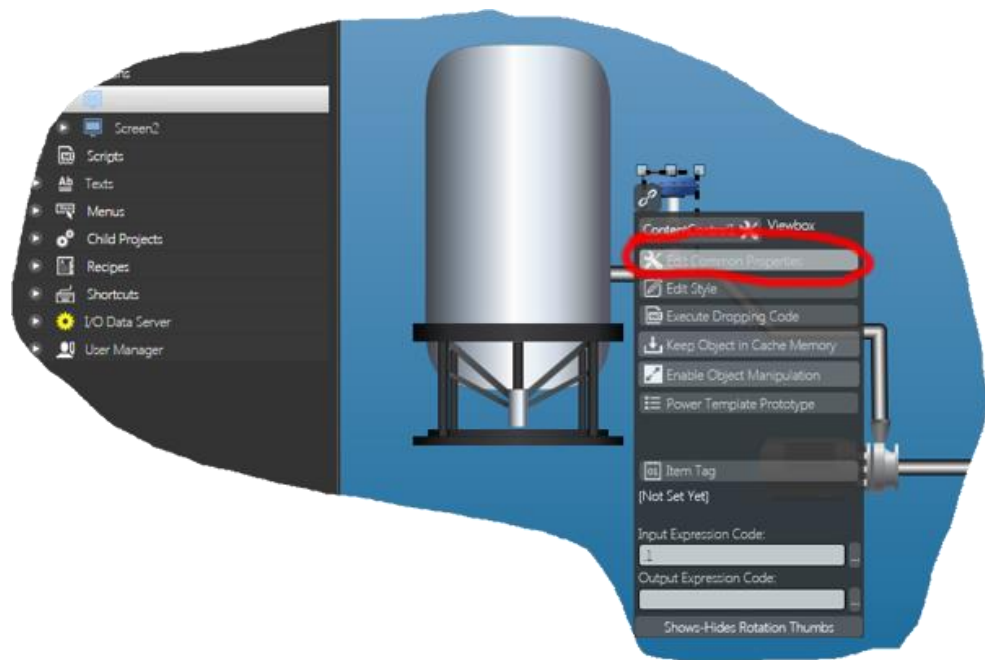


Dynamic animations can be set for multi-object selections. For further information about this, please refer to the topic on Command Assignments for Multi-Selections.

#### Animation Settings Window

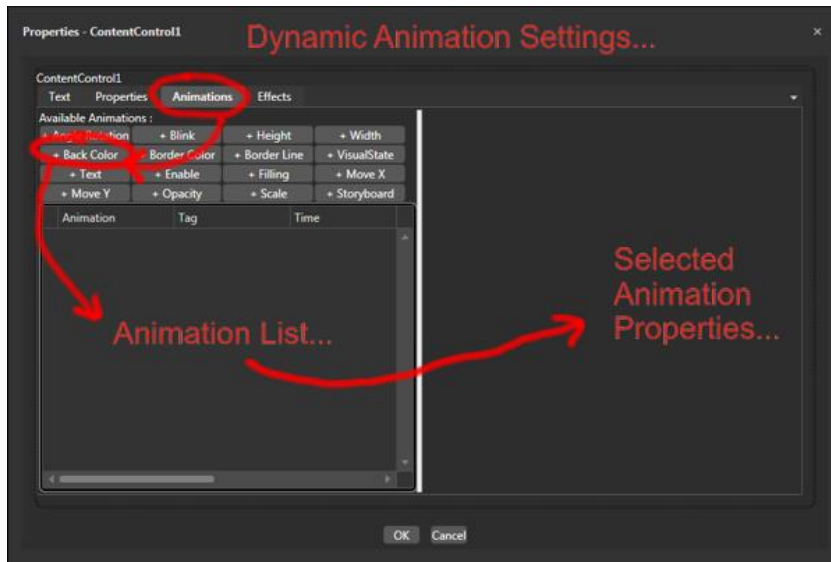
The animation settings window can be accessed through the object's command Tooltip or by double clicking on the object. This window will open as a modal type window. To configure an object's animations proceed as follows:

1. Open a screen and select the graphical object desired.
2. Activate the command Tooltip using the usual small button and then click on the "**Edit Common Properties**" item.



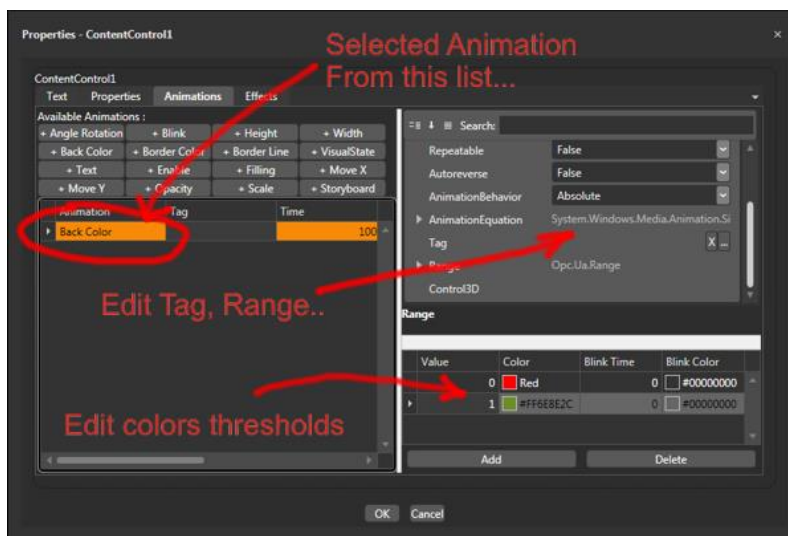
*The Animation Settings Window opens when clicking the "Common Property Editor" button from the object's Tooltip.*

3. The "Common Property Editor" window has a selection of tabs containing different types of settings. To access the animation settings, select the **"Animations"** Tab.
4. The list of possible animations (Angle Rotation, Blink, Back Color, etc.) are shown at the top of the **"Animations"** window. Click on the name of the animation you wish to insert for the object. The animation will be added to the list ready for configuring. Various animations can be added to one object.



Animation Settings Window

5. After having added the Animation desired, you can now proceed in configuring its parameters by double clicking the row on the list to update the window with the selected Animation's parameters. For instance, if you wish to create an animated back color, you will need to select the "Back Color" from the list of animations, as shown in the image below. Note that you can also insert the same command several times if wishing to animate the back color in different ways with different Tags.



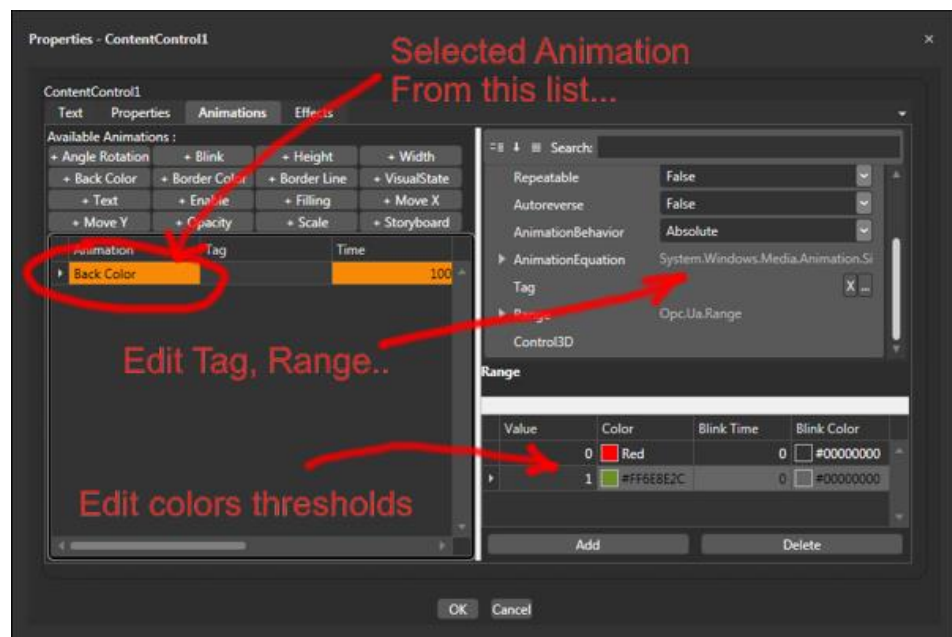
Window showing Animation parameters.



- Some of the parameters of the various Animations remain the same while others change according to the Animation type. Once the Animation parameters have been configured, close the window with OK to confirm settings.

## 8.2. Dynamic Animation Properties

All of the dynamic animation functions have the same set of common properties as indicated below:



### Execution

**Apply to Child Projects:** When enabled this property applies the animation to all the symbol's child object components.

**Expression:** Allows you to set the expression whose result will be used in the animation.



Certain syntaxes can be used in this property to obtain the following:

- to connect several variable bits insert the '.nBit' value (eg. .1 to indicate the Bit 1)
- to connect several array elements, insert the '[nElem]' value

(eg. [1] to indicate item 1)

**Repeatable:** Enables animation repetition. When activated the performed animation will be repeated cyclically until condition is deactivated.

**Auto-Reverse:** Enables the animation in the reverse.

**Animation Behaviour:** Defines the animation's behaviour according to these possible selections

- **Absolute:** the variable value corresponds to the rotation angle value
- **Proportional:** the max range value of the variable is proportional to the animation's end value
- **Trigger:** the variable value is managed as a "trigger", therefore a  $\geq 1$  value will set the animation to the end value.

### General

**Animation Time:** This is used to define the dynamic animation time coefficient (in milliseconds) so that it flows better.

**Animation Tag:** This is used to define the Animation Tag using the browser window of those tags inserted in the Data Server's Address Space.

**Value Range:** This is used to define the variable value range with which the animation will be executed.

### Thresholds

The threshold setting will be made available when needed to establish animation behavior in function with the value to be assumed by the associated variable. Generally the thresholds can be associated with a different color for each value desired for the associated variable. Therefore, the value = 0 can be set with one color, while value = 1 will activate another color that will remain activated until the color set for the next threshold is activated and so forth.

<b>Value</b>	This is used to define the absolute numeric value that once reached will activate the color or the threshold function.
<b>Color</b>	This is used to select a color that the object will obtain when the threshold value has been reached. The chosen color will remain active until the next set threshold value has been reached.
<b>Blink Time</b>	This is used to define a blink time in milliseconds between the color set for the threshold and the color set for the blink. The value is = 0 (default) means that the blink function is not active.
<b>Blink Color</b>	This is used to select a blink color which will be used as a alternating color with the one set for the threshold if the blink time has been set.
<b>Text</b>	This is used to select the text type to be entered.



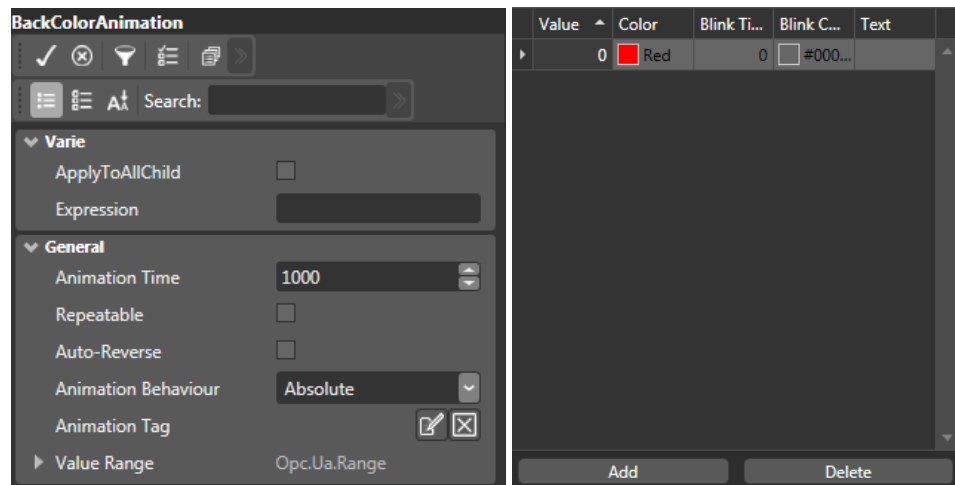
The **ADD** and **DELETE** commands are used to add or remove thresholds from the associated animation function. Since the number of thresholds that can be inserted is technologically restricted, the system can guarantee a maximum number of 32 thresholds for each dynamic function.

## 8.3. Dynamic Animation Functions list

### 8.4. Back Color Animation

The Dynamic "Back Color" Animation command is used for assigning a variable to the object to determine the coloring of the object's background. This command requires that the relative configuration parameters be defined along with colors relating to the value thresholds.

The function's behaviour is conditioned by the setting of the properties described in "Dynamic Animation Properties".



### Back Color Animation Properties

#### Execution

**Apply to Child Objects:** When enabled, this animation will be propagated to all child objects composing the symbol.

**Expression:** Used to set the expression whose result will be used in the animation "Dynamic Animation Properties".

**Repeatable:** This is used for enabling/disabling animation repetition. When enabled the animation will be repeated in a loop while Animation Condition remains active "Dynamic Animation Properties"

**Auto-Reverse:** This is used for enabling/disabling the auto-reverse. When enabled the animation will reverse in a loop until the "Animation Condition" remains active **Dynamic Animation Properties**".

**Animation Behaviour:** Sets the animation mode in respect to the value range entered in the "Value Range" property for the TAG connected to the animation "Dynamic Animation Properties".

#### General

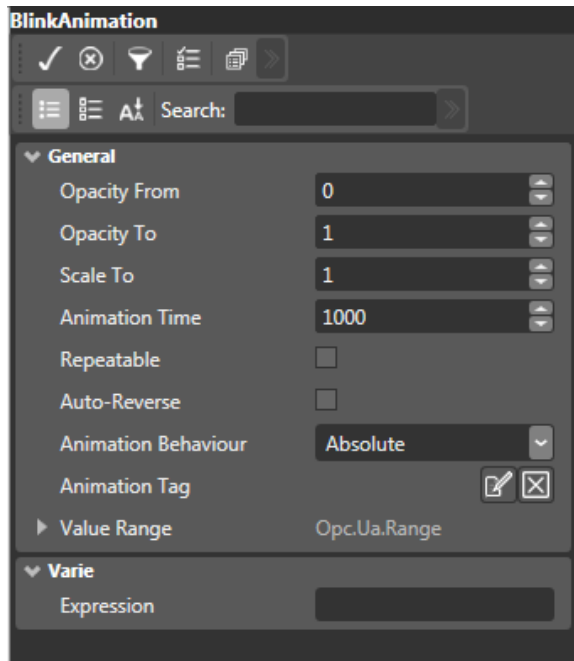
**Animation Time:** Sets how long the Animation lasts "**Dynamic Animation - Animation Time Properties**".

**Animation Tag:** Sets the variable to associate to the animation "Dynamic Animation Properties".

**Value Range:** Sets the value range allowed for managing the animation "Dynamic Animation Properties"

## 8.5. Blink Animation

The Dynamic "**Blink**" Animation command is used for assigning a variable to the object to determine visual blink animation by managing a scaled zoom and a transitory transparency value according to the following parameters:



### Blink Animation Properties

#### General

**Transparency From:** Defines the object's minimum transparency value where the value = 0 means "invisible"

**Transparency To:** Defines the object's maximum transparency value where the value = 1 means "fully visible"

**Scale To:** Used for defining a scale value during blink where the value = 1 (default) refers to the object's natural scale size (100%).

**Animation Time:** Sets how long the Rotation Animation lasts "**Dynamic Animation - Animation Time Properties**".

**Animation Tag:** Sets the variable to associate to the animation "Dynamic Animation Properties"

**Value Range:** Sets the value range allowed for managing the animation "Dynamic Animation Properties"

#### Execution

**Expression:** Used to set the expression whose result will be used in the animation "Dynamic Animation Properties".

**Repeatable:** This is used for enabling/disabling animation repetition. When enabled the animation will be repeated in a loop while Animation Condition remains active: "Dynamic Animation Properties"

**Auto-Reverse:** This is used for enabling/disabling the auto-reverse. When enabled the animation will reverse in a loop until the "Animation Condition" remains active: "**Dynamic Animation Properties**".

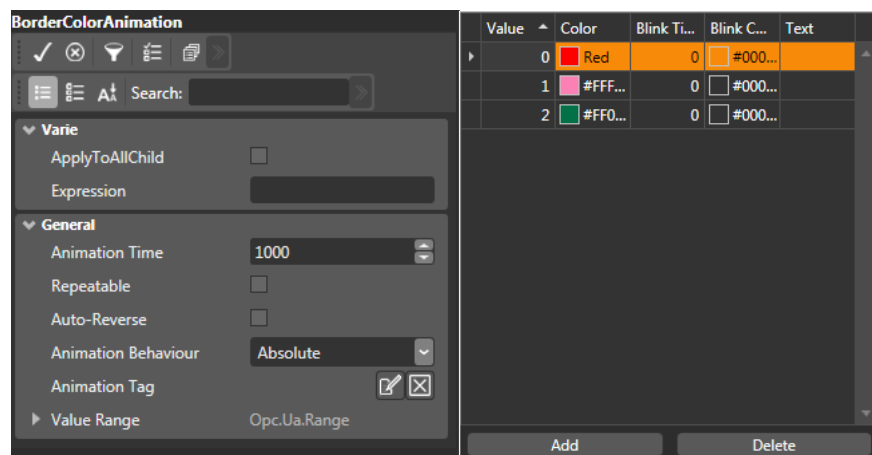
**Animation Behaviour:** Sets the animation's behaviour mode management mode in respect to the low and high values inserted in the 'Value Range' property for the TAG connected to the animation: "Dynamic Animation Properties". The value options are:

- **Trigger:** if the reference Tag is set to zero, the object is visibly fixed. If the tag is set other than zero, the object will blink the minimum and maximum transparency value.
- **Absolute:** the object will continuously blink between the minimum transparency value and the reference Tag value. In this case, if the transparency is set as Min=0 and Max=1, when the reference tag is set with a zero value the object will blink with an transparency between 0 and 0 and will result invisible. When the Tag is set to 1, the object will blink with an transparency between 0 and 1. When the Tag is set to 0.5, the object will blink with an transparency between 0 and 0.5.
- **Proportional:** the object will continuously blink between the minimum transparency value and the and the reference Tag's scaled value. The scale factor is applied between the set tag range, for example 0-100, and the transparency range, for example 0-1. In this case, if the transparency is set with a Min=0 and Max=1, when the reference Tag is set with a zero value, the object will blink with an transparency between 0 and 0 and therefore result invisible. When the tag is equal to 100, the object will blink with an transparency between 0 and 1. However, when the Tag is set to 50, the object will blink with an transparency between 0 and 0.5

## 8.6. Border Color Animation

The Dynamic "**Border Color**" Animation command is used for assigning a variable to the object to determine the color of the object's borders. This command requires that the relative configuration parameters be defined along with the colors relating to the value thresholds.

The function's behaviour is conditioned by the property settings described in "Dynamic Animation Properties".



## Border Color Animation Properties

### Execution

**Apply to Child Objects:** When enabling this property the animation will be applied to the symbol's child object.

**Expression:** Used to set the expression whose result will be used in the animation "Dynamic Animation Properties".

**Repeatable:** This is used for enabling/disabling animation repetition. When enabled the animation will be repeated in a loop while Animation Condition remains active "Dynamic Animation Properties"

**Auto-Reverse:** This is used for enabling/disabling the auto-reverse. When enabled the animation will reverse in a loop until the "Animation Condition" remains active **Dynamic Animation Properties**".

**Animation Behaviour:** Sets the animation mode in respect to the value range entered in the "Value Range" property for the TAG connected to the animation "Dynamic Animation Properties".

### General

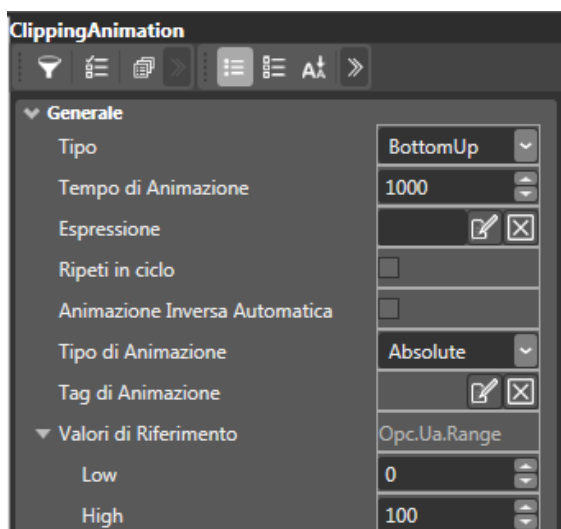
**Animation Time:** Sets how long the Animation lasts "**Dynamic Animation - Animation Time Properties**".

**Animation Tag:** Sets the variable to be associated to the animation "Dynamic Animation Properties".

**Value Range:** Sets the value range allowed for managing the animation "Dynamic Animation Properties"

## 8.7. Clipping Animation

The dynamic "Clipping" animation command is used to selectively display parts of the symbol according to the parameters set for this animation type.



## Clipping Animation Properties

### General

**Clipping Type:** Sets the clipping animation type to be applied.

**Animation Time:** Sets how long the Animation lasts "**Dynamic Animation - Animation Time Properties**".

**Animation Tag:** Sets the variable to associate to the animation "Dynamic Animation Properties"

**Value Range:** Sets the value range allowed for managing the animation "Dynamic Animation Properties"

### Execution

**Expression:** Used to set the expression whose result will be used in the animation "Dynamic Animation Properties".

**Repeatable:** This is used for enabling/disabling animation repetition. When enabled the animation will be repeated in a loop while Animation Condition remains active "Dynamic Animation Properties"

**Auto-Reverse:** This is used for enabling/disabling the auto-reverse. When enabled the animation will reverse in a loop until the "Animation Condition" remains active **Dynamic Animation Properties**".

**Animation Behaviour:** Sets the animation mode in respect to the value range entered in the "Value Range" property for the TAG connected to the animation "Dynamic Animation Properties".

## 8.8. Composed Movement Animation

The Dynamic "**Composed Movement**" animation is used to determine animated movement along a trajectory which can be composed as pleased by using a "Polyline", "Polygon" or "Line" object. The trajectory can be defined as needed on screen by graphically drawing a path. It can also be set with simultaneous X and Y movements to create diagonal movements for example.

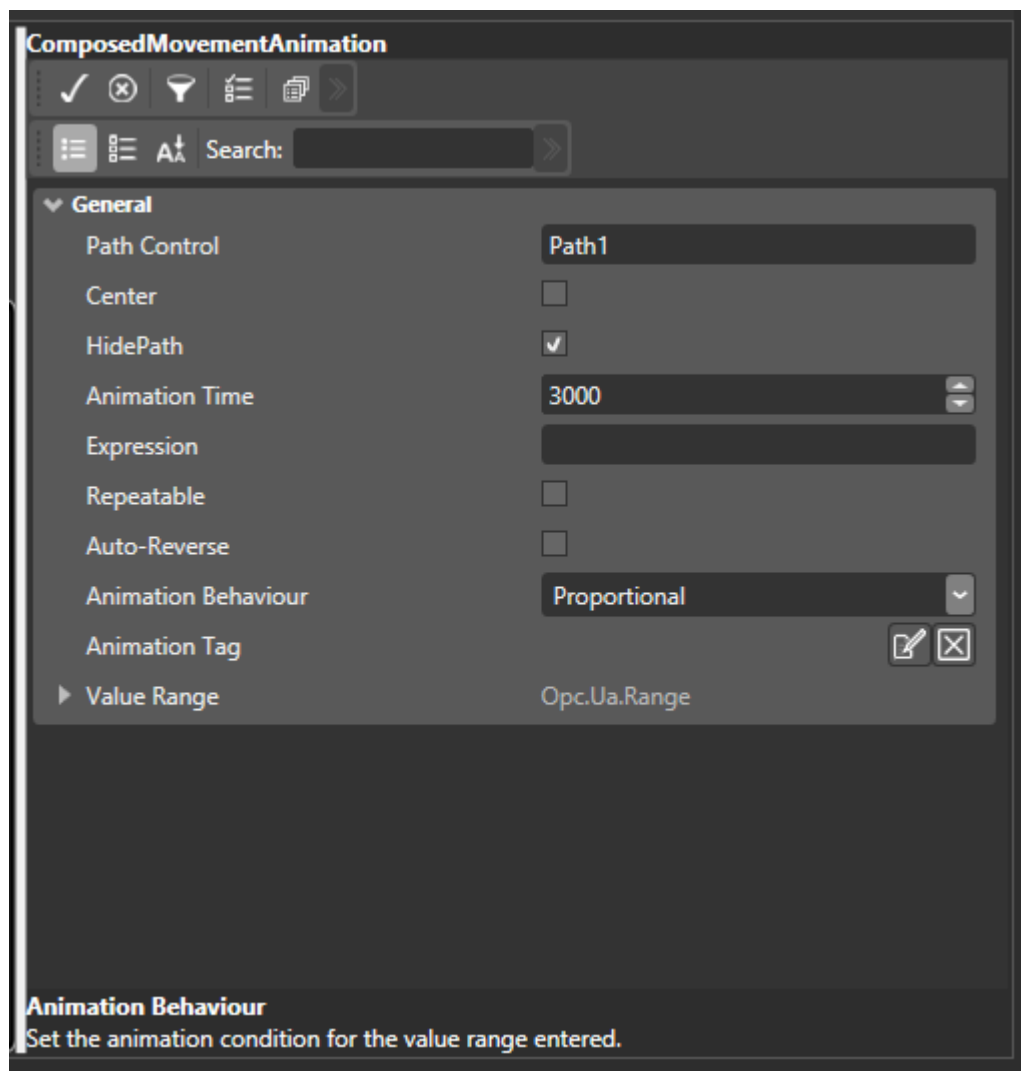


The use of this animation feature requires the user to insert a "Polyline", "Polygon" or "Line" element on screen from the Toolbox. Each point of the inserted polygon builds a path for the object or symbol to follow. The 'name' assigned to the "Polyline", "Polygon" or "Line" objects must therefore be specified in the dynamic movement function.

The "Polyline", "Polygon" or "Line" used as the path can be made visible during runtime by enabling the corresponding animation property provided for this purpose.

Other elements from the 'Draws' category (eg. rectangle, Ellipse) used for defining the trajectory will have no effect on the symbol's animation which will remain in its position without moving during runtime.

Composed Movement animation are set using the following parameters:



## Composed Movement Animation Properties

### General

**Path Control:** Defines the name of the "Poly-Line" to be used as the trajectory path for the object to follow. Therefore, you will need to have already inserted a "Poly-Line" from the toolbox with set trajectory path points for the object to follow as well as a name so that it can be entered here.

**Hide Path:** This is used for hiding the trajectory path to be used by the polyline during project runtime.

**Animation Time:** Sets how long the Animation lasts "**Dynamic Animation - Animation Time Properties**".

**Animation Tag:** Sets the variable to associate to the animation "**Dynamic Animation Properties**"

**Value Range:** Sets the value range allowed for managing the animation "**Dynamic Animation Properties**"

### Execution

**Use Barycenter:** This is used to indicate whether the movement reference point, in respect to the object, must be related to the object's centre. If not selected, the reference point will be taken from the top right angle.



**Expression:** Used to set the expression whose result will be used in the animation "Dynamic Animation Properties".

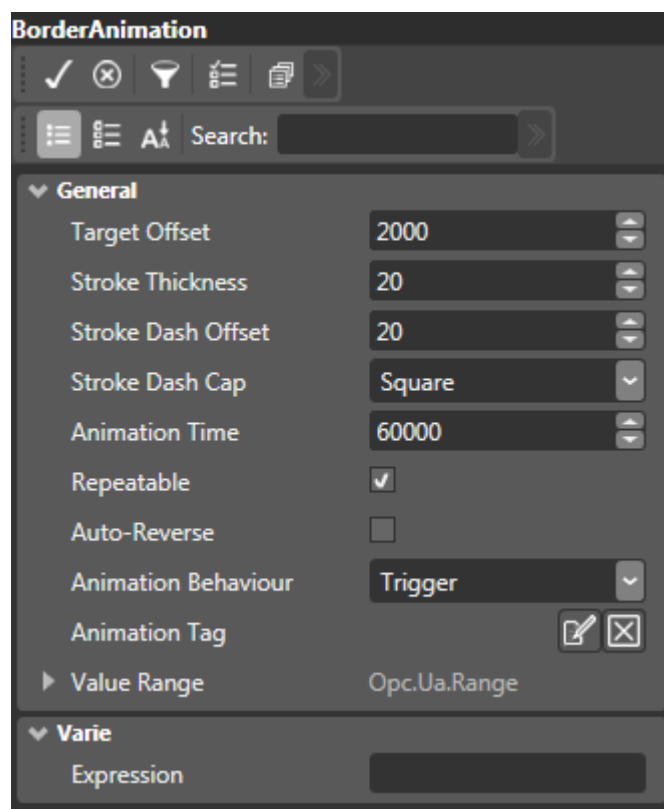
**Repeatable:** This is used for enabling/disabling animation repetition. When enabled the animation will be repeated in a loop while Animation Condition remains active "Dynamic Animation Properties"

**Auto-Reverse:** This is used for enabling/disabling the auto-reverse. When enabled the animation will reverse in a loop until the "Animation Condition" remains active **Dynamic Animation Properties**".

**Animation Behaviour:** Sets the animation mode in respect to the value range entered in the "Value Range" property for the TAG connected to the animation "Dynamic Animation Properties".

## 8.9. Dashed Line

The Dynamic "**Border Line**" Animation command is used for assigning a variable to determine a dynamic dash stroke line with animated movement to frame objects such as the line or polyline objects, according to the following parameters:



### Dynamic Border Animation Properties

#### General

**Tag:** Used to set the variable to associate to the animation "Dynamic Animation Properties"

**Animation Time:** Used to set the Animation's duration "**Dynamic Animation - Animation Time Properties**".

**Target Offset:** Defines the velocity factor of the animation's motion path. A negative value changes the direction of the rotation.

**Expression:** Used to set the expression whose result will be used in the animation "Dynamic Animation Properties".

**Repeatable:** This is used for enabling/disabling animation repetition. When enabled the animation will be repeated in a loop while Animation Condition remains active "Dynamic Animation Properties"

**Auto-Reverse:** This is used for enabling/disabling the auto-reverse. When enabled the animation will reverse in a loop until the "Animation Condition" remains active **Dynamic Animation Properties**".

**Animation Behaviour:** Used to set the animation mode in respect to the value range entered in the "Value Range" property for the TAG connected to the animation "Dynamic Animation Properties".

**Value Range:** Used to set the value range allowed for managing the animation "Dynamic Animation Properties"

#### Stroke Details

**Stroke Thickness:** Defines the thickness of the border line in pixels.

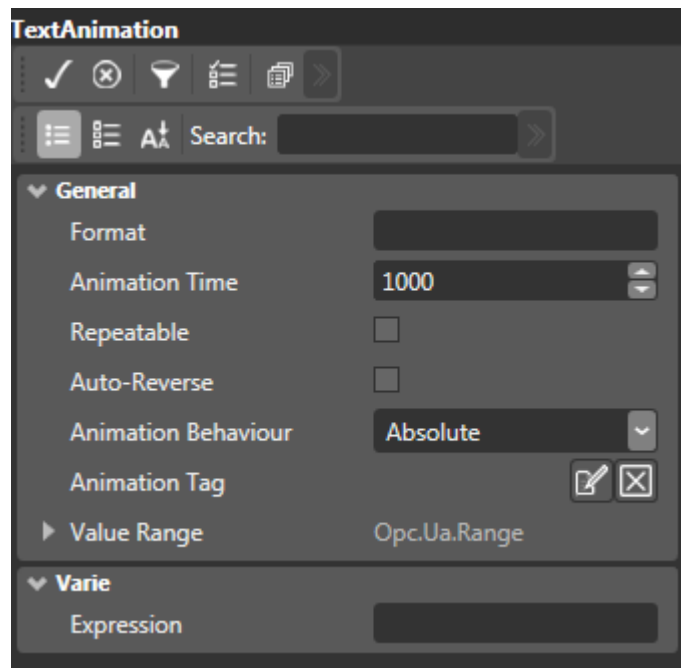
**Stroke Dash Array:** Used to set the line type to be displayed when animation is active. In addition to the options proposed by the drop down menu, the user can also set the displayed line type using an array of values (eg. 1,2,1,3 ecc.). In this case, the first value which corresponds to the 0 index, specifies the dash length, the second value which corresponds to the 1 index, specifies the gap length between the dashes, the third, if present, will specify the dash length again and so on.

**Stroke Dash Offset:** Used to set define the gap between stroke dashes.

**Stroke Dash Cap:** This is used to select the shape type at the start and end of the trajectory line. The options are Round, Square, Triangle.

## 8.10. Dynamic Text Animation

The Dynamic "**Text**" command is used for activating the "Change Text" function in the selected object. This will allow the component to change the text displayed in runtime in function with the associated variable value.



## Dynamic Text Animation Properties

### General

**Format:** This is used to set the text string format.

**Animation Time:** This is used to set Animation's duration "**Dynamic Animation - Animation Time Properties**".

**Animation Tag:** This is used to set the variable to associate to the animation "**Dynamic Animation Properties**"

**Value Range:** This is used to set the value range allowed for managing the animation "**Dynamic Animation Properties**"

### Execution

**Expression:** This is used to set the expression whose result will be used in the animation "**Dynamic Animation Properties**".

**Repeatable:** This is used to enable/disable animation repetition. When enabled the animation will be repeated in a loop while Animation Condition remains active "**Dynamic Animation Properties**"

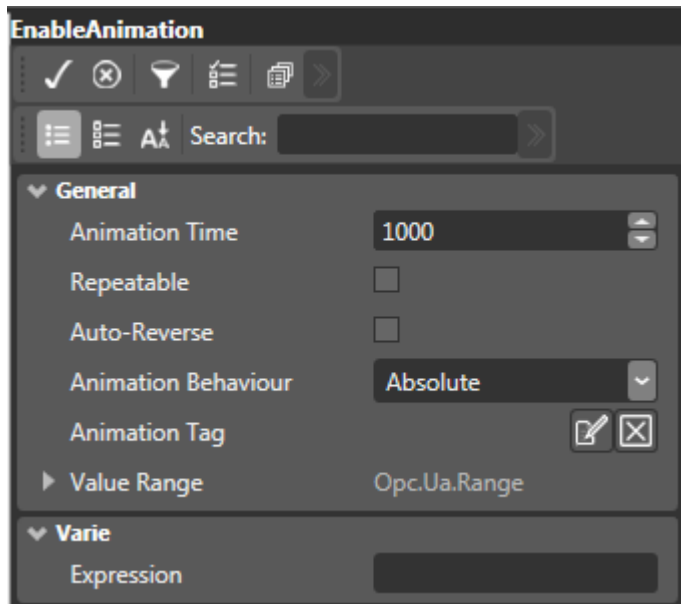
**Auto-Reverse:** This is used to enable/disable the auto-reverse. When enabled the animation will reverse in a loop until the "Animation Condition" remains active "**Dynamic Animation Properties**".

**Animation Behaviour:** This is used to set the animation mode in respect to the value range entered in the "Value Range" property for the TAG connected to the animation "**Dynamic Animation Properties**".

## 8.11. Enable Animation

The Dynamic "**Enable**" Animation command is used for assigning a variable to the object so that it can be enabled for activation or manipulation. This type of animation is referred to the objects that invoke commands (e.g. buttons).

The associated variable determines whether to enable or not to enable the object's activation to then permit the execution of any associated commands.



### Enable Animation Properties

#### General

**Animation Time:** Sets how long the Rotation Animation lasts "**Dynamic Animation - Animation Time Properties**".

**Animation Tag:** Sets the variable to associate to the animation "Dynamic Animation Properties".

**Value Range:** Sets the value range allowed for managing the animation "Dynamic Animation Properties"

#### Execution

**Expression:** This is used to set the expression whose result will be used in the animation "Dynamic Animation Properties".

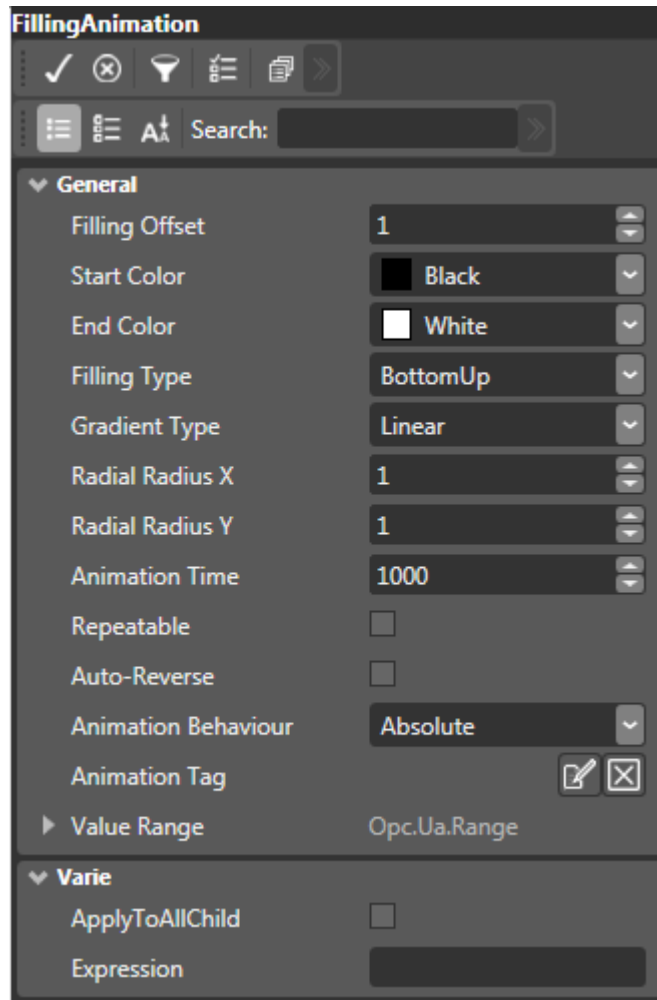
**Repeatable:** This is used for enabling/disabling animation repetition. When enabled the animation will be repeated in a loop while Animation Condition remains active "Dynamic Animation Properties"

**Auto-Reverse:** This is used for enabling/disabling the auto-reverse. When enabled the animation will reverse in a loop until the "Animation Condition" remains active **Dynamic Animation Properties**".

**Animation Behaviour:** This is used to set the animation mode in respect to the value range entered in the "Value Range" property for the TAG connected to the animation "Dynamic Animation Properties".

## 8.12. Filling Animation

The Dynamic Filling Animation command is used for assigning a variable to the object to determine the dynamic filling of objects such as polygons according to the following parameters:



### Filling Animation Properties

#### Fill Details

**Filling Offset:** This is used for defining an "offset" to the gradient color factor value (1 = Default value)

**Gradient Type:** This is used for selecting which type of gradient to use between linear or radial for the colour filling.

**Radial Radius X:** This is used for defining the Radius X value when selecting the radial gradient type.

**Radial Radius Y:** This is used for defining the Radius X value when selecting the radial gradient type.

#### General

**Start Color:** This is used for selecting a starting color to be used in the filling process

**End Color:** This is used for selecting a color to end the filling process.

**Filling Type:** This is used for selecting the filling animation type to use.

**Animation Time:** This is used to set the animation duration "**Dynamic Animation - Animation Time Properties**".

**Animation Tag:** This is used to set the variable to associate to the animation "Dynamic Animation Properties"

**Value Range:** This is used to set the value range allowed for managing the animation "Dynamic Animation Properties"

## Execution

**Expression:** This is used to set the expression whose result will be used in the animation "Dynamic Animation Properties".

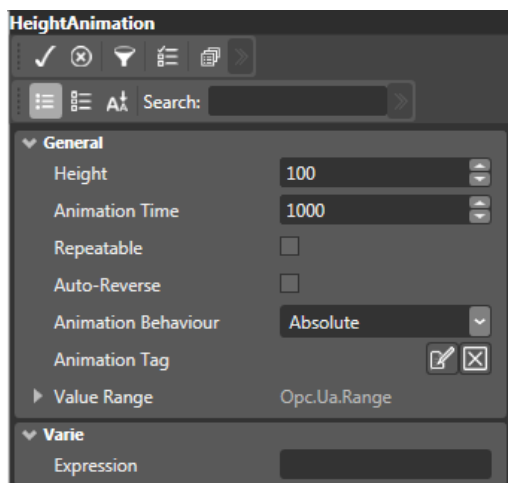
**Repeatable:** This is used for enabling/disabling animation repetition. When enabled the animation will be repeated in a loop while Animation Condition remains active "Dynamic Animation Properties"

**Auto-Reverse:** This is used for enabling/disabling the auto-reverse. When enabled the animation will reverse in a loop until the "Animation Condition" remains active **Dynamic Animation Properties**".

**Animation Behaviour:** This is used to set the animation mode in respect to the value range entered in the "Value Range" property for the TAG connected to the animation "Dynamic Animation Properties".

## 8.13. Height Animation

The "**Height**" Animation command permits a variable to be assigned to the object to determine the object's height size (in pixels) according to the following parameters:



### Height Animation Properties

#### General

**Height:** Defines the value in pixels of the height size to reach. This function's behaviour is conditioned by the property settings described in paragraph: "Dynamic Animation Properties".

**Animation Time:** Sets how long the Animation lasts "**Dynamic Animation - Animation Time Properties**".

**Animation Tag:** Sets the variable to associate to the animation "Dynamic Animation Properties"

**Value Range:** Sets the value range allowed for managing the animation "Dynamic Animation Properties"

### Execution

**Expression:** Used to set the expression whose result will be used in the animation "Dynamic Animation Properties".

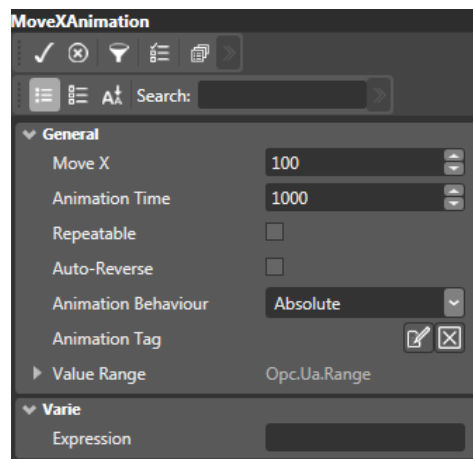
**Repeatable:** This is used for enabling/disabling animation repetition. When enabled the animation will be repeated in a loop while Animation Condition remains active "Dynamic Animation Properties"

**Auto-Reverse:** This is used for enabling/disabling the auto-reverse. When enabled the animation will reverse in a loop until the "Animation Condition" remains active **Dynamic Animation Properties**".

**Animation Behaviour:** Sets the animation mode in respect to the value range entered in the "Value Range" property for the TAG connected to the animation "Dynamic Animation Properties".

## 8.14. Horizontal X Movement Animation

The "**Move X**" Dynamic Animation command is used for assigning a variable to the object to determine the object's position using a horizontal movement along the X axis according to the following parameters:



### Move X Animation Properties

#### General

**Horizontal Movement value:** Sets the value in pixels of the object's movement on the Y axis in respect to its initial point. This behaviour is conditioned by the property settings described in the "Dynamic Animation Properties" topic.

**Animation Time:** Sets how long the Animation lasts "**Dynamic Animation - Animation Time Properties**".

**Animation Tag:** Sets the variable to associate to the animation "Dynamic Animation Properties"

**Value Range:** Sets the value range allowed for managing the animation "Dynamic Animation Properties"

#### Execution

**Expression:** Used to set the expression whose result will be used in the animation "Dynamic Animation Properties".

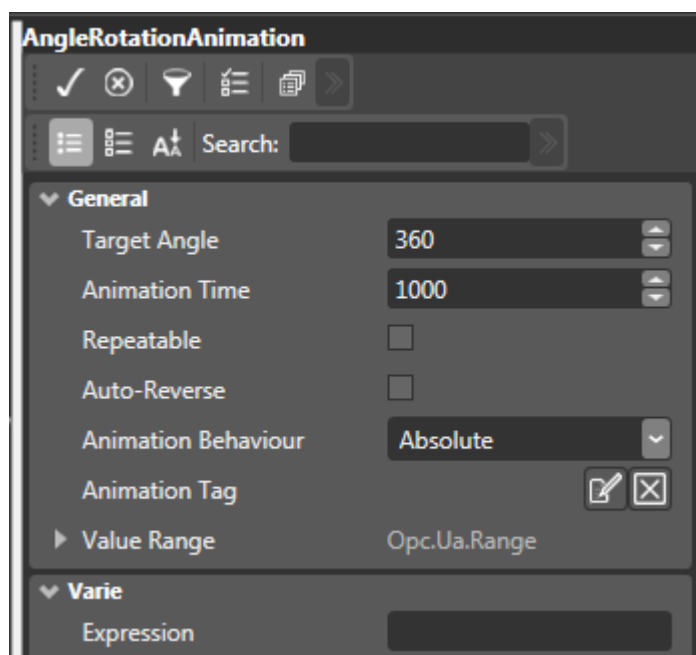
**Repeatable:** This is used for enabling/disabling animation repetition. When enabled the animation will be repeated in a loop while Animation Condition remains active "Dynamic Animation Properties"

**Auto-Reverse:** This is used for enabling/disabling the auto-reverse. When enabled the animation will reverse in a loop until the "Animation Condition" remains active **Dynamic Animation Properties**".

**Animation Behaviour:** Sets the animation mode in respect to the value range entered in the "Value Range" property for the TAG connected to the animation "Dynamic Animation Properties".

## 8.15. Rotation Animation

The Dynamic "**Angle Rotation**" Animation command is used for assigning a variable to the object to determine the angle rotation based on its centre of gravity according to the following parameters:





## Angle Rotation Animation Properties

### General

**Angle Rotation:** Sets the value of the angle position to be reached. This function's behaviour is conditioned by the property settings described in the "**Dynamic Animation Properties**" topic.

**Animation Time:** Sets how long the Animation lasts "**Dynamic Animation - Animation Time Properties**".

**Animation Tag:** Sets the variable to associate to the animation "Dynamic Animation Properties"

**Value Range:** Sets the value range allowed for managing the animation "Dynamic Animation Properties"

### Execution

**Expression:** Used to set the expression whose result will be used in the animation "Dynamic Animation Properties".

**Repeatable:** This is used for enabling/disabling animation repetition. When enabled the animation will be repeated in a loop while Animation Condition remains active "Dynamic Animation Properties"

**Auto-Reverse:** This is used for enabling/disabling the auto-reverse. When enabled the animation will reverse in a loop until the "Animation Condition" remains active **Dynamic Animation Properties**".

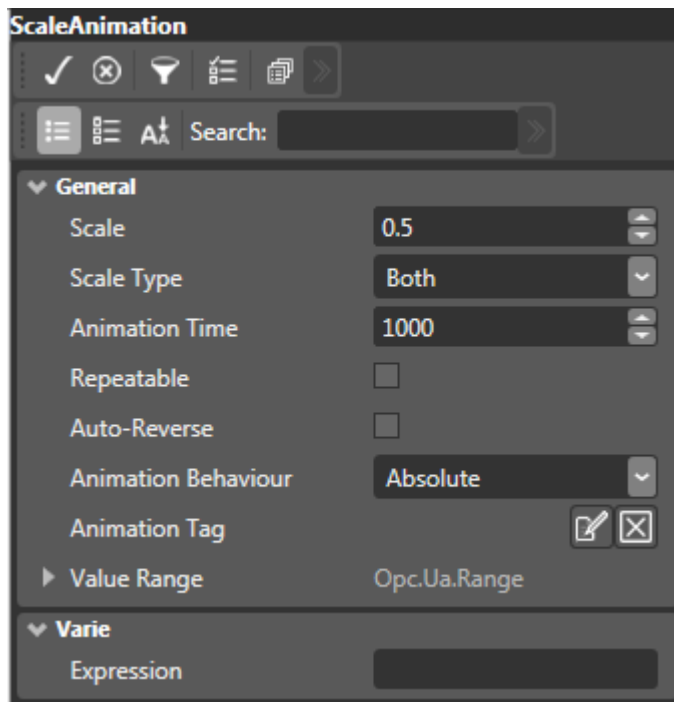
**Animation Behaviour:** Sets the animation mode in respect to the value range entered in the "Value Range" property for the TAG connected to the animation "Dynamic Animation Properties".



The default rotation barycenter is in the center of the object but can be changed when editing the graphics by activating the display of the barycenter's dragging point using the Rotate item from the object's command menu.

## 8.16. Scale Animation

The Dynamic "**Scale**" Animation command is used for assigning a variable to the object to determine the scale size percentage according to the following parameters:



## Scale Animation Properties

### General

**Scale Factor Value:** Sets the percentage rate of the scale's size in respect to its original size. The value = 1 represents the object's original size. This function's behaviour is conditioned by the property settings described in the "Dynamic Animation Properties" topic.

**Scaling Type:** Sets which axis the scale is to be applied on.

**Animation Time:** Sets how long the Animation lasts "**Dynamic Animation - Animation Time Properties**".

**Animation Tag:** Sets the variable to associate to the animation "Dynamic Animation Properties"

**Value Range:** Sets the value range allowed for managing the animation "Dynamic Animation Properties"

### Execution

**Expression:** Used to set the expression whose result will be used in the animation "Dynamic Animation Properties".

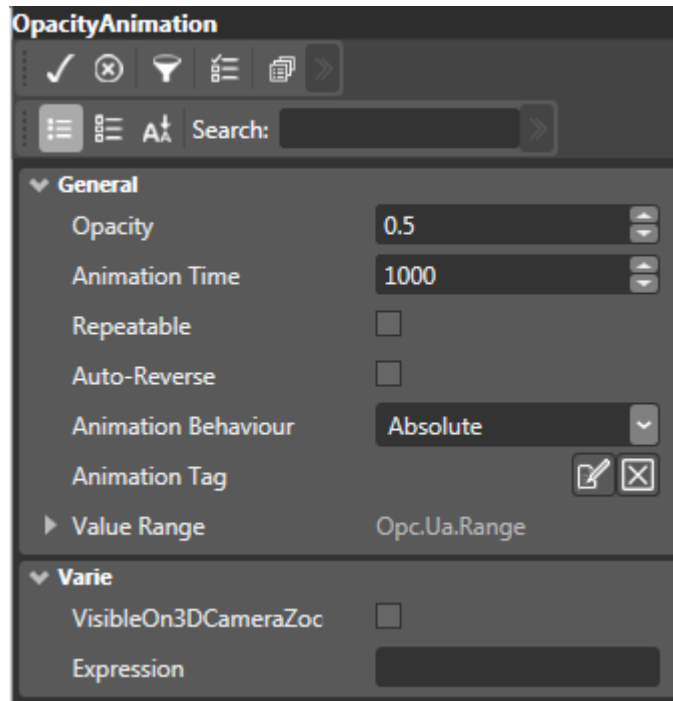
**Repeatable:** This is used for enabling/disabling animation repetition. When enabled the animation will be repeated in a loop while Animation Condition remains active "Dynamic Animation Properties"

**Auto-Reverse:** This is used for enabling/disabling the auto-reverse. When enabled the animation will reverse in a loop until the "Animation Condition" remains active "**Dynamic Animation Properties**".

**Animation Behaviour:** Sets the animation mode in respect to the value range entered in the "Value Range" property for the TAG connected to the animation "Dynamic Animation Properties".

## 8.17. Transparency Animation

The Dynamic "**Transparency**" Animation command is used to assign a variable to the object to determine the object's transparency and visibility values according to the following parameters:



### Animation Transparency Visibility Properties

#### General

**Transparency (Visibility) value:** Sets the object's visibility rate by defining its transparency level in respect to its original visibility rate. The value = 0 determines maximum transparency (INVISIBLE), while the value = 1 determines maximum visibility. The function's behaviour is conditioned by the setting in the properties described in the "Dynamic Animation Properties" topic.

**Visible On 3D Camera Zoom:** Used for setting the transparency animation based on the zoom level of the object instead of the tag value.

**Scale To:** Used for defining a scale value during blink where the value = 1 (default) refers to the object's natural scale size (100%).

**Animation Time:** Sets how long the Rotation Animation lasts "**Dynamic Animation - Animation Time Properties**".

**Animation Tag:** Sets the variable to associate to the animation "Dynamic Animation Properties"

**Value Range:** Sets the value range allowed for managing the animation "Dynamic Animation Properties"

#### Execution

**Expression:** Used to set the expression whose result will be used in the animation "Dynamic Animation Properties".

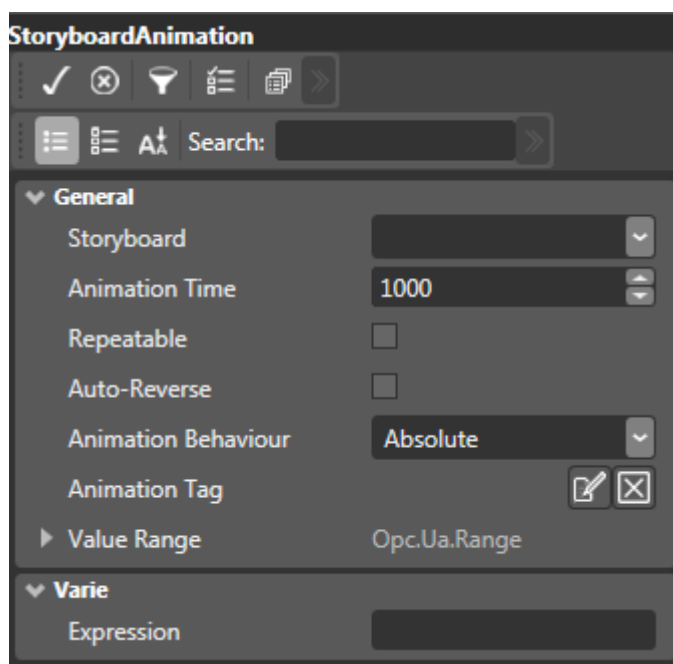
**Repeatable:** This is used for enabling/disabling animation repetition. When enabled the animation will be repeated in a loop while Animation Condition remains active "Dynamic Animation Properties".

**Auto-Reverse:** This is used for enabling/disabling the auto-reverse. When enabled the animation will reverse in a loop until the "Animation Condition" remains active **Dynamic Animation Properties**".

**Animation Behaviour:** Sets the animation mode in respect to the value range entered in the "Value Range" property for the TAG connected to the animation "Dynamic Animation Properties".

## 8.18. Storyboard Animation

The Dynamic **"Storyboard"** Animation command is used for activating a storyboard, available as an imported symbol, that is conditioned by a variable value. When the symbol is provided with conditioned dynamic animation in its XAML structure (defined Storyboard), the animation can be activated with the associated variable trigger. The function's behaviour is conditioned by the property settings described in "Dynamic Animation Properties".



### Storyboard Animation Property

#### General

**Storyboard Name:** This is used to select a storyboard to link to the project.

**Animation Time:** This is used to set the duration of the animation "Dynamic Animation - Animation Time Properties".

**Animation Tag:** This is used to set the tag to associate to the animation **"Dynamic Animation Properties"**.

**Value Range:** This is used to set the value range permitted for managing the animation "Dynamic Animation Properties".

#### Execution

**Expression:** This is used to set the expression whose result will be used in the animation "Dynamic Animation Properties".

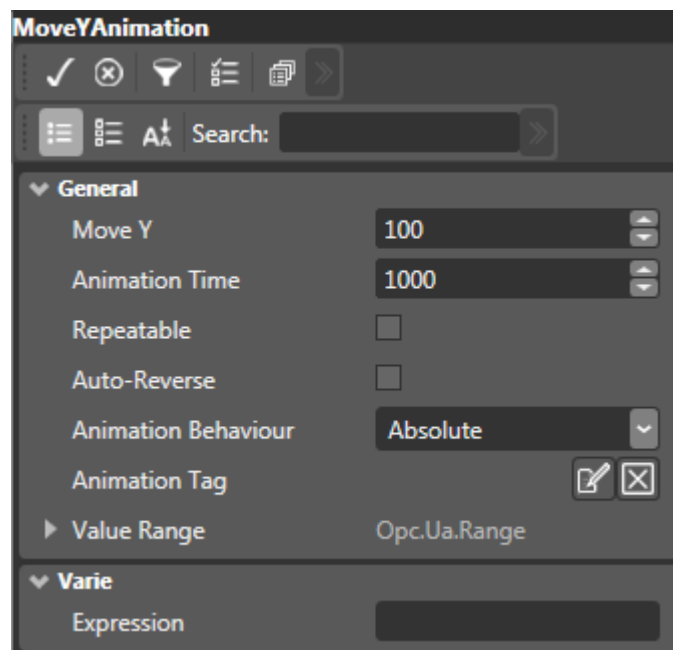
**Repeatable:** This is used to enable/disable animation repetition. When enabled the animation will repeat in a loop while the "Animation Condition" is active "Dynamic Animation Properties".

**Auto-Reverse:** This is used to enable/disable the reverse animation repetition. When enabled the reverse animation will be repeated while the "Animation Condition is active "Dynamic Animation Properties".

**Animation Behaviour:** This is used to set the animation mode in respect to the value range entered in the "Value Range" property for the TAG connected to the animation "Dynamic Animation Properties".

## 8.19. Verticle Y Movement Animation

The "**Move Y**" Dynamic Animation command is used to assign a variable to the object to determine its position by using a vertical movement along the Y axis according to the following parameters:



### Move Y Animation Properties

#### General

**Vertical Movement value:** This is used to set the value in pixels of the object's movement on the Y axis in respect to its initial point. This behaviour is conditioned by the property settings described in the "Dynamic Animation Properties" topic.

**Animation Time:** This is used to set how long the Animation lasts "**Dynamic Animation - Animation Time Properties**".

**Animation Tag:** This is used to set the variable to associate to the animation "Dynamic Animation Properties"

**Value Range:** This is used to set the value range allowed for managing the animation "Dynamic Animation Properties"

## Execution

**Expression:** This is used to set the expression whose result will be used in the animation "Dynamic Animation Properties".

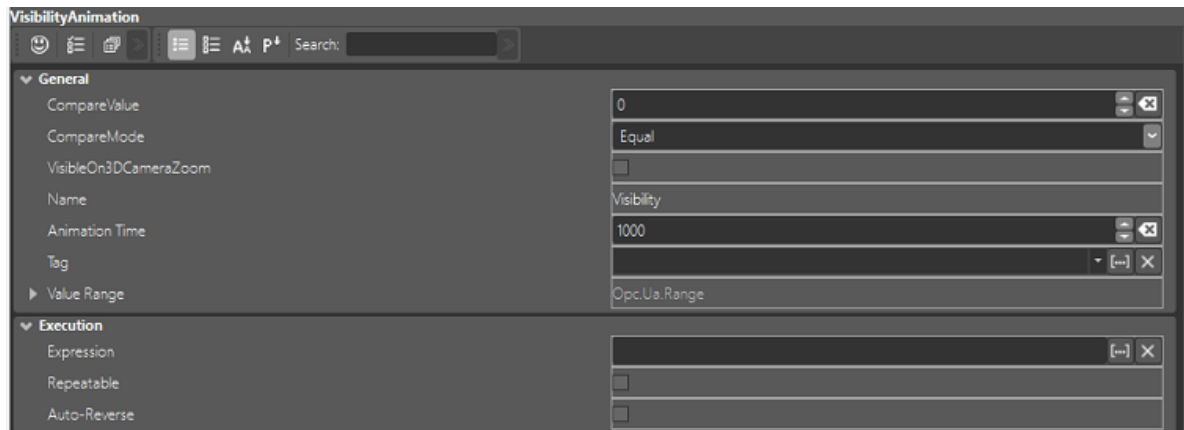
**Repeatable:** This is used to enable or disable animation repetition. When enabled the animation will be repeated in a loop while Animation Condition remains active "Dynamic Animation Properties"

**Auto-Reverse:** This is used to enable or disable the auto-reverse. When enabled the animation will reverse in a loop until the "Animation Condition" remains active **Dynamic Animation Properties**".

**Animation Behaviour:** This is used to set the animation mode in respect to the value range entered in the "Value Range" property for the TAG connected to the animation "Dynamic Animation Properties".

## 8.20. Visibility Animation

The Dynamic **Visibility** Animation command allows you to assign objects with a variable whose value will determine their visibility.



## Visibility Animation Properties

### General

**Tag:** This is used to select a Tag that will be used as a reference to manage the animation with. When left blank, the animation will inherit the object's contextual value if one exists. In cases where the object is part of a symbol, the contextual Tag of this symbol will be inherited if one exists. "Dynamic Animation Properties".

**Animation Time:** This is used to set the Animation's duration. "**Dynamic Animation Properties**".



String type variables can also be used as reference Tags. In this case, however, the comparison function functions only if a numeric value has been inserted in the tag. For example, if the compare is 1, inserting the '1' character in the string variable will activate the visibility animation otherwise the object will remain invisible.

**Compare Value:** The reference value to determine the object's visibility or invisibility is entered in this field. This value will be compared with the value of the tag assigned to the animation as a reference tag or inherited from the symbol according to the selection made in the 'Compare Mode' field.

**Compare Mode:** this field is used to select the compare mode with which the object's visibility will depend on. The comparison modes are:

- Equal
- Major
- Minor
- MajorEqual
- MinorEqual

A comparison will be performed on the 'Compare Value' field and the value contained in the reference Tag.

**Zoom Level Visibility On 3D Camera:** this property has effect only when the visibility animation is applied to a 3D symbol element and not to the symbol as a whole. When this option is enabled, the element's visibility will be management exclusively according to the zoom level executed in the object's 3D camera (right click + drag object). In addition, there is no need to define a Tag to use as a reference and the Compare Mode will be ignored. In this case, the element will become visible when the object's scale level is lower than the value indicated in the 'Compare Value'. Take into account that the value 1 represents the 3D object's initial scale value. For example, if you set the value = 2, the element will become visible only when the 3D symbol reaches a zoom value that is twice the initial size.

**Expression:** This is used to set the expression whose result will be used in the animation. "**Dynamic Animation Properties**".

**Repeatable:** This is used to enable/disable the animation's repetition. When enabled, the animation will be repeated in a loop while the 'Animation Condition' is active. "**Dynamic Animation Properties**".

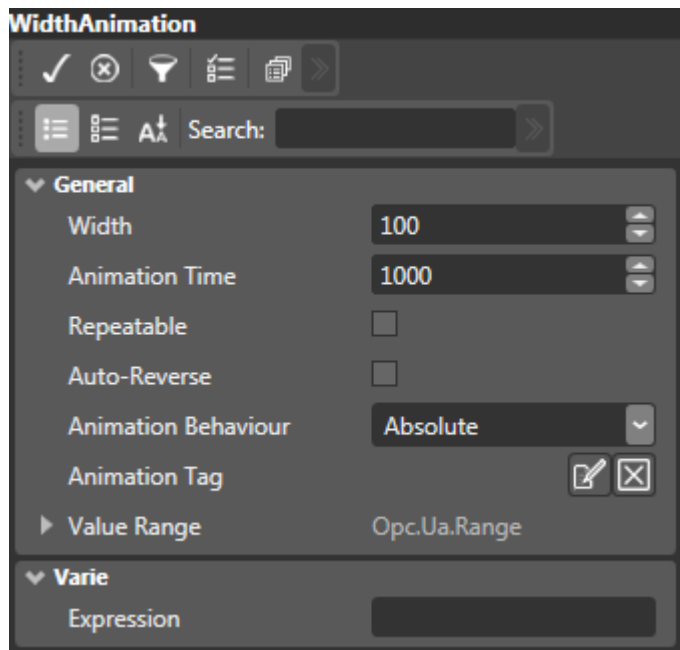
**Auto-Reverse:** This is used to enable/disable the animation's auto-reverse. When enabled the animation will be repeated in a loop in auto-reverse mode while the 'Animation Condition' is active. "Proprietà di Animazione

**Reference Value/Value Range:** This is used to set the value range consented to manage the animation. "**Dynamic Animation Properties**".

**Animation Behaviour:** This is used to set the animation's behaviour mode management mode in respect to the low and high values inserted in the 'Value Range' property for the TAG connected to the animation. "**Dynamic Animation Properties**".

## 8.21. Width Animation

The Dynamic "**Width**" Animation is used for assigning a variable to the object that will determine the object's width size in pixels according to the following parameters:



### Width Animation Properties

#### General

**Width value:** This is used to define the width size in pixels to be reached. The behaviour of this function is conditioned by the settings in the properties described in the paragraph entitled: "Dynamic Animation Properties".

**Animation Time:** This is used to set how long the Animation lasts "**Dynamic Animation - Animation Time Properties**".

**Animation Tag:** This is used to set the variable to associate to the animation "**Dynamic Animation Properties**"

**Value Range:** This is used to set the value range allowed for managing the animation "Dynamic Animation Properties"

#### Execution

**Expression:** This is used to set the expression whose result will be used in the animation "Dynamic Animation Properties".

**Repeatable:** This is used to enable/disable animation repetition. When enabled the animation will be repeated in a loop while Animation Condition remains active "Dynamic Animation Properties"

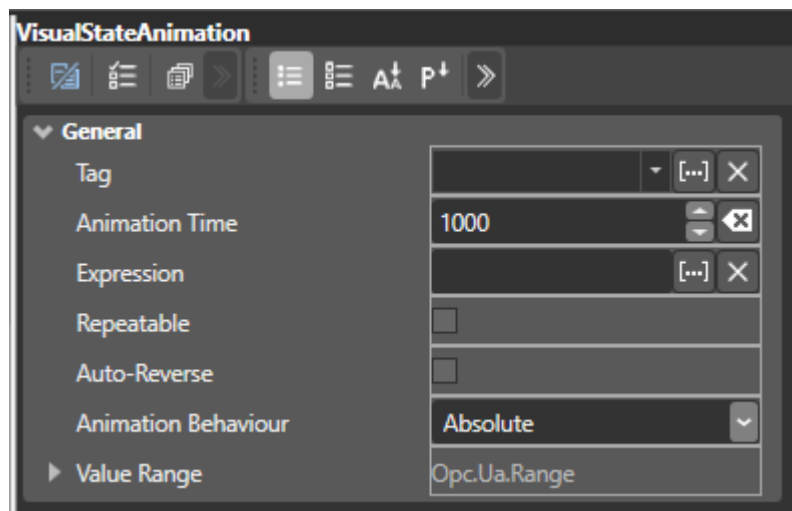
**Auto-Reverse:** This is used to enable/disable the auto-reverse. When enabled the animation will reverse in a loop until the "Animation Condition" remains active "**Dynamic Animation Properties**".

**Animation Behaviour:** This is used to set the animation mode in respect to the value range entered in the "Value Range" property for the TAG connected to the animation "Dynamic Animation Properties".



## 8.22. Visual State Animation

The 'Visual State' Dynamic Animation command allows you to set the object with a 'state' (eg. Normal, Pressed, MouseOver, etc.) based on the value of the referenced Tag. In this case, the object in question will expose a list of Visual States. Not all objects provide this list which is typically managed by Button type objects.



### Visual State Animation Properties

#### General

**Tag:** This field is used to select the reference Tag to manage the animation. When leaving this field empty, the animation will inherit the object's contextual tag if one exists. If the object is part of a symbol, it will then inherit that of the symbol container's and if one exists (also see: "Proprietà di Animazione Dinamica").

You can either insert a String Type tag or an integer type tag to use as an animation Tag. If you opt to use a String type tag, you must insert the name of the visual state to be applied as its value. For example:

- Normal
- Pressed
- MouseOver
- Disabled
- Focused
- Checked
- Unchecked
- Indeterminate

If you opt to use the Integer type Tag, the value you insert will be used as an index to search for the Visual State in the list exposed by the object.

NOTE: to learn more about Visual State Lists of WPF objects, please refer to Microsoft's standard documentation.

**Animation Time:** This is used to set the Animation's duration (also see: "Proprietà di Animazione Dinamica").

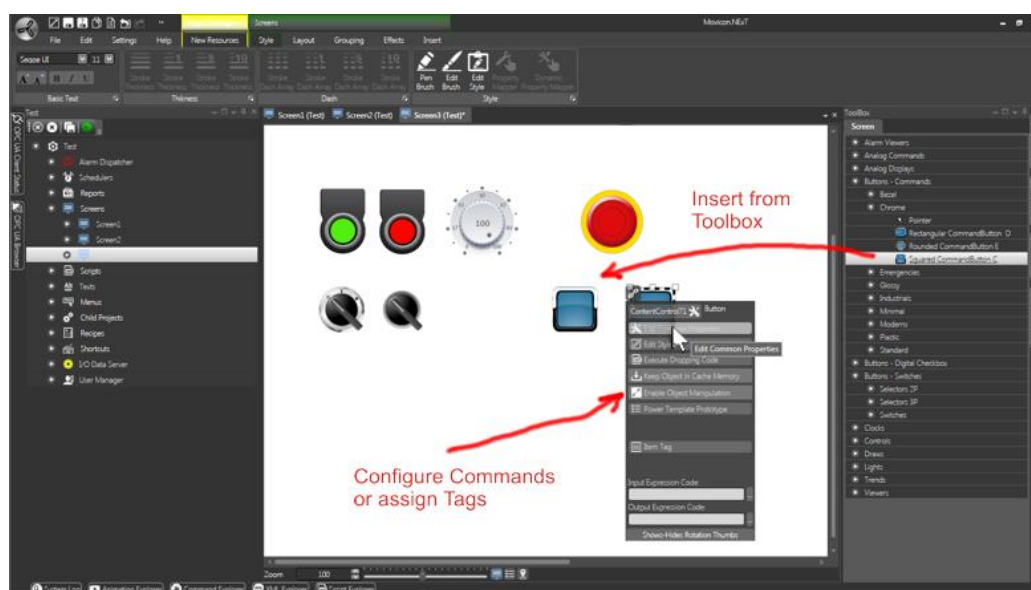
**Expression:** This is used to set the expression whose result will be used in the animation (also see: "Proprietà di Animazione Dinamica ").

## 9. Commands for objects

### 9.1. Command Settings

ToolBox and Symbol Library objects that can be inserted on screen also have different ways of functioning such as displaying/changing a variable value for instance (display, Potentiometers, sliders, check-box, etc.) or displaying real time or historical data (Alarm windows, DataAnalysis, Trend, etc.) or for creating plant layouts with specific objects (graphical design elements from the toolbox or symbols from the library). Independently from which object is inserted on screen, it is always possible to associate each object with a command list that will be executed when the object is clicked on.

The objects that are normally predisposed for executing command lists are those Buttons from the ToolBox's 'Buttons>Commands' category. However, it is now possible to associate commands to any object. What distinguishes a 'Command Button' type from all the other objects is that the 'Command Button' type properties ('Button Behaviour' > 'Click Mode') can be set to execute a command or command list by either pressing the mouse key down (Mouse Press) or upon releasing it (Mouse Release) or hovering the mouse over the button (Mouse Hover). Instead, all the other objects' commands or command lists are executed only when the mouse button is released (Mouse Release).



**Even though it is possible to associate a Command list to any object from the Toolbox, there are some objects that have been specifically designed to 'react' on individual variables without being set with a command list to do so.**

#### Objects that act upon variables

These types of objects are designed to act upon variables, both for displaying and changing the variable's value. After inserting these objects on screen, they only need to be set with a reference Tag (Contextual Tag) on which the object will act upon by

setting/displaying the value as a consequence of the action executed by the user at runtime. These objects include the 'Check-Box', 'Display', 'Potentiometers' and 'Selector' types for example.

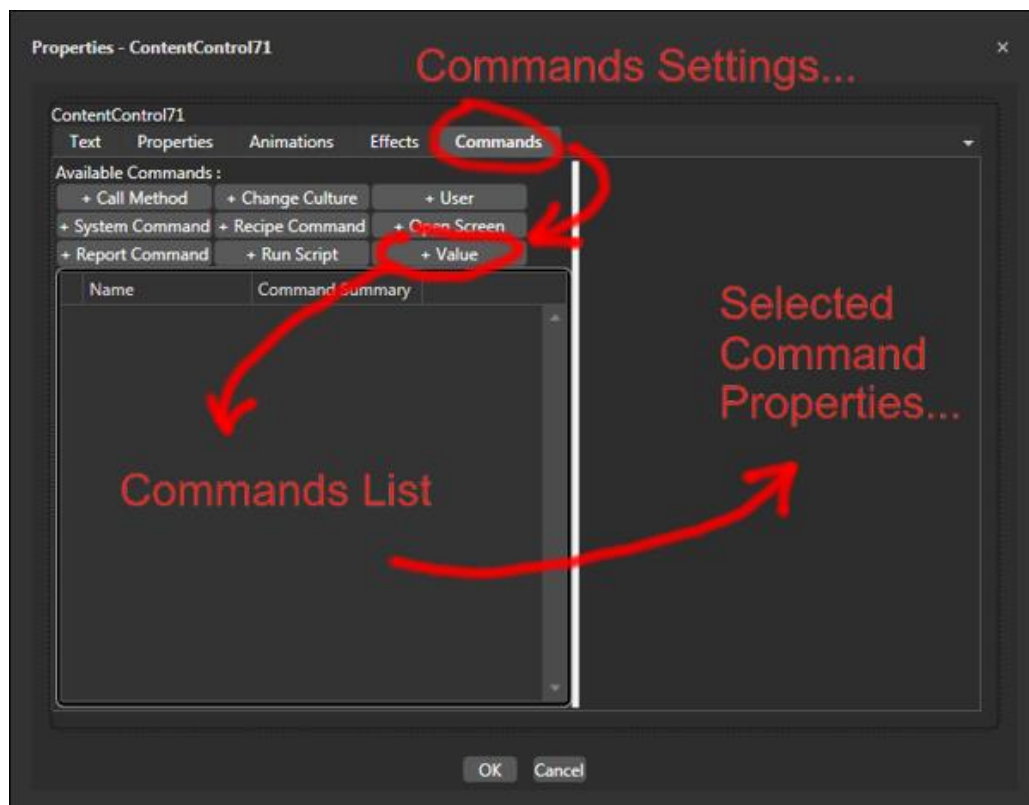
### Objects that execute commands

As already mentioned above, all objects that can be inserted on screen from the toolbox and Symbol library can also execute Command Lists. Those objects most indicated to do this are the Command Button Types which are found in the ToolBox 'Buttons>Commands' category.

### Assigning Commands

Assigning commands to objects can either be done by right clicking on the object to open the 'Common Property Editor' window straight away and then selecting the 'Commands' tab or by selecting and clicking the object to open the adorer menu to do the same thing.

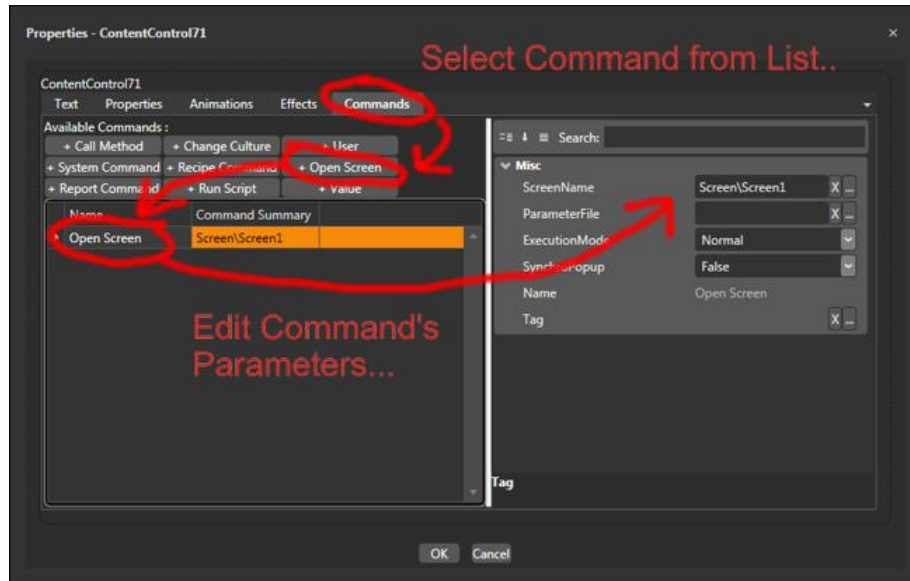
For example, when you open the 'Common Property Editor' using one of the techniques explained beforehand, the window below will appear containing the Commands tab. Once clicking this tab, a list of the available commands, that can be associated to objects, will show on the top right of the window in the form of buttons:



After having added the desired command using the appropriate buttons, you can then proceed to setting the relative parameters which appear on the right side of the window when clicking on the command added. The window will update with parameters according to the command selected.

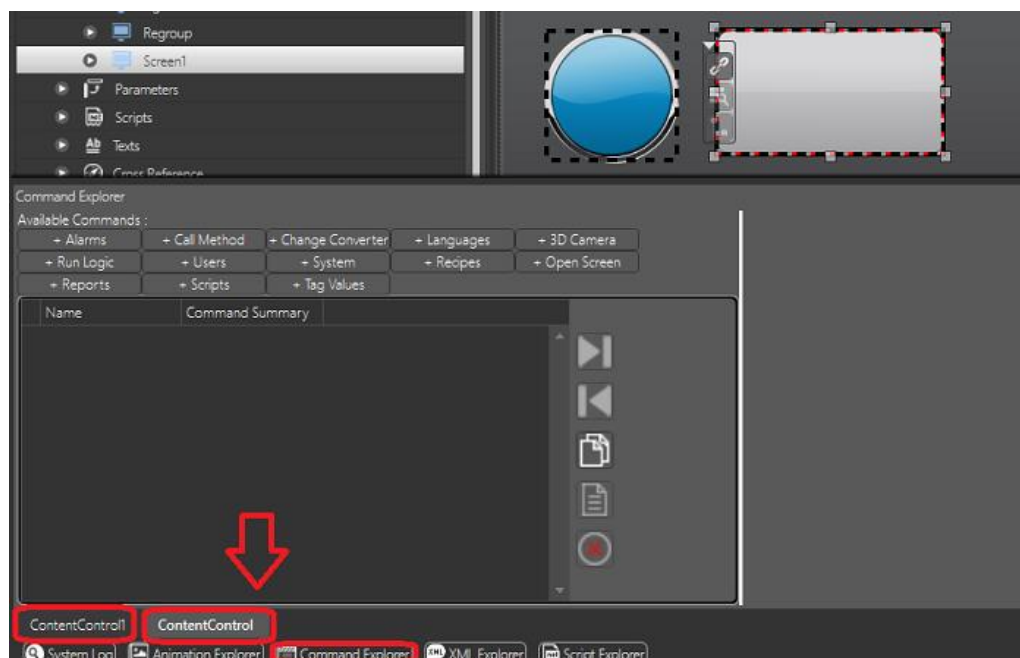


For example, if you wish to create a change page command, you will need to select the 'Open Screen' command from the command list as indicated in the below screenshot. Please also note that the same command can be inserted a number of times and a list of commands can be created and executed with one single click.



### Assigning Commands to multiple object selection

When you make a multiple object selection, you will still be able to insert the desired Command for each one. However, this is done by using the 'Command Explorer' window at the bottom of the screen and not the 'Common Property Editor' window, which cannot not be used with multiple object selections. When you make a multiple object selection and open the 'Command Explorer' window, a tab will show for each object selected at the bottom. When selecting one of these tabs the list of commands available for that specific object will appear ready for you to add and edit the ones you wish to add to the object's command list:



It is not possible to apply the same command to a multiple object selection all at once. However, it is possible to copy and paste a Command from one object to another using the appropriate buttons located vertically in the middle of the Command Explorer window.

## Speech Command

By means of using the speech command you will be able to execute a series of commands associated to the object. In order to configure this resource, you will need to associate one or more commands to an object and then specify the word or phrase to be used to activate the desired commands in the 'Speech Command' property. For example, the word 'start' can be inserted in the 'Speech Command' property :



When pronouncing 'Start' correctly, the previously set object's command list will start execution.



In order to use the Speech Command function you will need to pre-install an audio card and a microphone. English is currently the only language supported for Speech Command terminology.

## 9.2. Script Execution Commands

The following properties are used to define the parameters of a "Script" command to execute a previously inserted VB.NET Basic Script resource in the project:

### ScriptName

The name of the script to be called.

### Parameter

This field is used for inserting a string which will be passed to the script as its execution parameter. This parameter can be retrieved in the script's code with this function:

```
StartupContext.Parameter
```

When dealing with a string, it can be formatted in way to provide its passage through several parameters separated by a predefined character. For example, a series of values can be passed by separating them with the "pipe" character:

```
1|5,6|29
```

In this way once the parameter has been retrieved, the string can be split using the "pipe" character as a separator.

### ExecutionMode

The script execution mode:

- **Syncro**: script execution is synchronized with call
- **Normal**: default mode, script execution is asynchronized with call
- **Shared**: The script shares the same execution thread of the other scripts that have been started up in the same way. This consents a more limited use of the machine's resources
- **Stop**: this command stops the script from running and loads it into memory. In this way variable subscriptions used by the script will be removed from the Server according to

the time set in the "Connection Settings" (belonging to the project or the script itself if this has its own session).

#### Tag

Sets the Tag associated to the command.

#### Express

Sets the expression associated to the command.

#### Delay Command Secs

This command functions normally when set with the default value. When set to a value higher than 0, it only functions when kept pressed down for duration of the time set in seconds.

Commands activated in this mode (delay) exclude any normal commands eventually associated and vice versa.



This modality is not available in **webclient**; when a button is kept pressed down, all the delay commands that it has been associated with will be executed once only in sequence.

## 9.3. System Commands

The following properties are used for configuring the settings of System Commands to execute a command type selected from the proposed list of commands:

#### CommandType

The type of command to be executed.

- **Quit**: aborts project being run
- **Logoff**: stops project being run and the logs off Windows user.
- **Lock**: executes the Windows Lock command
- **Shutdown**: Shuts down project being run and then Windows
- **Restart**: stops project being run and then restarts Windows
- **RunApp**: Runs or stops the application specified in the "Parameter" field based on the "ExecutionMode", field

#### Parameter

The name and path of the application name to be started up or stopped using the "**RunApp**" command is specified in this field. The path can be omitted if the application has been inserted in variables within the Windows (e.g. C:\Windows\system32\Notepad.exe, or Notepad.exe)

#### ExecutionMode

The "**RunApp**" command's execution modes. The mode options are:

- **Normal**: the application indicated in the "**Parameter**" field will be started up normally i.e. not synchronized with Movicon.
- **Syncro**: the application indicated in the "**Parameter**" field will be started up in synchronized mode with Movicon, i.e. the Movicon interface will not be accessible until the application has terminated.
- **Shared**: not used.

- **Stop:** the application indicated in the "**Parameter**" field will be terminated.

### Delay Command Secs

This command functions normally when set with the default value. When set to a value higher than 0, it only functions when kept pressed down for duration of the time set in seconds.

Commands activated in this mode (delay) exclude any normal commands that may have been associated and viceversa.



This modality is not available in **webclient**; when a button is kept pressed down, all the delay commands that it has been associated with will be executed once only in sequence.

## 9.4. Alarm Commands

This group includes commands to acknowledge and reset Movicon alarms.

### General

#### Type

Sets the command type required

**AckAll:** This command is used to acknowledge all active alarms.

**ConfirmAll:** This command confirms all the active alarms by resetting them.

**ToggleSound:** This command enables or disables the warning sound that is activated for those alarms which have not yet been acknowledged according to which user is active.

**ShowStatisticReport:** This command enables a preview of the selected report to be shown in DevExpress format for Alarm statistics.

**PrintStatisticReport:** This command prints the selected report of the alarm statistics in DevExpress format without opening a print preview window beforehand.

**SaveStatisticReport:** This command creates and saves a new report file in the format selected from the FileType property. For further information please go to the 'Various' > '**FileType**' section.

**SendStatisticReport:** This command exports and sends the selected alarm statistics report by email. The email is sent in accordance with the properties set in the SMTP settings in the "Users and Groups".

The email recipient will be the user or user group defined in the property.

#### ReportType

The three types of default Reports interface with the project's Historian Log and show the following information:

- **OrderByDate:** Report showing alarms in date order
- **OrderByDuration:** Report showing alarms in order of duration (starting with the longest to the shortest duration).
- **OrderByOccurrence:** Report showing alarms grouped by occurrence.



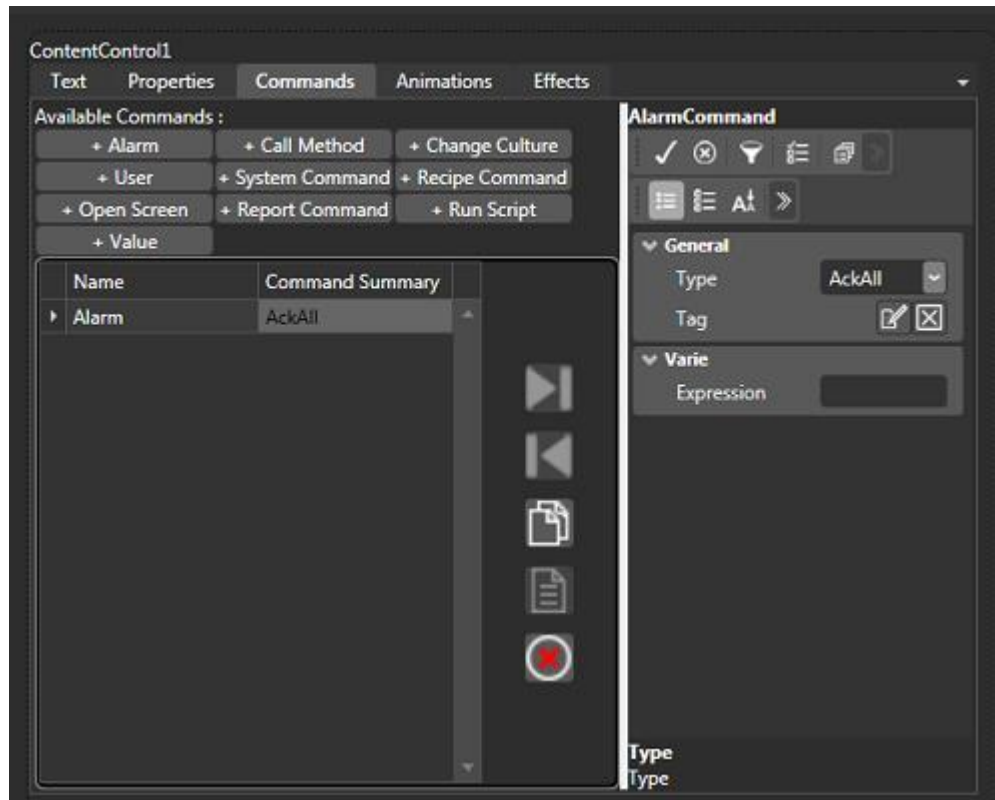
### Recipient

The user or user group, to send the email with attached report file to, is entered in this field.

This parameter is only shown when the "**SendStatisticReport**" command has been selected from the "Type" field.

### ConnectionString

This field is used for configuring the connection string towards the DB server in which the referenced Historian log database resides.



### Delay Command Secs

This command functions normally when set with the default value. When set to a value higher than 0, it only functions when kept pressed down for the duration of the time set in seconds.

Commands activated in this mode (delay) exclude any normal commands that may have been associated and vice-versa.



This modality is not available in **webclient**. When this command button is kept pressed down, all the associated delay commands will execute once in sequence only.

### Various



Each of the following properties display according to which command has been selected in the previous "Type" section.

### PeriodType

Used for setting the referential period of time for collecting data in the report.

### FileName

Used for setting the file name with which the report will be saved.

**FileType**

Sets the file type with which the report will be saved/exported.

**From**

Sets the senders address to be shown within the email

**FromAlias**

This property allows you to set an "alias" that will be shown instead of the senders name within the email

**MailSubject**

Sets the email's subject.

**MailObject**

Sets the email's body text.

## 9.5. Change Language Command

The "Change Culture" command settings are used for establishing Change Language executions for project text strings using the languages (Cultures) defined in the String Table. This command's parameters can be defined by configuring the following properties:

**Culture Name**

This field is used for selecting the language to activate from a list of language names. This list is populated with the languages that have been inserted in the project's String Table.

If this field is left empty, Movicon will display a window through which the user can choose a language to activate from those available in Runtime.

**Delay Command Secs**

This command functions normally when set with the default value. When set to a value higher than 0, it only functions when kept pressed down for duration of the time set in seconds.

Commands activated in this mode (delay) exclude any normal commands eventually associated and viceversa.



This modality is not available in **webclient**; when a button is kept pressed down, all the delay commands that it has been associated with will be executed once only in sequence.

## 9.6. Recipe Commands

The properties below are used for configuring the parameters of command executions of Recipes inserted in the project:

**RecipeName**

Name of Recipe to be called.

**ExecutionMode**

Recipe display or activation mode set in the "Command Type" parameter. The mode options are:

- **Normal:** executes the command in normal mode
- **Silent:** executes the command in silent mode i.e. the pop-up window will not show in the event of error messages

### CommandType

The command type that can be executed for a recipe. The options are:

- **Show:** opens the pop-up window with the Recipe defined in the "Recipe Name" parameter
- **Load:** loads the values of the Recipe defined in the "Tag Recipe Index" from the DataBase and transfers them to the "Tag Data Value" defined for the recipe fields (see topic on "Using Tags to manage recipes")
- **Save:** saves the values set in the "Tag Data Value" defined for the recipe field in the DataBase with the Recipe name defined in the "Tag Recipe Index". If the recipe does not exist in the DataBase, a new record will be added. If it already exists, it will be modified instead. (see topic on "Using Tags to manage recipes").
- **Remove:** removes the recipe with name defined in the "Tag Recipe Index" from the DataBase (see topic on "Using Tags to manage recipes").
- **Activate:** activates the Recipe with the name specified in the "Tag Recipe Index" (see topic on "Using Tags to manage recipes"). The values will be loaded from the DataBase and transferred to the device. If a recipe does not exist with this name, an error will be issued in the Tag Recipe State. Attention: the values that are transferred to the device are read by the DataBase and not by the recipe's Tag Data Value". In this case, a recipe is loaded and then modified using the "Tag Data Values". If the recipe is activated before being saved with the "Save Recipe" command, the old values in the database will be transferred. This command can nevertheless be executed without defining the recipe's "Tag Data Values".
- **Read:** reads the device values and copies them to the recipe's "Tag Data Value" (see topic on "Using Tags to manage recipes").

### Command Timeout

Timeout for executing the "Activate" and "Read" recipe commands. This time is set in milliseconds.

### Delay Command Secs

This command functions normally when set with the default value. When set to a value higher than 0, it only functions when kept pressed down for duration of the time set in seconds.

Commands activated in this mode (delay) exclude any normal commands eventually associated and viceversa.



This modality is not available in **webclient**; when a button is kept pressed down, all the delay commands that it has been associated with will be executed once only in sequence.

## 9.7. Commands in Methods

The settings of a "Call Method" command type sets the execution of a Method function. The following properties can be configured for setting this command:

### Tag

This field is used for selecting the name of the method to be called.

### Delay Command Secs

This command functions normally when set with the default value. When set to a value higher than 0, it only functions when kept pressed down for duration of the time set in seconds.

Commands activated in this mode (delay) exclude any normal commands eventually associated and vice versa.



This modality is not available in **webclient**; when a button is kept pressed down, all the delay commands that it has been associated with will be executed once only in sequence.

## 9.8. Commands for Reports

"Reports" command types can be set and used for Reports inserted in the project. The Report command type settings are configured using the properties below:

### Report Name

Name of the report to be displayed or managed.

### Command Type

Command can be display or print type.

- **Show:** opens the report
- **Print:** prints the report specified in the "ReportName" property using the system's default printer
- **PrintDialog:** prints the report specified in the "ReportName" property by first opening the window to select the printer to be used
- **Send:** Sends the specified report via email
- **Save:** Saves the specified report

### Modo di Apertura Report

This is used to set the mode with which to open Reports:

- Normal: Full screen
- Syncro: Modale pop-up window
- Shared: Non-modal pop-up window
- Stop: Close report

### File Name

Sets the file name with which the report will be saved.

### File Type

Sets the file type with which the report will be saved/exported.

**Recipient**

This field is used to enter the name of the user or user group to send the email with attached report file to.

**From**

Used to set the sender's address to be displayed in the email.

**FromAlias**

This is used for setting a "alias" name which will display in the email instead of the sender's address.

**Mail Subject**

Sets the email's subject.

**Mail Object**

Sets the email's text (body)

**Connection String**

This field is used for configuring the connection string towards a database which is different to the one defined in the Report properties. When left empty, the Report will use the one set in its properties.

**Parameters**

This is used to set the list of parameters to pass to the report when opened.

**Delay Command Secs**

This command functions normally when set with the default value. When set to a value higher than 0, it will only function when kept pressed down for the duration of the time set in seconds.

Commands activated in this mode (delay) exclude any normal commands eventually associated and vice versa.



This modality is not available in **webclient**; when a button is kept pressed down, all the delay commands that it has been associated with will be executed once only in sequence.

## 9.9. User Commands

The parameter settings of "Users" command types set to execute operative commands for Users in the Password and Protection management are configured in the following properties:

**CommandType**

The command type to be executed.

- **Login:** executes user login. A dialog window opens for inserting the user's name and password.
- **Logout:** executes logoff of the user currently logged in.
- **UnlockUser:** A string type tag can be associated to the command with the name of the user to be re-enabled as its value. In cases where a tag is not specified or has a null

value, a new popup window will open containing a list of users names that have been blocked within the project.

- **UnlockAllUsers:** This command unlocks all the users disabled within the project.
- **Edit:** Opens the users edit window through which it is possible to add or change users at runtime.
- **Change password:** Sets the command to change password using a pop-up that displays after activating the command.



To set the desired restrictions for the "edit" command, you will need to specify the "access level" or "access role" properties of the button used to invoke the command.

### Delay Command Secs

This command functions normally when set with the default value. When set to a value higher than 0, it only functions when kept pressed down for duration of the time set in seconds.

Commands activated in this mode (delay) exclude any normal commands eventually associated and vice versa.



This modality is not available in **webclient**; when a button is kept pressed down, all the delay commands that it has been associated with will be executed once only in sequence.

## 9.10. Commands for Screens

The "Open Screen" commands set to execute operations in project screens can be configured with the following properties:

### Screen Name

Name of screen to be opened or closed.

### Screen Opening Mode

The options are:

- **Normal:** opens the screen specified in the "ScreenName" property in normal mode, using the usual change page technique. When the "ScreenName" parameter is left empty, the command will load the previous page by executing a "close and return to previous" command
- **Synchro:** opens the screen specified in the "ScreenName" property as Modal or PopUp window according to the "SynchroPopup" parameter. When the parameter is left empty the command will have no effect.
- **Shared:** screens opened as Shared will open within one container and display in tile format. This effect is the same one used when opening the Server's "General Settings" tab. In this case there are different windows, "Settings", "Status", "Drivers" and "Transports" displayed within one container whereby each one can be shown in foreground in alternation mode.
- **Stop:** when the "ScreenName" parameter is left empty this command opens the "Tile Page" window or the startup screen, according to the project startup modality type. If the name of the screen is inserted in the "ScreenName" parameter with screen opened

"Synchro" mode, the modal screen will simply close. On the other hand, if the screen was opened in "Normal" mode, a Back command will be used to load the previous screen

### Parameterization File

The parameterization file path is selected in this edit box using the "..." browse button to the right in cases where the screen is to open by passing it parameters.

For further information on using parameter-driven screens please refer to the paragraph on "Screen Parameterization".

### Requested Monitor

Next supports the multi-monitor management. The "Requested Monitor" parameter is managed when opening the screen to identify where it is to be displayed according to this parameter's settings. The requested monitor can also be indicated in the screen's properties. However, the 'Open Screen' command button has priority over the Screen parameters if both have been set.



The "-1" value identifies the default monitor set in Windows. When opening a screen in a different monitor that is not the one set for default (therefore by setting a value that is different from -1), it will always open at the "0;0" coordinates. This will also happen with the Frame and Synchro commands and when specifying the open coordinates.

Monitor identification can be obtained from the operating system's menu which identifies connected monitors.

### Synchro Popup

This parameter only appears when the "Synchro" execution mode has been selected. If this parameter is left set to "False", the screen will open as a modal window subordinated to the called screen. When set to 'True', the screen will open as a Popup at the side of the command that activated it. In this case the screen will not be modal and it can be closed by simply clicking the area nearby.

### Synchro Frame

This parameter is considered only when the "Synchro" execution mode has been selected. When this parameter is enabled, the window will open in frame mode as a PopUp and not as a modal window and will remain open in the foreground when clicking on screen objects behind it.

### IsRelative

This parameter is considered only when the 'Synchro' mode has been selected. In this case the screen, whether Modal or Frame, will open at the X and Y coordinates inserted in the appropriate fields in absolute mode (False value) or in the relating called screen mode (True value). Basically, if the parameter is set to True, the screen, whether Modal or Frame, will open at the coordinates indicated by taking the called screen's position as origin and not the monitor screen's origin.



This parameter is only considered when the open screen command is invoked from a Screen's object, such as a button for instance. It will not be considered if the command is invoked from a resource such as a Menu, Accelerator, Event resource and other such like.

## X

This parameter is only considered when 'Synchro' mode has been selected. In this case the screen, whether Modal, Frame or Synchro will open in the X position indicated by the parameter and not in the centre of the monitor.

## Y

This parameter is only considered when 'Synchro' mode has been selected. In this case the screen, whether Modal, Frame or Synchro will open in the Y position indicated by the parameter and not in the centre of the monitor.

### AutoClose Sec

This parameter is only considered when the "Synchro" screen opening mode has been selected. When inserting a value that is different from zero in this field, the screen, whether Modal or Frame, will close automatically when time expires (expressed in seconds). For example, if you insert the value 5, once open the Modal or Frame screen will automatically close after 5 seconds.

### Delay Command Secs

The command will be executed normally with the default value. However when using a value that is higher than 0, the command will be executed only when pressed down for the duration set in seconds.

The commands executed in this mode (delay) exclude any normal commands eventually associated and vice versa.



This mode is not available in **webclient**; when keeping a button pressed down, all its associated delay commands will be executed once only in sequence.

## 9.11. Commands for Tag Variables

The "value" command type settings are used for setting value associations to project Tag variables. This command's parameters are defined by configuring the following properties:

### Tag

When the command is associated to an object from the toolbox, such as a Button or Hot Region, and this Tag field is left empty, the command will be applied to any default variable associated to the object.

### Value to Set

Value which will be set in the variable or which will be used as increase/decrease step.



Attention! In cases where a decimal value is inserted (eg. 1.3 etc.) you will need to use the point (".") as a decimal separator, independently of Windows settings.

### Command Type

Possible command types include:

- **Set:** sets the variable with the value defined in the "Value" parameter



- **Increase:** Increases the variable value defined in the "Value" parameter
- **Decrease:** decreases the variable value defined in the "Value" parameter
- **Toggle:** Toggles the variable between the "0" value and the one set in the "Value" parameter
- **Impulsive:** when pressing the button, the variable will set to the value defined in the 'Value' parameter and will remain with these values until the button is released. When not set at zero, the 'Time' parameter will indicate the impulse's maximum time.



On Web Client:

1. The ImpulsiveLatch command type can be used only in cases in which the 'Time' parameter has been set with a value that is NOT '0'.
2. The Impulsive and ImpulsiveLatch commands present the same behaviour.

- **ImpulsiveLatch:** When pressing the button, the variable will set it to the value defined in the 'Value' parameter until the time set in the 'Time' parameter has expired independently from the fact that the button remains pressed or released. The 'Time' parameter indicated the impulse's minimum time duration. In cases where the 'Time' parameter is set to zero, the command will behave like the 'impulsive' command.



On Web Client:

1. The ImpulsiveLatch type commands can be used only in cases where the 'Time' parameter has been set with a value that is NOT '0'.
2. The Impulsive and ImpulsiveLatch commands present the same behaviour.

- **TransferValue:** This allows the value of one variable to be transferred to another variable. In this case the original value is the one specified in the 'Tag' parameter and the destination variable is the one specified in the 'Destination Tag' parameter. Different variable types can also be used whereby Movicon.NExT will perform the appropriate conversions providing that the two tag types are compatible to allow the correct value transfer. For example it will not be possible to transfer alphanumeric text of one variable string to a numeric variable.



The "Transfer Value" command always updates the whole Tag. This means that if an Array element or a Tag Bit is defined in the Expression field, the whole destination Tag will be however written and therefore all the Array elements and Tag bits will be updated. Basically, in this context it would not make sense to specify the element of an Array or the Bit of a Tag.

- **NumericPad:** This command opens a Numeric pad as an alternative to using the keyboard for inserting values. The value will be reported to the variable once confirming with the 'OK' pad key. This numeric Pad is fundamentally used with Touch-screens.



**Numeric Variables:** when the variable is a numeric type, the number set will be represented by the same number of decimal figures after the point. For example, the '2' value will be set in the 'x.xx' format. When the set value is negative, the absolute value will be taken, therefore when entering '2' or '-2' the same result will be achieved. When the value is a

floating point type with decimal figures, the value will be rounded off to the nearest number. Therefore when entering '1.4', it will be rounded off and entered as '1' and when entering '1.6,' it will be rounded off and entered as '2'.

- **AlphanumericPad:** This command opens a Alphanumeric Pad as an alternative to using the keyboard for inserting characters. The value will be reported to the variable once confirming with the 'OK' pad key. This numeric Pad is fundamentally used with Touch-screens.
- **ResetStatistic :** Resets the statistics information, when enabled, of the variable associated to the command.

### Synchronous

When this option is enabled, the command will be executed synchronized with the User Interface. This means that if an open screen command comes after the Tag's setting command in the command list, the screen will only open after the Tag has been set,

### TransferToTag

This field shows only when the 'TransferValue' command has been selected. This field is used to select the destination Tag to which to transfer the tag value, defined in the 'Tag' field.

### Delay Command Secs

This command functions normally when set with the default value. When set to a value higher than 0, it only functions when kept pressed down for duration of the time set in seconds.

Commands activated in this mode (delay) exclude any normal commands eventually associated and vice versa.



This modality is not available in **webclient**; when a button is kept pressed down, all the delay commands that it has been associated with will be executed once only in sequence.

### Expression

This is used to set the expression associated to the command.

This will make is possible, for example, to execute the command on a specific Tag bit or on a specific element in cases where the Tag is an array type. It is also possible to insert an expression for the 'Transfer Value' command only, to determine the destination value (eg. = [x] + 10).

### TransferToTag

This field will only appear when the 'TransferVaule' command type has been selected and it is used to select the destination Tag in which the variable value defined in the 'Tag' field will be transferred.

### MinValue

This field only appears when the "NumericPad" or "AlphanumericPad" Command Type has been selected and it is used to define the minimum value or minimum number of characters that the Tag can be set with.

When setting the "MinValue" value equal to the "MaxValue" value, the Pad will open without imposing limits on the value to be set.

**MaxValue**

This field appears only when the "NumericPad" or "AlphanumericPad" Command Type has been selected and it is used to define the maximum value or maximum number of characters that Tag can be set with.

When setting the "MinValue" value equal to the "MaxValue" value, the Pad will open without imposing limits on the value to be set.

**TagMinValue**

This field only appears when the "NumericPad" or "AlphanumericPad" Command Type has been selected and it is used to select a project Tag whose value will be taken in runtime as the minimum value or minimum number of characters that the Tag can be set with. In this way, this value can be made dynamic in runtime.

When setting the "MinValue" value equal to the "MaxValue" value, the Pad will open without imposing limits on the value to be set.

**TagMaxValue**

This field only appears when the "NumericPad" or "AlphanumericPad" Command Type has been selected and it is used to select a project Tag whose value will be taken in runtime as the maximum value or maximum number of characters that the Tag can be set with. In this way, this value can be made dynamic in runtime.

When setting the "MinValue" value equal to the "MaxValue" value, the Pad will open without imposing limits on the value to be set.

**Decimals**

This field only appears when the "NumericPad" Command Type has been selected and it is used to define the number of decimal figures to be displayed. When associating the Numeric Pad with Floating type variables, this parameter determines the number of decimal figures displayed. When the integer type variables are associated to the Numeric Pad, this parameter allows the value to be displayed with a floating point. For example, when setting the field to 2 with the integer variable with a 123 value, the value will display as 1.23. In this case, the minimum and maximum limits will not take into account the set format and will continue to work in the variable's entire value.

For example, with a one decimal figure format and a maximum limit of '100', it will be possible to edit a maximum value in the variable equal to '100' and therefore the value displayed in the Numeric Pad will be a maximum of '10.0'.

## 9.12. 3D Camera Viewer Commands

The "3D Camera" command settings for view activation are used for displaying previously stored 3D model scenes inserted on screen.

This command requires the 'views' of the 3D camera model in order to activate them and display the scene previously stored.

**Call 3D Control**

The name of the project's 3D object (or control) to be interacted with is entered here.

**View Name**

The name of the "View" to be activated is set here (needs to be recorded previously in runtime).

### Delay Command Secs

This command functions normally when set with the default value. When set to a value higher than 0, it will only function when kept pressed down for duration of the time set in seconds.

Commands activated in this mode (delay) exclude any normal commands that may have been associated and viceversa.



This modality is not available in **webclient**. When the button is kept pressed down, all the commands that it has been associated with will be executed once only in sequence.

## 9.13. Change Converter Commands

The Change Converter commands is used to change the converter type used in all those objects that have been set with the same converter.

For further information regarding the use of converters please refer to the topic on Unit\_Converter

### Converter Name

The name of the convertor is set here.

### Execution Mode

- Normal: activates the selected convertor
- Stop: deactivates the selected convertor. When the active convertor is deactivated, the variable will only be displayed with its effective value.
- Syncro: not supported
- Shared: not supported

### Delay Command Secs

This command is executed in normal mode when set with the default value. When setting this command with a value that is higher than 0, it will only execute when kept pressed down for the time set in seconds. Commands executed in delay mode exclude any normal commands associated and vice versa.



This mode is not available on the **webclient**; when keeping the button pressed down, all the delay commands associated to it are only executed once in sequence.

## 9.14. Logic Commands

The settings of logic type commands are used to start or stop logic executions.

### Logic Name

This field is used to select the logic name.

### Execution Mode

- **Normal:** logic is started and kept in continuous cyclic execution. A 'Stop' command must be used to stop logic execution which will however stop when project stops running.
- **Synchro:** logic is executed once only then stopped automatically. Therefore only one complete cycle of the logic will be executed.
- **Shared:** logic is started up in "debug" mode, meaning that it will be started up in 'Normal' mode with an opened debug window showing the logic blocks edited in design mode. These logic blocks will be animated in the debug window according to the Tag values and combined logic results. Closing the debug window will also stop logic execution.
- **Stop:** Stops the logic set in the command.

### Delay Command Secs

The command is executed in normal mode when set with the default value. When set with a value higher than 0, the command will only be executed when pressed down for the time set in seconds.

The commands executed in this mode (delay) exclude any normal commands associated and vice versa.



This mode is not available in the **webclient**; all the delayed commands associated to a button are executed once only in sequence when it is pressed down, .

## 9.15. Open Map Commands

The Open Map command settings support normal mode to open maps in the mainframe or synchro mode to open modal geomaps.

The command also supports geographical areas that are zoomed upon opening the map.

This function can be used to invoke modal geomaps in the zone desired or to create a main map invocation layout screen in different areas.

### Execution Mode

This is used to select the which mode to open the map with.

### Zoom To

This is used to insert the coordinates with which the map will open up on.

### Enable Zooming Scrolling

Activates the possibility to use the scroll and zoom on the map.

### Show MiniMap

Activates the minimap at the top left.

### ShowNextButton

Allows the Next button to be shown to navigate through the screens on the GeoPage.

## 9.16. Text to Speech Commands

The settings of "Text to Speech" command types are used to vocally synthesize a string inserted in the command. This feature requires the use of a speaker to reproduce the desired text string into speech.

The Text To Speech command can be used wherever possible to define a command list.



This feature is not available on **webclient**

### Voice Name

This is used to specify which voice to use among those installed in the system and displayed in the Text To Speech section in the control panel.

This field can also be left empty to use the voice selected for default in the control panel's Text To Speech section.

### Volume

The value range starts from 0 to 100 to indicate the volume intensity to use when reproducing the command into speech.

### Rate

Indicates the voice speed rate ranging from 0 being the slowest to 10 being the fastest.

### Speak

Represents the 'read' text string by the command execution. If the text is presented as string ID in the text table, the string of the language active at runtime will be used.

### Long Click Time

When set with the default value, this command will be executed normally. When set with a value higher than 0, this command will only execute when pressed for the time duration set in seconds.

Commands executed in this mode (delayed) will exclude any normal commands associated and vice versa.

### Execution Mode

This property can assume the following values:

**Normal:** executes asyncho speech

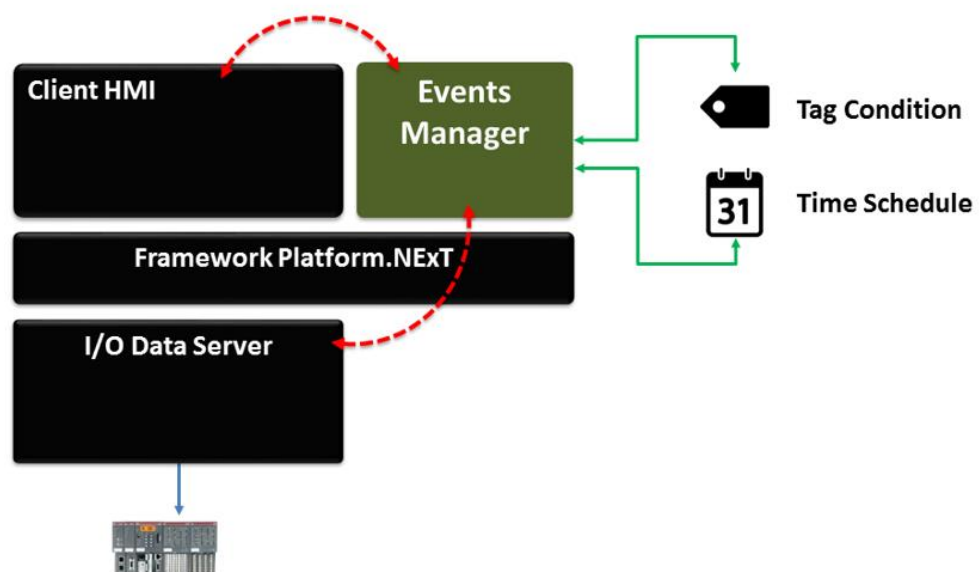
**Syncho:** executes speech and waits until terminated

**Shared:** cancels all pending speak and emits an asyncho speech

# 10. Events Manager

The Event Objects allow commands or command lists to be executed in function with a project Tag state change or according to a time schedule that can be associated to the actual event.

The commands can therefore execute any project operation according to all the various possibilities that can be set as described in the topics relating to Object Commands. That is to say that an event can execute the same commands that can be executed by command buttons but the execution of these commands will be determined according to a time schedule or a Tag status.



The Commands on Event are very handy in projects that have to:

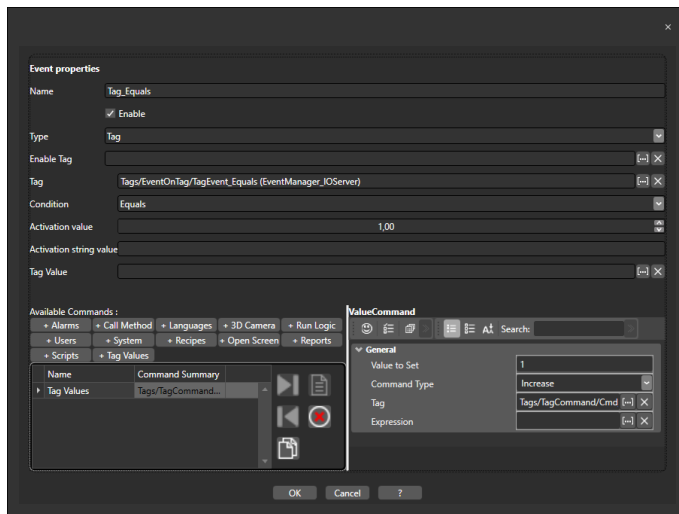
- manage operations for logic, commands, change pages, launching scripts, actions in recipes or reports for example, based on a Tag status change.
- manage operations or command as above but at certain prefixed time intervals (schedulers).

The possibility to manage Event objects saves the programmer a great deal of time in having to write code needed to execute one or several commands. However, the VB.NET script management can always be used for managing the more complex logic combinations.

## 10.1. Commands on Event

In order to insert Event objects in the project you need to select the Event Manager resource from the project's tree structure then use the "Add New Event" task ribbon or the right mouse key.

Once a new Event has been added to the project, it then has to be configured by means of using its properties window.



*An Event object's property window*

## Event object properties

Event objects that are inserted in the project need to be configured by means of using their property windows.

### Name

This is used to assign the Event object with a name.

### Enable

This is used to enable or disable the Event object.

### Type

This is used to define the event's function type. There are two options to choose from:

- **Tag** : the event is determined by the state and condition of the tag associated to it.
- **Schedule** : the event is determined by a predefined schedule.

The Event is then configured according to which option has been chosen:

### Enable Tag

This is used to assign a Tag which will enable the Event object's function in runtime.

### Tag

If the **Tag** function type has been selected, you will need to define the tag whose condition will determine the event's activation in runtime according to the property settings described below.

### Condition and Activation Value

This allows you to select the Tag's logic condition which will determine the Event object's activation. As a consequence it may be necessary to establish the 'Activation Value', i.e. the prefixed numerical value with which to verify the activation condition against the value of the reference Tag.

The condition options are:

- **Equals**: the reference tag value becomes equal to the set value.
- **Greater Than**: the reference tag value becomes greater than the set value.
- **Less Than**: the reference tag value becomes less than the set value.



- **Greater Than or Equal:** the reference tag value becomes greater than or equal to the set value.
- **Less Than or Equal:** the reference tag becomes less than or equal to the set value.
- **Not Equal:** the reference tag is not the same as the set value.
- **On Change:** the reference tag changes in respect to the previous value.



The commands are executed only at the rising edge of the set condition.

### Scheduling Type

If you have chose the Schedule option as the event's function type, you will need to define the event's time schedule to determine the event's activation.

The schedule options are cyclic as follows:

- **Every Minute**
- **Every Hour**
- **Every Day**
- **Every Sunday**
- **Every Monday**
- **Every Tuesday**
- **Every Wednesday**
- **Every Thursday**
- **Every Friday**
- **Every Saturday**
- **Every Month**
- **Every Year**

### Date and Time

Based on the type of scheduling set beforehand, you can also set a precise time and date to activate the event accordingly.

### Activation Value

The event's numeric activation value based on the selected condition. This value is taken into consideration when the event's reference tag is numeric type.

### Activation String Value

The event's activation string value based on the selected condition. This value is taken into consideration when the event's reference tag is string type. The use of ID String is not supported in this field.

### Tag Value

When a tag is inserted in this field, the reference value for activating the event will be the value of this tag. This will render the compare value for activating the event dynamic in runtime. When a tag is inserted in this field, the activation values inserted in the above described "Activation Value" and "Activation String Value" will be ignored.

## Event object commands

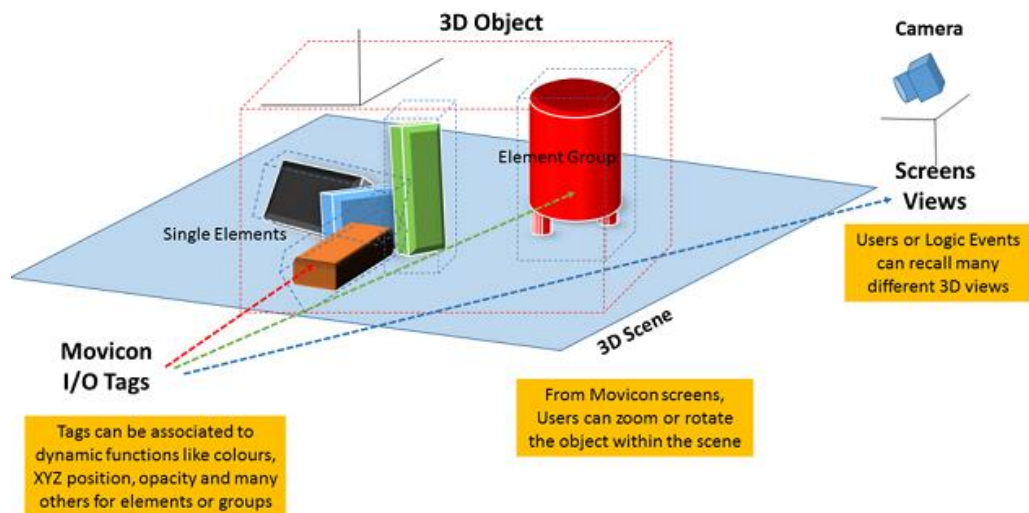
After having set the event's activation modes you will need to define a list of commands to be executed in function with the event.

For details on commands that can be associated to a scheduler, please refer to the commands that can be associated to objects (e.g. command buttons).



# 11. 3D Graphics

Movicon.NExT uses XAML graphics with WPF technology which enables you to insert and use objects with 3D graphics. Each individual 3D symbol component of the object can be animated and the object's observation angle can also be varied. (camera).



This diagram illustrates the concepts used for managing 3D objects within the Movicon.NExT screens. The system accepts previously created 3D models that have been made available for importing into Movicon screens with the aim to render the screens dynamic and interactive with the visualization and supervision system users. Once a 3D model has been inserted, it can be associated with project tags in order to manage the dynamic animation functions. These functions can be associated to each element or group of elements configured within the inserted 3D model scene. This will allow users to create powerful animated graphical interfaces in 3D with objects that can, for instance, change colour according to the associated tag status or move around within the scene in function with the tag values. In addition to this, the operator can interact with the 3D object on screen and rotate or zoom it as pleased. Different 'views' can be stored and recalled for later use. The views may be called on event so that the object can to be positioned appropriately in function with the system's status or events.



Movicon.NExT is not yet designed for editing or creating 3D scenes and graphics models. Movicon's job is to insert this objects on screen and provide real-time animation graphics by linking dynamic Tags to animation types. The Movicon symbol library contains some example 3D graphics models allowing the user use their own 3D models (generally created from Cad 3D or simulation software) by importing and inserting them on screen.



Movicon supports 3D in XAML format which is used to define 2D and 3D vector graphics.



*An example of 3D graphics used in a Movicon.NExT screen*

The WPF 3D technology offers the advantage of being able to use the graphics acceleration hardware to the full, if available. As an alternative, WPF also provides fully automatic software rendering features that greatly simplify 3D management for the user without having to revert to using expensive hardware configurations or setups.

## Importing 3D files on screen

Before going ahead with dynamically managing 3D elements in Movicon.NExT, the 3D model needs to be imported onto the desired screen. The 3D model must be available in XAML being the format supported by Movicon.NExT.

The desired graphics file can be selected and imported to the project using the **"Import XAML"** command from the **"Insert"** group Ribbon to create a new Movicon symbol.



Make sure you specify the file format using the filter in the file browser window.



Warning! If you want to animate the elements of an imported 3D file, you will need to insert the `"ProblematicXamlWriter"` tag in the symbol's code as indicated in the topic on `"Structure of the xaml file to be imported"`.

## 3D elements in the Symbol Library

The Movicon Symbol Library contains some example 3D models that can be inserted in screens and used as needed. The user can also import their own 3D models as described previously, and save them in the symbol library to create their own custom 3D libraries by using the techniques described in the `"Movicon Symbols"` topic to create them.

However, it is also possible to insert the `".xaml"` file of the 3D object directly into the library folder for immediate use. For example by arranging of the `"3DName.xaml"` file of the object you want to use, you can import the file directly by using the `"Import XAML"`

command described above. This file can also be copied directly into the symbol library (for example in the path ".. \ Project1 \ Project1 \ Symbols \ 3DObjects" in order to insert the object in the project library).

What ever mode you use, the 3D object can be made available in the library and inserted on screen using the "drop link" mode (drag connection) that "merge code" (incorporates the code).



The 3D graphics management is not supported during Web Visualization.



Warning! If you need to animate the elements of an inserted 3D file from the Library in "merge code" mode (incorporates code) you must enter the "ProblematicXamlWriter" tag in the symbol code as indicated in the paragraph "Structure of the xaml file to be imported".

## XAML File Structure for importing

If you wish to animate the elements of an imported 3D symbol, you will need to take precautions importing the ".xaml" file beforeforehand. This is done by adding the following tag in the initial code:

Tag = "ProblematicXamlWriter"

For example, if the original file has this structure:

```
<Viewbox x: Name = "MOV3DViewbox" ClipToBounds = "true" xmlns =  
"http://schemas.microsoft.com/winfx/2006/xaml/presentation"  
xmlns: x = "http://schemas.microsoft.com/winfx/2006/xaml" xmlns: d =  
"http://schemas.microsoft.com/expression/interactivedesigner/2006"  
xmlns: c = "http://schemas.openxmlformats.org/markup-compatibility/2006" c:  
Ignorable = "d">
```

.....

Simply add the tag as follows:

```
<Viewbox x: Name = "MOV3DViewbox" ClipToBounds = "true" xmlns =  
"http://schemas.microsoft.com/winfx/2006/xaml/presentation"  
xmlns: x = "http://schemas.microsoft.com/winfx/2006/xaml" Tag =  
"ProblematicXamlWriter"  
xmlns: d = "http://schemas.microsoft.com/expression/interactivedesigner/2006" xmlns: c  
= "http://schemas.openxmlformats.org/markup-compatibility/2006"  
c: ignorable = "d">
```

.....

This precaution will ensure that the object's 3D animation functions correctly when imported by means of using the "Import XAML" command and inserted on screen from the Symbol Library using the "merge code" mode (incorporates code).

This ".xaml" file modification can be done with any text editor.

## 11.1. 3D Animation Graphics

After you have inserted a 3D model on screen, you can then set it with dynamic animation which is done by associating the relevant tags.

The dynamic animation functions can be associated to each individual component of the 3D model, or groups of components. **Group definitions must be predefined in**

**the model beforehand.** Movicon.NExT will only select the group and dynamically animate all the components belonging to it.

## Selecting elements or groups in 3D models

To proceed with selecting a 3D model element you must first enable the "**Enable 3D Edit**" command from the object's command menu. After having enabled the object's edit mode, click on an element to select it or use SHIFT+Click to select a group of elements (predefined in the model beforehand).

Elements will highlight when selected.

## Animation property settings



For further information on "Graphic Animations" please refer to the relating topic.

### 3D X Angle Rotation

The object's angle rotation on the X axis in function with the Tag value.

- **Target Angle:** Sets the rotation value for the selected angle.
- **Center X:** Sets the rotation center point on the X axis.
- **Center Y:** Sets the rotation center point on the Y axis
- **Center Z:** Sets the rotation center point on the Z axis.

### 3D Y Rotation Angle

The object's angle rotation on the Y axis in function with the Tag value.

- **Target Angle:** Sets the rotation value for the selected angle.
- **Center X:** Sets the rotation center point on the X axis.
- **Center Y:** Sets the rotation center point on the Y axis
- **Center Z:** Sets the rotation center point on the Z axis.

### 3D Z Rotation Angle

The object's angle rotation on the Z axis in function with the Tag value.

- **Target Angle:** Sets the rotation value for the selected angle.
- **Center X:** Sets the rotation center point on the X axis.
- **Center Y:** Sets the rotation center point on the Y axis
- **Center Z:** Sets the rotation center point on the Z axis.

### Back Color

Back Color based on the Tag's threshold values. Please refer to the relating topic: "Back Color Animation"

### 3D Move X

The object's linear movement on the X axis in function with the Tag value.

- **Move X:** sets the number in pixels with which the object will be moved along the X axis.

### 3D Move Y

The object's linear movement on the Y axis in function with the Tag value.

- **Move Y:** sets the number in pixels with which the object will be moved along the Y axis.

### 3D Move Z

The object's linear movement on the Z axis in function with the Tag value.

- **Move Z:** sets the number in pixels with which the object will be moved along the Z axis.

### Scale 3D

Object's scale size in function with the Tag value.

- **Object Scale:** Sets the scale factor to reach where 1 refers to the object's standard size.
- **Center X:** Sets the center of the scale factor on the X axis.
- **Center Y:** Sets the center of the scale factor on the Y axis.
- **Center Z:** Sets the center of the scale factor on the Z axis.

### Transparency

The Transparency animation of a 3D object manages the transparency but not the animation time which is always managed as if set at zero.

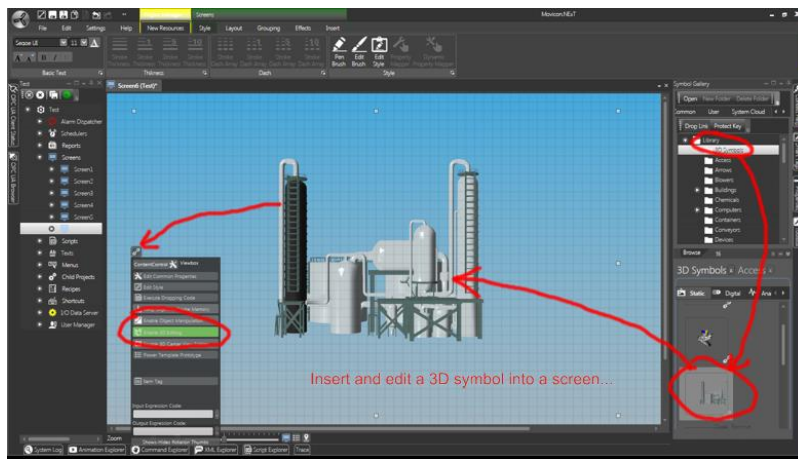
- **Transparency:** Sets the transparency factor to reach where 1 refers to full transparency and 0 corresponds to invisible object.
- **Visible On 3D Camera Zoom:** This property is used for managing the animation with a 3D camera zoom level instead of the associated tag value, to make the object visible or invisible. The value inserted in the "Transparency" property will in this case be used as the zoom factor that when exceeded will make the object invisible. However, please consider that the scale transformation will not be applied to the object but to the camera, therefore the result will be the opposite. This means that when the object is smaller, the scale value is bigger. If "0.5" is inserted as the Transparency value, the object will remain invisible until enlarged by 150%. Conversely, when set with "1.5" the object will become invisible when reduced by 150%.

## 3D Animation example

We shall now look at the basic techniques used for applying graphical animation to objects. To do this we will use the "NITRO ENGINE" object from the symbol library as an example to animate the engine's transmission coupling upon Start-Stop command activation:

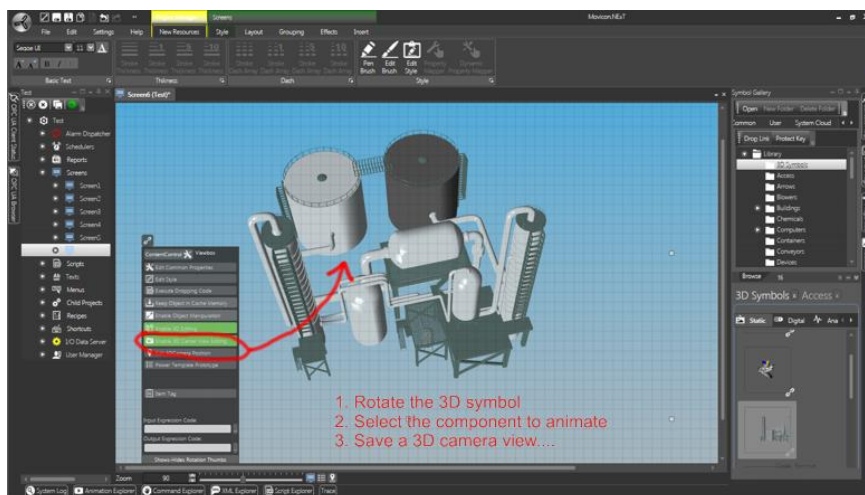
1. Open an existing screen or add a new one.
2. Open the Symbol Gallery and use the mouse click to select the "Refinery" object from the "3D Symbols" group.
3. Position the mouse pointer on an area within the screen and click to release and insert the object. Size and position the object as pleased.
4. Select the object and access the commands menu by clicking the icon located on the object's borders and select and activate the "**3D Edit**" item.





The "3D Edit" button is available from the Object's Commands Tooltip

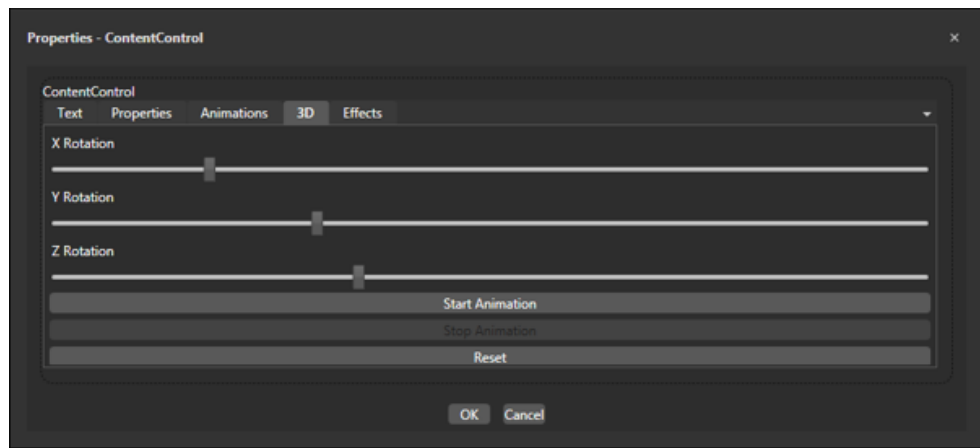
5. Note that when clicking the various 3D symbol components, the selected symbol's parts will highlight with a pulsating color effect to show which parts of the 3D symbol the animations refer to.  
If the 3D symbol is created with '**Groups of components**', the whole group will be selectable using the **CTRL+Click** command.
6. The 3D object can also be rotated while being edited to add other components to it or to record different events viewed (different 3D camera scene positions), which can then be set by the user in runtime with a right mouse click.
  - To use the rotation and recording functions of eventual views, use the "**Enable 3D Camera View**" and "**Edit 3D Camera Position**" commands which are available after activating the "**3D Edit**" item described previously.



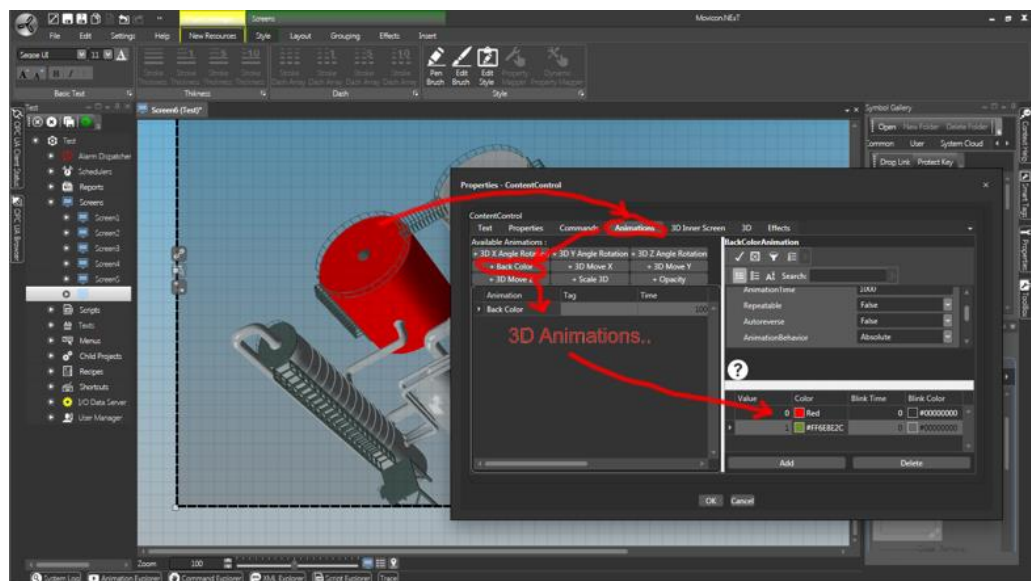
When selecting a symbol component it will animate with a pulsating effect.

- In addition, when the object has been selected as a whole (and not just one of its components) you can use the X, Y, Z rotation cursors from the Common Property Editor's "3D" tab settings to rotate the object or perform a simple rotation simulation.





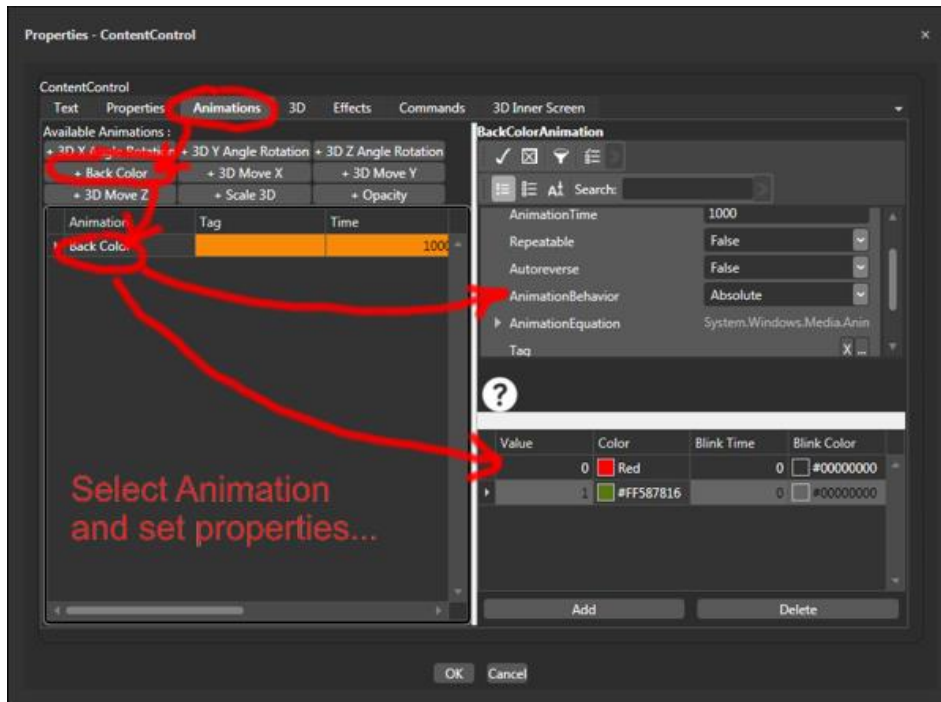
6. After having selected the 3D component model desired for animating, open the object's command menu again and use the "**Common Property Editor**" command.
7. Select the "**Animation**" tab from the dialog window that opens to access the **graphical animation commands** available for individual components and component groups that compose the 3D model and which are listed in the table below.
8. In our example we have used the animated object's Back Color. Any other type of graphical animation can be applied by using the same procedures.



*Window used for selecting Animation type*

10. Go to the "Animation" window and select the desired dynamic animation command. In this example we have selected the "Back Color" animation.
11. When inserting the "Back Color" animation on the list of animations for the selected component, a window will appear on the right to define its settings. In this window it will be possible to insert the colour threshold values by adding the colour thresholds desired to function with the values that will be obtained by the tag (e.g. 0 = the colour red, 1 = the colour green). It will then be possible to select the Tag to associate as well as other parameters as already seen with the other dynamic animations.

- If you do not specify a tag in the Animation's "Tag" field, Movicon will use the tag associated to the object for managing Animations (see paragraph on "Assigning Tags to Objects").

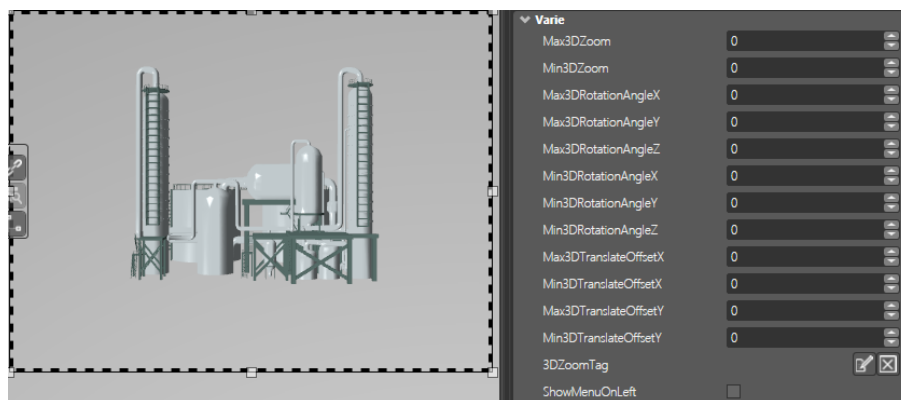


Window used for setting Y Angle Rotation Animations in 3D

12. Close the animation settings window by confirming with "OK".
13. Take a "ToggleButton" from the Toolbox and insert it on the screen to be used for starting and stopping the animation.
14. Assign the button with the same tag defined in the animation settings. By doing this, the 3D object's animation activation tags can be toggles using this button in Runtime.
15. At this point run at test to see if this Animation works by starting up the project in Runtime and use the button to toggle the tags defined. The colour of the component selected should animate according to the tag value.

## 11.2. 3D Symbol Properties

In addition to the Zoom and View properties, the 3D objects can also be set with properties to rotate and move on the X, Y and Z axis as described below.



## General

**Flow Direction:** This is used to set the flow direction of text and other elements within the object.

**Tooltip:** displays the text indicated by the mouse pointer when used by the user on the control.

**Unit Converter:** This is used to set a converter type for the variable associated to the object. Before applying this property set the Unit Converter resource accordingly.

**Max 3D Zoom:** This is used to set the maximum 3D symbol zooming level.

**Min 3D Zoom:** This is used to set the minimum 3D symbol zooming level.

**Max 3D Rotation Angle X:** This is used to set the maximum clockwise rotation angle during runtime. The operator will not be able to manually rotate the object with an angle major to the one set here in respect to the X axis. The allowed values are in decimals from 0 to 1. Zero value = no constraints.

**Max 3D Rotation Angle Y:** This is used to set the maximum clockwise rotation angle during runtime. The operator will not be able to manually rotate the object with an angle major to the one set in this property in respect to the Y axis. The allowed values are 0 and 1 in decimals. Zero value = no constraints.

**Max 3D Rotation Angle Z:** This is used to set the maximum clockwise rotation angle during runtime. The operator will not be able to manually rotate the object with an angle major to the one set in this property in respect to the Z axis. The allowed values are 0 and 1 in decimals. Zero value = no constraints.

**Min 3D Rotation Angle Z:** This is used to set the minimum anti-clockwise rotation angle during runtime. The operator will not be able to manually rotate the object with an angle less than the one set in this property in respect to the X axis. The allowed values are 0 and 1 in decimals. Zero value = no constraints.

**Min 3D Rotation Angle Y:** This is used to set the minimum anti-clockwise rotation angle during runtime. The operator will not be able to manually rotate the object with an angle less than the one set in this property in respect to the Y axis. The allowed values are 0 and 1 in decimals. Zero value = no constraints.

**Min 3D Rotation Angle Z:** This is used to set the minimum anti-clockwise rotation angle during runtime. The operator will not be able to manually rotate the object with an angle less than the one set in this property in respect to the Z axis. The allowed values are 0 and 1 in decimals. Zero value = no constraints.

**Max 3D Translate Offset X:** Maximum 3D object movement translation on the X axis.

**Min 3D Translate Offset X:** Minimum 3D object movement translation on the X axis.

**Max 3D Translate Offset Y:** Maximum 3D object movement translation on the Y axis.

**Min 3D Translate Offset Y:** Minimum 3D object movement translation on the X axis.

**3DZoom Tag:** a variable can be associated to pilot the 3D zoom on the symbol. The value is inversely proportional to the applied zoom. Nearer the value to zero, the more the symbol will increase in size. Greater the value, the smaller the symbol will become (A 0 the symbol will result invisible). If you used the "-" sign, the symbol will rotate 180 degrees without influencing the applied zoom level.

**UseIntellisense:** This is active for default and when deactivated you will need to use "GetVariableValue" e "SetVariable" to access to variables. This option is deactivated to improve performances when opening screens.

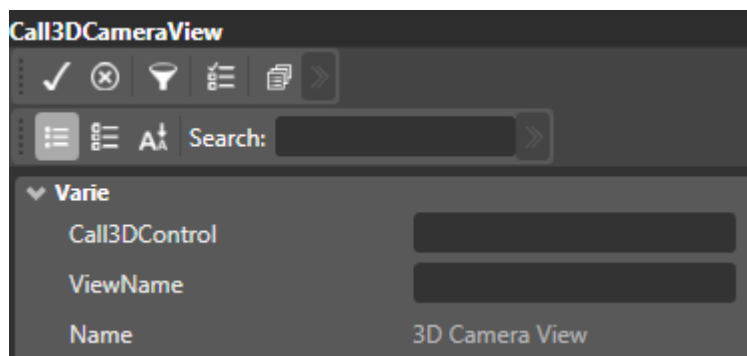
**Menu Activation:** This is used to set the name of the contextual menu that displays when clicking on the object with the right mouse key.

**Menu with Left button:** enables the object's menu to display by using the mouse left button.

### 3D Object Commands

In addition to the classic commands that can be associated to objects, where 3D symbols are concerned, the **3D Camera View** commands can also be accessed by selecting the part desired of the object using the mouse.

This will permit you to call a 3D view that was previously configured within the symbol using the following properties:



- **Call3DControl:** Sets the 3D command to be called
- **ViewName:** Sets the view name to be called
- **Name:** Name of the object (cannot be changed)

## 11.3. Inner Screens for 3D Animation Graphics

### Inner Screen Animation

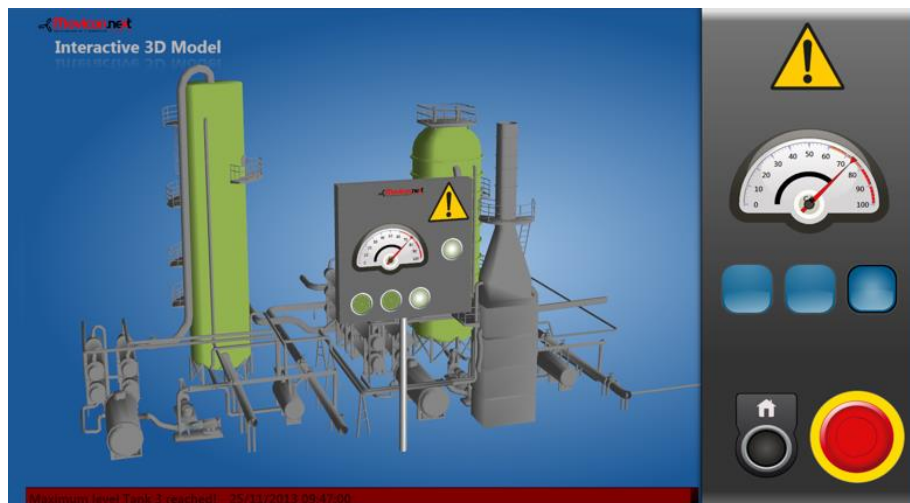
Movicon provides a very interesting feature called Inner Screen to be used in 3D objects. This feature enables you to manage a whole screen within the 3D model, which will be placed on the 'face' of the 3D model's components inserted on screen.

In this way the associated screen will be displayed within the 3D model and will be dynamic in both component content representations and command object interactivity.

- Example: predispose a 600 x 600 pixel sized screen containing a control keypad, display objects, gauges and start/stop buttons all associated with the relevant variables. After having accomplished this you can go ahead and insert this small screen within the 3D model using the "Inner Screen" tab from the object's Common Property Window after having selected the 3D component in which the screen is to be inserted. When in project runtime it will be possible to rotate the 3D model and see the 600x600 screen in its position and fully interactive within the 3D model.



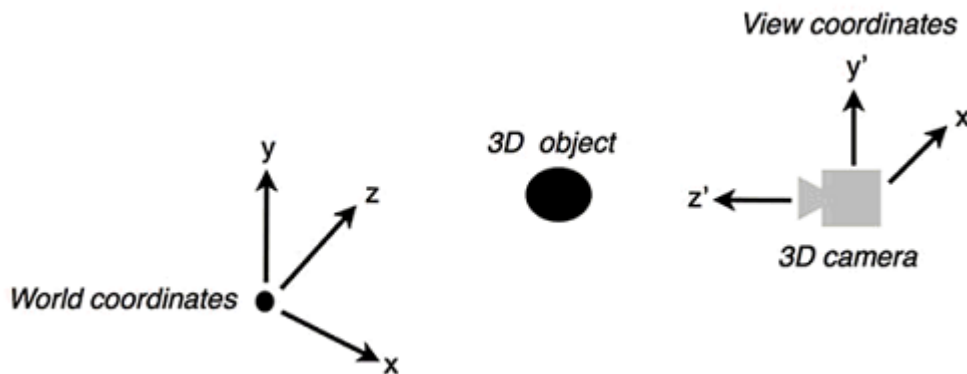
*This example shows a screen with display and command objects that will be managed as an 'Inner Screen' within a 3D model scene*



*This example shows how a screen can become an Inner Screen 3D model component. The screen functions as a 3D Inner Screen positioned and fully interactive within the model.*

## 11.4. Using the Camera in 3D Graphics

The 3D objects inserted on screen support the 'Zoom' and the 'Views' management according to a freely configurable angle rotation defined "Camera".



### Zooming 3D objects

The displaying of each 3D object inserted on screen can be managed with the 'Zoom In' and 'Zoom Out' commands which refer to the object's scene and not the screen.

In runtime the zoom is applied by pointing and right clicking and keeping the key pressed down on the object to drag the cursor to zoom in and out. When using touch screens, this procedure is done by using the equivalent "pinch & zoom" command.

To perform this operation in edit mode you must first enable the 'Editing' function using the object's command menu and then enable the **"Enable 3D Editing"** and **"Enable 3D Camera View"** items.

### 3D object rotation

The displaying of each 3D object inserted on screen can be managed using the Rotation commands which refer to the scene of each individual object and not the screen containing them.

In runtime the Rotation command is used by pointing and clicking the object and keeping the mouse key down while dragging the cursor.

To perform this same operation in edit mode, first enable the 'Editing' functions using the object's command menu and then the **"Enable 3D Editing"** the **"Enable 3D Camera View"** items.

### Camera settings

More than often it is very handy to record one or more 3D object views so that the runtime user can have different views of the camera's preset angle of vision at their disposition. View recording is performed in project edit mode where the design engineer can position the object's angle of vision as pleased and then used the object's command menu to select **"Edit 3D Camera Position"** item.

This command displays a window through which the user can specify the name to be assigned to the view and thus create a list of recorded display 'positions' that can be selected in runtime.

## **Tips & Tricks**

### **How to activate a Camera in runtime**

During editing mode the custom camera positions can be recorded using the "3D Camera Position" command from the object's command menu. As a consequence, these recorded camera positions can be activated by the user in runtime by right clicking on the object to display a menu that contains a list of the saved camera positions.

### **How to move the Camera of a 3D object in runtime**

Click and drag the object with the left mouse key.

### **How to Zoom a 3D object in runtime**

Click and drag the object with the right mouse key.

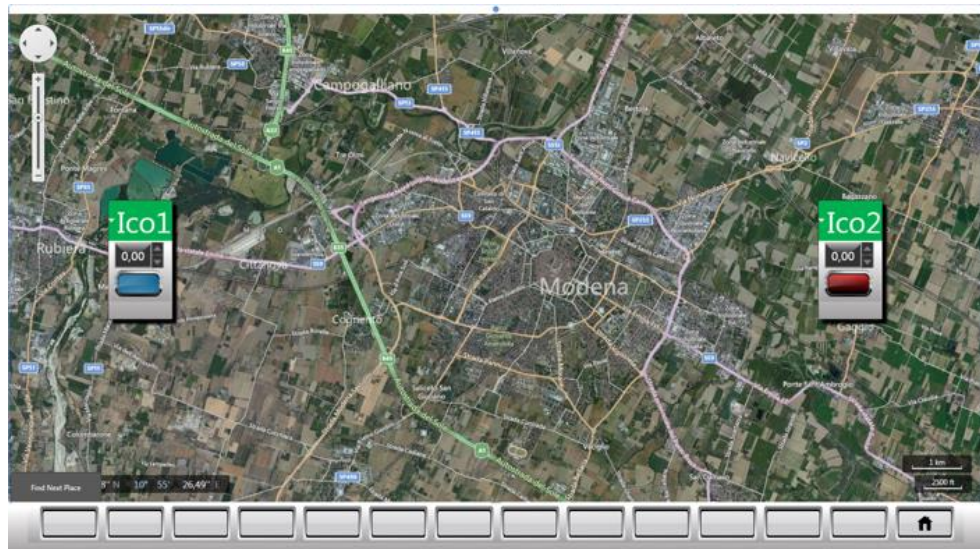




# 12. GeoLocation

## 12.1. Geo Page

Movicon.NExT supports a Geo Page feature which integrates geographic maps and geo-localization managements in screens. This feature is used for creating geo-localized projects to enable the handling of maps within the screens which can be navigated and integrated with dynamic data from the project.

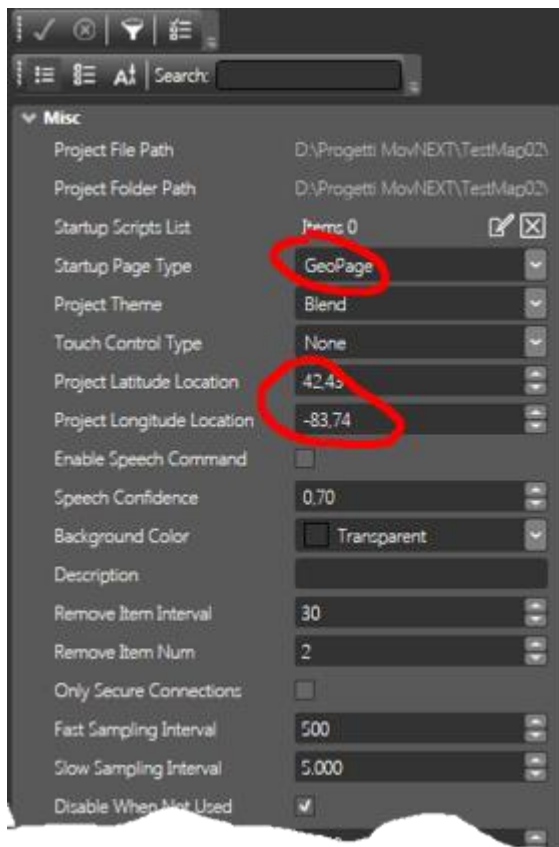


*This is an example of a geo-localization page with dynamic values.*

Starting up the project with the "**Geo Page**" startup page, Movicon will display a geographic map as the opening screen pointing at the latitude and Longitude coordinates specified. One first opening this page the standard zoom factors will be assigned and can be adapted as pleased. The zoom setting will be memorized and used for the opening of the next page.

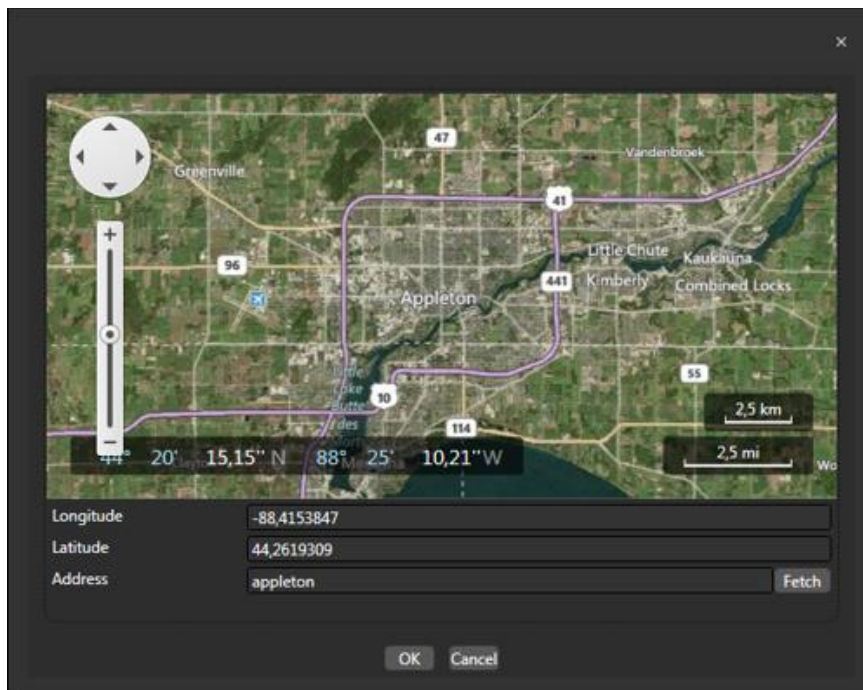
### Creating a Geo Localization project

To be able to manage a geo localization project you must first select the "Geo Page" item as the "Startup Page" type in the project's properties. this selection will determine project startup with the home page showing a geographical map positioned on the coordinated established in the set latitude and longitude properties. The user can select zoom level or navigate the maps freely like any other map management system.



## Editing the geographical coordinate properties

Movicon has provided an **"Edit Geo Location"** tool to facilitate operations for editing the Latitude and Longitude properties for the localization coordinates. This tool can be accessed using the right mouse key on the project root or using the appropriate command from the project Ribbon. The following window will display:



This window is used for navigating the maps and when clicking on any point of the map this tool will return the selected coordinates expressing the **Latitude** and **Longitude**. A

precise address can be entered in the "**Address**" edit box whose coordinates will be searched for when confirming with the "**Fetch**" button. The selected value coordinates can then be used in the project properties.



Using maps in a project requires a connection to the internet services. Movicon uses the **Microsoft Bing® Maps** services for default that are subject to the license terms of use as stipulated by Microsoft at

link: <http://www.microsoft.com/maps/product/terms.html>

However the cartographics and map service provider can be changed to OpenStreetMap.org for example or other.

For further information please see the relating documentation.

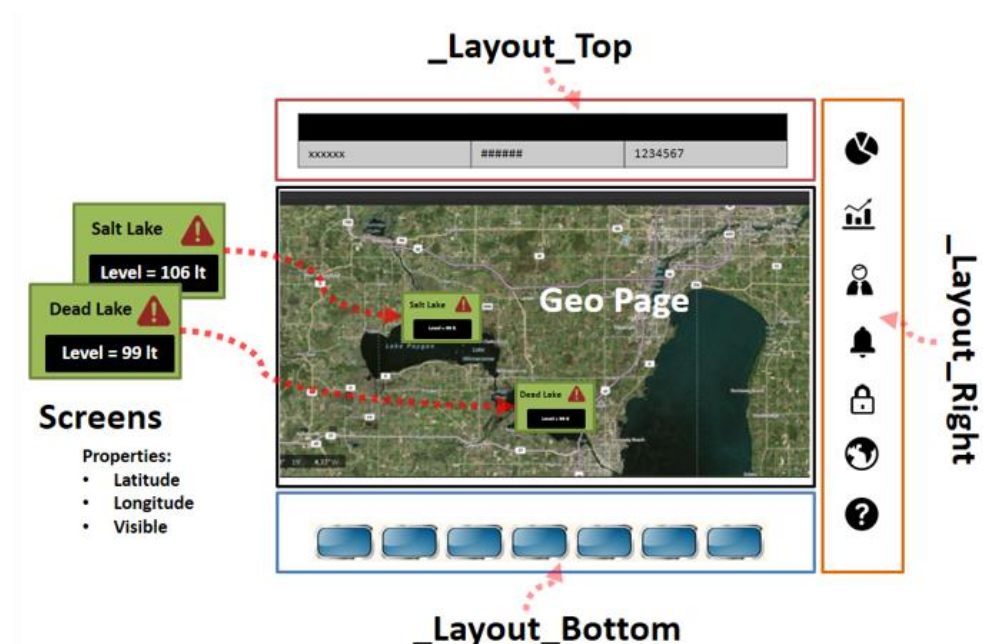
## Screen Layout

As described for Screens the Screen Layout can also be used for creating navigation bars among other things. The page containing maps will therefore be considered the home page to start from and additional screen layouts can be used for those functions which have been coordinated and integrated in the project such as in the initial illustration.

## 12.2. Dynamic Data in Geographic Maps

The Geo Page screen containing geographical maps can be integrated with dynamic information managed by the project. This mechanism is designed to enable dynamic information to be entered passing through screens, the relative properties and contents. The Geo page screen represents contents of other project screens by dynamically displaying them on map at the geographical coordination points defined in the screen's properties.

The image below illustrates the data display criteria on the map:



The designer can therefore create a specific screen with the desired pixel sizes for each dynamic point on the map. The screen will then be displayed on the map dynamically displaying its graphical object contents with command interactivity.

In order to get a screen to display on a map you will need to defined its properties as follows:

<b>Visible</b>	The startup mode of the "Geo Page" project, the Visible option is used for displaying the screen on the map.
<b>Screen Latitude Location</b>	Specifies the latitude coordinates for the screen's location on the map
<b>Screen Longitude Location</b>	Specifies the Longitude coordinates for the screen's location on the map

Any desired number of screens can be created and displayed on various geographical maps for users to navigate, display and interact with the contents dynamically. Therefore, the user will not only be able to view dynamic information but interact with the commands and links to other pages containing detailed information as well.



Please be reminded that the screen will be shown on the map with the sizes specified in Pixels.  
Page navigation requires both a home page and the page with the map.

# 13. MultiTouch

## 13.1. Using multitouch

The modern user interface systems can be equipped with touch screens that support multitouch, as used smartphone or tablet systems, to offer new ways of using industrial HMI systems that are more intuitive and enjoyable to use.

The Movicon.NExT user interface is predisposed natively for Multitouch use.

This Tutorial quickly shows you which are the concepts and the possibilities that can be used to design engineering Movicon screens.

### 13.1.1. Manipulating objects

Multitouch use can also be enabled individually for any one of the Movicon graphical objects. By doing this, the user can only manipulate those objects predisposed with the multitouch feature by the programmer for repositioning, enlarging, rotating them on screen as pleased during runtime.

To enable objects with "multitouch" manipulation simply select the desired object, activate the command menu using the icon located on its border, and select the **"Enable Object Manipulation"** from the command menu as shown in the image below:



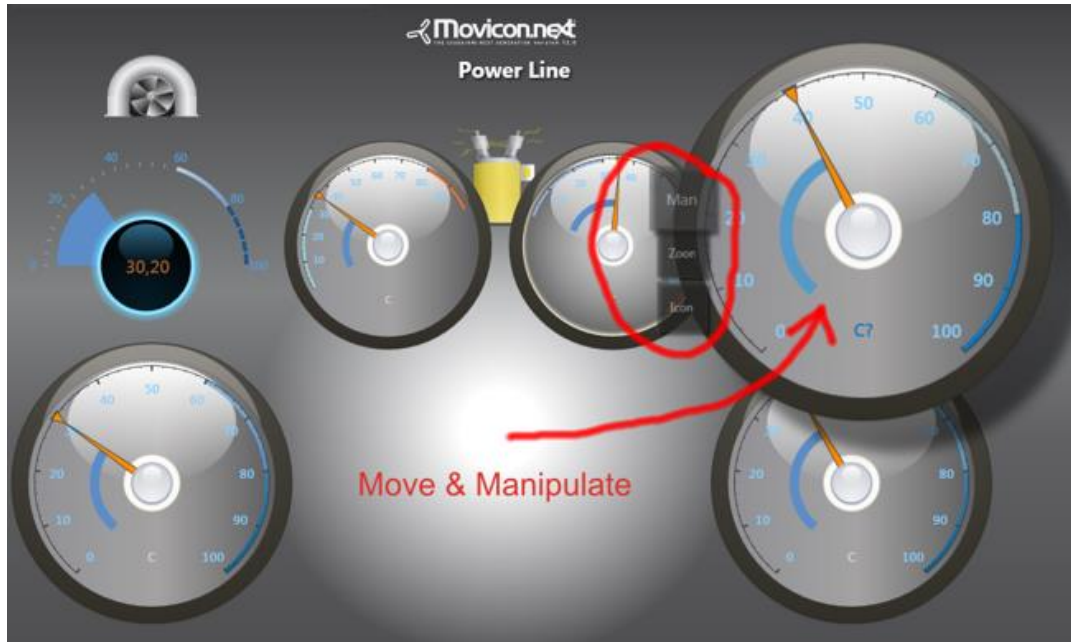
*This example shows you how to enable the touch manipulation of graphical Movicon objects in design mode.*

**During runtime**, "Manipulation" enabled objects will be predisposed for manipulation within the user interface by using the following procedures:



Click the object desired to display the **command buttons on the left border**. Press the **"Man."** button to enable the object's manipulation. The object will obtain a shadowed background to indicated that is manipulable. The "Man." button is bistable meaning that you must press it again to disable the object's manipulation.

When the object is enabled with manipulation, it can be dragged, positioned, rotated or enlarged as pleased using the above described multitouch techniques but with individual objects.



*This image shows and example of a Gauge object enabled with 'touch' manipulation during Movicon runtime using buttons.*

To **restore position**, as defined in design mode, simple use the restore command found on the screens' system command bar which appears with a double click on the top screen border as shown below:

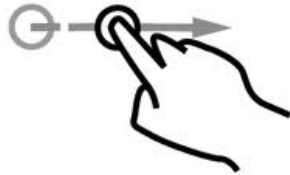


*The command in the Movicon system command bar, shown in this example, enables you to restore manipulated objects back to their original positions.*

## 13.2. Multi Touch Gestuality

This guide will briefly go through the concepts and possibilities used in design engineering Movicon screens.

### Swipe



Swiping is one of the most intuitive and natural user interface experiences in HMI systems where page navigation is much smoother compared to traditional systems.

All you need to do is simply place your fingertip lightly on screen and move it gently to the left or right in order to scroll to the next or previous pages. In order to return to the main page simply tap at the bottom to display the hidden toolbar if not provided otherwise. The main page supports navigation using the touchscreen or scrolling techniques as predefined in the project's general property settings (described in the topic on "Screen Navigation").

### Pinch & Zoom, Pan



The Pinch & Zoom command is used with two fingertips on the screen to produce open and close gestures. This command produces a Zoom In or Zoom Out effect to enlarge or reduce the screen.

The Zoom effect usually enables you to scroll within the panned area or visualize design objects and elements that cannot be seen in normal displays considering that all objects have display properties which function according to the zoom value given to the screen.

### Simultaneous Multitouch Support



A very useful feature to have in the industrial ambient is the possibility to execute commands that have been conditioned to respond to the simultaneous touch of two graphical command objects. For example, to safeguard operators it might be required that an operating command, used for putting dangerous machinery into motion, be invoked by obliging the operator to use both hands to press two buttons, placed at opposite sides of the screen, simultaneously.



The function to simultaneously press two objects is only applied if the command is impulsive type on variable. For all the other command types, only the command of the object pressed first will be executed. In addition, the multitouch function only works for the first two simultaneous touches. This function does not support three or more touches simultaneously but only two at the same time.





# 14. String and Multilanguages

## 14.1. Text strings

Generally a graphical interface project has to display and manage text strings to be displayed in screens as simple text and as headings, names and text in objects. The text strings can be freely typed in any object directly by clicking on the area provided, or entered in the property window fields. Texts in screens can also be managed using the **"Text"** viewer object available from the basic shapes section in the **Toolbox**.

Some Platform.NExT objects support the association of texts to be displayed as object titles (e.g. the 'Text' object, or Buttons). In this case static text or String ID can always be associated. The use of String ID is the most preferred because not only does it guarantee the possibility to change languages during runtime, it also allows text to be kept centralized in one container known as the String Table.



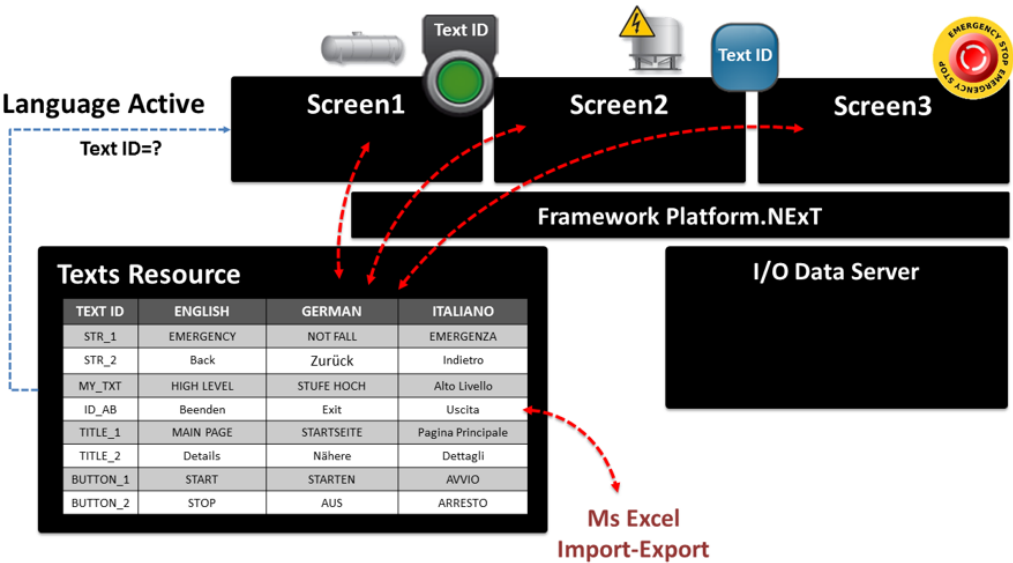
For further information about static text in objects please see the topic on Objects from the ToolBox.



In all circumstances, text in Movicon projects is managed either in static or dynamic mode, according to whether the text is inserted as an **ID identifier** in the text table of the project's String Resource.

### The String Resource

The Resource of the Project's String Tables (Texts) is the container of all the texts that can be dynamically managed and organized in the Movicon project. The String Table is a text table organized in columns which are compatible with Excel and correspond to the ID identifier texts and their languages as shown in the image below:



For each text string inserted on screen, Movicon searches for its corresponding text as an "ID identifier" in the String Table to replace the ID Text with the text contained in the corresponding language column. In this way, by using a univocal string id Movicon will be able to manage the dynamic language change function in runtime by displaying the text contained in the column of the active language.



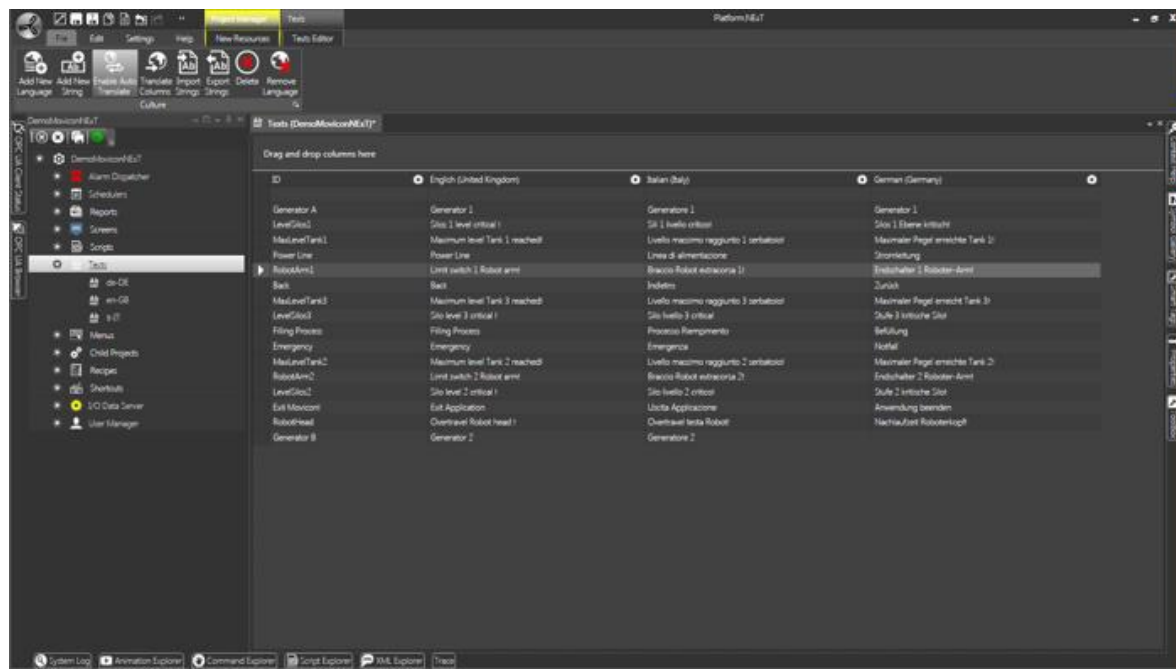
A "Change Language" command can be assigned to any command object to allow the user to active the column of the language desired. All the Movicon texts will dynamically update as a consequence.

The text Strings are needed dynamically managing for example alarm text, headings of objects or windows, any text used in screens or controls that provide use of text. When there is the possibility to manage text in the project's user interface, this text can be kept in the String Table and therefore centralized in one point to be managed dynamically in function with the language desired.



Remember that any text entered directly in Movicon drawings or components without any corresponding ID existing in the String Table will be represented how it was written in static mode in the source representing them and therefore will not be subject to change language.

To open a project String editor double click on the "Texts" in the Project Explorer Window.



The String Editor window appears in table format with an "ID" column and a column for each language inserted



Movicon provides the possibility to load the string table from custom files inserted in the project's string folder. For further information please see the topic on: Runtime Strings.



It is possible to use the following "\n" escape sequence to force the text to go to the beginning of the next line. For example the string:

Item 1\nItem 2\nItem 3

in the visualization phase, for example a text box or button will appear as follows:

- Item 1
- Item 2
- Item 3

## 14.2. Text Management

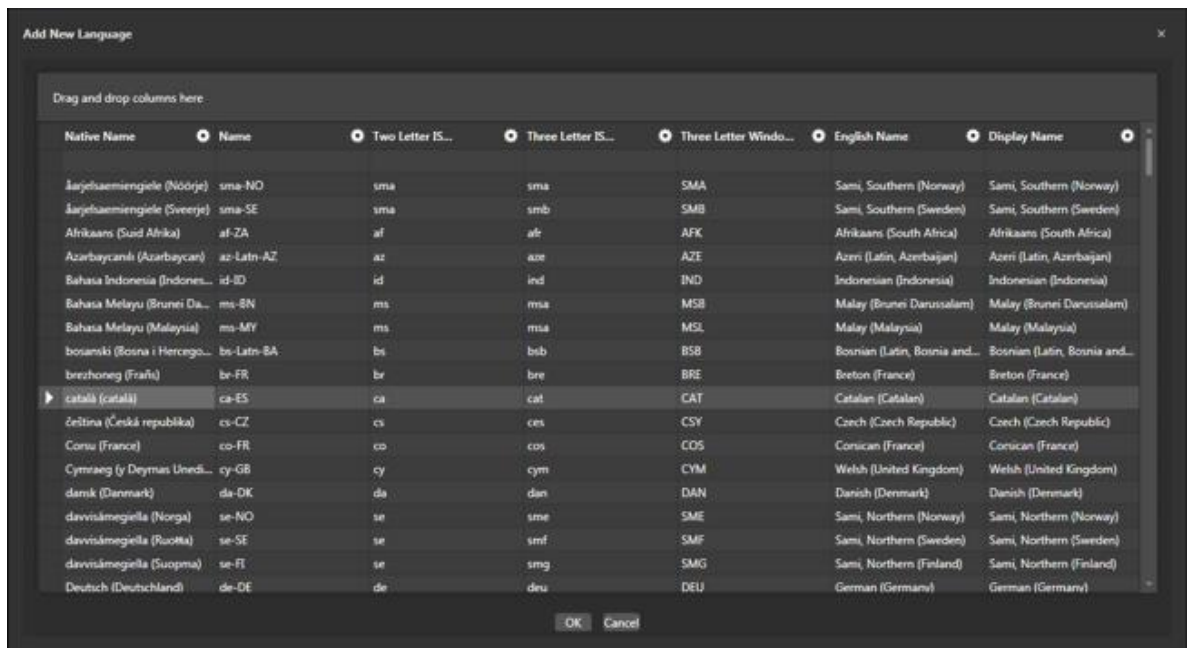
When opening the String Resource for the first time you will need to create at least one column relating to a text language using a window which automatically opens for selecting a language (or Culture). In this window you will find listed all the languages identified by the Windows operating system. Each one of these languages is represented with an identification that reports the language and geographical area. For example, "en-GB" and "en-US". In this case the first part of the identifier represents the language (en = English) and the second part represents the geographical area (GB = Great Britain and US = United States).



The definition of the language name is preset by the operating system and cannot be changed. The language name is conventional and necessary to eventual automatic text translations as we will see below.

### Inserting a new language column

In order to be used the String Table requires at least one text column to be entered referring to the selection of a language. The first time this resource is opened a window for inserting a new language will open automatically. After which all other languages can be inserted using the "Add New Language" command available from the "Text Editor" Ribbon or with a right mouse key click.



The Add New Language command is used for adding new languages to the String Table

This command can be used each time a new language has to be added to the project.



Movicon does not impose any limits on the number of languages that can be inserted in the project. However, the product may not be guaranteed to work correctly with projects using more than 32 language columns at the same time.

## Inserting a new text string

After having created the Columns corresponding to the languages, new text strings can be inserted using the "**Add New String**" command in the "Text Editor - Culture" Ribbon to add a new String ID to the table. The text can be modified by double clicking the ID box to enter into edit mode. The direction arrow or Tab keys can be used to pass over from one table box to another. Below is a screenshot showing the string resource's editing window.



The Add New String command is used for adding a new ID to the String Table

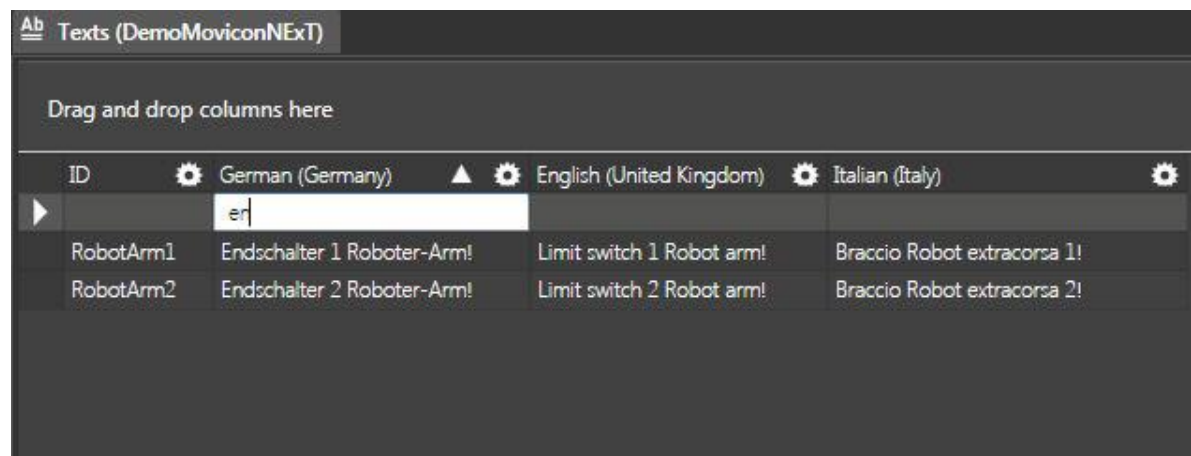
As you can see, the grid is composed of one 'ID' column, which contains the univocal name associated to each string declared. The following columns contain the string texts

translated into the various cultures of which language name is found as the heading of each single column with the following syntax: Language Name (Country).

### Sort by and Filters

The String Table provides the use of Sort by and Filters on the inserted texts. These actions include:

- Dragging columns to filter a specific language
- Ordering any of the column texts in alphabetical order by clicking on the column heading.
- Filtering contents of any column by clicking on the row beneath the interested column heading and typing in the filter text.



An example of a filter operation in the string table

### Runtime Strings

Movicon provides you with the option to load the string table from any custom file inserted in the project string folder instead of loading the one defined in design mode. In this way the project strings can be customized without having to open it in design mode.

The Server and the Client will search for the files in the project's 'Strings' folder that have the ".custstrings" extension. Once the files have been found, they will be imported and used instead of the string table defined in design mode.



In case where there is at least one 'runtime' file, the table defined in design mode will not be loaded. If the runtime file does not include all the strings that were defined in design mode, those not included will be displayed associated with simple 'String IDs' in the object's or resource's text.

The runtime files must have an import file structure for the string editor, being:

1st row: ID;[cultura1];[cultura2],...  
next row: Key;string1;string2,...

A practical file example would be as follows:

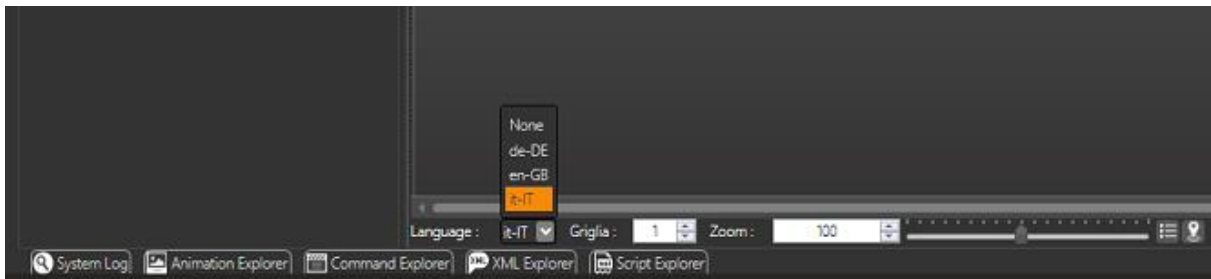
*ID;it-IT;en-US;es-ES*  
*NewTextID\_01;Ciao;Hello;Hola*  
*NewTextID\_02;Allarme;Alarm;Alarma*



The runtime strings are also loaded even when the string table is empty in design mode. However, you will need to define the table in design mode with the cultures that are going to be used and that are also present in the runtime files.

A drop-down menu at the bottom of the screen edit window can also be used to set languages while in development mode.

The selection of languages to choose are those set in the project.



When launching the project into runtime without having selected a language, the project will startup with the language selected in editing mode. However any languages set for commands for the objects will have priority in runtime over the one selected in editing mode.

## 14.3. Automatic Text Translation

Movicon.NExT supports the automatic text translation function. This function can be used when wishing to have an automatic translation of text into any language inserted in the project.

This Movicon feature uses the automatic online translation services and therefore requires that the PC, being used for this operation, be connected to the internet.



The quality of the translation cannot be guaranteed depending on the automatic online translation services provided. Therefore it is advised to check translations with a translator when in doubt to the correct translation.

The Automatic Translation feature can be enabled or disabled using the "On-Off" command from the Ribbon, as described in the paragraph about commands. Therefore you can operate without the automatic translation auxiliary by deactivating the command.

The translation operations can be performed using the following concepts:

### **Active Automatic Translation**

Movicon will automatically translate any text inserted in any column and insert translated text into empty columns.

### Non Active Automatic Translation

No automatic translations will be performed

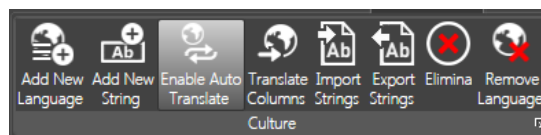
### Translation Command

This command forces the translation of text in a column and requires that the language of origin and the destination language be specified (even when the Automatic Translation is not active).

- **ATTENTION: The translation command overwrites any previously modified text.**

## 14.4. String Editor Commands

The commands for editing strings can be found in the "Text Editor - Culture" Ribbon.



*Ribbon containing commands for editing strings*

### Add New Language

This command is used to insert a new Language Column in the string table. The available languages will be shown in a list (which can be put into order) contained in a pop-up window. Simply click on the language desired and confirm with the 'OK' button to add it to the table.

### Add New String

This command is used to set a new String ID. The string ID will be initialized with a default value that can be changed using the following modes:

- select cell desired and press the "F2" key on keypad
- double click on desired cell
- select cell desired and start digiting text using the keypad.

This command is equivalent to pressing the "Ins" key on the keypad.

### Enable Auto Translate

This command (On-Off) is used to automatically translate the previously edited string into all the languages declared in the string table. In this way, when confirming a text inserted for a specific language, this text will automatically be translated into the other languages defined in the table.



This feature requires an active Internet connection.

### Translate Columns

This command is used to automatically translate all the strings declared in the table by using a pop-up window to select the language to translate and then the columns to translate this language.



Usually this command is used when adding new language columns to the already populated string table.



This feature requires an active Internet connection.

### Delete

This command allows you to delete the selected string from the string table. This feature supports multi-selection with "CTRL+click" and dragging techniques by selecting column row on the left hand side and dragging mouse cursor along the column. This command is equivalent to pressing the keypad 'Canc' key

### Remove Language

This command is used to remove language columns from the string table. This is done by selecting the desired column from the pop-up window that appears and confirming with OK. This command does not support multi-selections.

### Import Strings

This command is used to import text strings in CSV. file format. Movicon will import all the ID and language columns contained in the CSV. file to the string table.



The strings that already exist in the table will automatically be overwritten with those imported from the CSV file. Those language columns that have not been declared in the project but exist in the CSV file will automatically be inserted in the string table.

### Export Strings

This command is used to export the text strings by saving them in CSV. file format. This command supports multiple selections using the "CTRL+click" dragging techniques. This is done by selecting a column row on the left hand side with the "CTRL+click" technique and then moving the mouse cursor along the column to multi-select. If no strings have been selected in the table, this command will export the whole string table and its contents to the file.

## 14.5. Special String ID

Special string ID can be inserted in the Movicon.NExT string table to customize the texts of various objects from the ToolBox in runtime. These particular string id comes in two parts. The first part contains the Place folder with the name of the control ( `_ {ControlName}_` ) and the second part contains the control's reference text to be customized e.g:

### AlarmWindow\_Message

In this case, the text to be inserted in the box that corresponds to the language will change the original text in the object when in runtime. In order to change the language in the project during runtime, you will need to use a change language command associated to a button.



ID	Italian (Italy)	English (United Kingdom)
_AlarmWindow_Message	ITA m3ss4g3	m3ss4g3
_AlarmWindow_AckAllCmd		
New Text ID		

The replaced "ITA mess4g3" text will show in the AlarmWindow object's message column when changing the project language to Italian (Italy).



The special string ID (the only ones that can start with a \_ character) have priority over the other string ID. This means that if you have 2 strings that change the same text, where the first one is a normal string id and the second one is a special string id, only the second one will be used to change the text.

Below is a list of Special String ID that are currently available and listed by control type:

**Alarm Viewers**

Alarm Window	Auditing viewer	Alarm Banners
_AlarmWindow_AckAllCmd _AlarmWindow_AckCmd _AlarmWindow_AckedTransitionTime _AlarmWindow_ActiveTransitionTime _AlarmWindow_ActualSettings _AlarmWindow_AddCommentCmd _AlarmWindow_BestFit _AlarmWindow_BranchText _AlarmWindow_ChildProject _AlarmWindow_CollapseAll _AlarmWindow_Comment _AlarmWindow_Condition _AlarmWindow_ConfirmAllCmd _AlarmWindow_ConfirmCmd _AlarmWindow_ConfirmedTransitionTime	_Auditing_ActualSettings _Auditing_BestFit _Auditing_EditSettings _Auditing_EnabledState _Auditing_Message _Auditing_Method _Auditing_RefreshCmd _Auditing_ResetSettingsTooltip _Auditing_Source _Auditing_Time _Auditing_Type	_BannerAlarmWindow_NextAlarmCmd _BannerAlarmWindow_PrevAlarmCmd _BannerAlarmWindow_RefreshCmd

_AlarmWindow_DisableSoundCmd _AlarmWindow_EditSettings _AlarmWindow_ExecuteComandsCmd _AlarmWindow_ExpandAll _AlarmWindow_Message _AlarmWindow_Quality _AlarmWindow_Reason _AlarmWindow_RefreshCmd _AlarmWindow_ResetSettingsTooltip _AlarmWindow_ServerTitle _AlarmWindow_Severity _AlarmWindow_ShelveCmd _AlarmWindow_ShelvingTransitionTime _AlarmWindow_Source _AlarmWindow_State _AlarmWindow_Time _AlarmWindow_UnshelveCmd _AlarmWindow_Unshelved		
<b>Historical Log Viewers</b>		
_HistoricalEvents_ActualSettings _HistoricalEvents_All _HistoricalEvents_BestFit _HistoricalEvents_ComboAlarms _HistoricalEvents_ComboAll _HistoricalEvents_ComboDrivers _HistoricalEvents_ComboSystem _HistoricalEvents_DateTimeEnd _HistoricalEvents_DateTimeStart _HistoricalEvents_Day _HistoricalEvents_EditSettings _HistoricalEvents_EventComment _HistoricalEvents_EventDate _HistoricalEvents_EventDetails _HistoricalEvents_EventMessage _HistoricalEvents_EventOccurence _HistoricalEvents_EventSequence _HistoricalEvents_EventState _HistoricalEvents_EventTime _HistoricalEvents_EventType _HistoricalEvents_EventUtcDate _HistoricalEvents_EventUtcTime _HistoricalEvents_ExportDataCommand _HistoricalEvents_Hour _HistoricalEvents_Minute _HistoricalEvents_Month _HistoricalEvents_RefreshDataCommand _HistoricalEvents_ResetSettingsTooltip _HistoricalEvents_Severity _HistoricalEvents_SourceName _HistoricalEvents_UserName _HistoricalEvents_Week _HistoricalEvents_Year		

## Trends

Chart	Chart XY	Data Analysis
_Chart_LegendAreaVisible _Chart_PushTitle _Chart_ResetTitle _Chart_SeriesLabelVisibleTitle _Chart_SimplePieTitle _Chart_Style3DTitle _Chart_Zoom3DTitle	_ChartXY_LegendAreaVisible _ChartXY_PushTitle _ChartXY_ResetTitle _ChartXY_SeriesLabelVisibleTitle	_DataAnalysis_Hour _DataAnalysis_ActualSettings;ActualSetting;ITA Actual Setting _DataAnalysis_AllTitle _DataAnalysis_averageValue _DataAnalysis_ClearRecentTitle _DataAnalysis_ColorColumn _DataAnalysis_CompareTitle _DataAnalysis_CrossHairTitle

		_DataAnalysis_Day _DataAnalysis_DayTitle _DataAnalysis_EditSettings _DataAnalysis_EndDate _DataAnalysis_ExpandTitle _DataAnalysis_FetchTitle _DataAnalysis_GridDataTooltip _DataAnalysis_GridTagName _DataAnalysis_GridTitle _DataAnalysis_HistoricalNameColumn _DataAnalysis_Hour _DataAnalysis_HourTitle _DataAnalysis_LabelTitle _DataAnalysis_LegendTitle _DataAnalysis_MaxRecordTitle _DataAnalysis_maxValue _DataAnalysis_medianValue _DataAnalysis_MinTitle _DataAnalysis_Minute _DataAnalysis_minValue _DataAnalysis_Month _DataAnalysis_MonthTitle _DataAnalysis_NameTitleColumn _DataAnalysis_NextTitle _DataAnalysis_numCompressPoint _DataAnalysis_numCompressRation _DataAnalysis_numPoints _DataAnalysis_PrevTitle _DataAnalysis_PrintTitle _DataAnalysis_RecentTimeRanges _DataAnalysis_RefreshTitle _DataAnalysis_ResetSettingsTooltip _DataAnalysis_SourceTimeStamp _DataAnalysis_standardDeviationValue _DataAnalysis_StartDate _DataAnalysis_StatisticsLineColumn _DataAnalysis_StatisticsTitle _DataAnalysis_TagNameColumn _DataAnalysis_TimeRange _DataAnalysis_TodayBtnTooltip _DataAnalysis_TomorrowBtnTooltip _DataAnalysis_Value _DataAnalysis_varianceValue _DataAnalysis_VisibleColumn _DataAnalysis_Week _DataAnalysis_WeekTitle _DataAnalysis_Year _DataAnalysis_YearTitle
<b>Real Time Trend</b>		
_RealTimeTrend_ActualSettings _RealTimeTrend_EditSettings _RealTimeTrend_ExpandTitle _RealTimeTrend_GridDataTooltip _RealTimeTrend_GridTagName _RealTimeTrend_GridTitle _RealTimeTrend_LegendTitle _RealTimeTrend_MaxColumn _RealTimeTrend_MinColumn _RealTimeTrend_PenColorColumn _RealTimeTrend_PenNameColumn _RealTimeTrend_PrintTitle _RealTimeTrend_ResetSettingsTooltip _RealTimeTrend_SourceTimeStamp _RealTimeTrend_Value _RealTimeTrend_ValueColumn _RealTimeTrend_VisibleColumn _RealTimeTrend_YearTitle		

## Viewers

Cross Reference List Viewer	DB Connector Grid Viewer	Recipe Viewer
_CReDitor_Container _CReDitor_EntityReadablePath _CReDitor_FlatTitle _CReDitor_HierarchicalTitle _CReDitor_IsNotRefValid _CReDitor_IsNotValid _CReDitor_Name _CReDitor_ReadablePath _CReDitor_TabScreenTitle _CReDitor_TabTagsTitle _CReDitor_TagName	_DBControl_ActualSettings _DBControl_BestFit _DBControl_Column _DBControl_Column1 _DBControl_Column2 _DBControl_Column3 _DBControl_DeleteRow;DeleteRow;ITA_DeleteRow _DBControl_EditSettings _DBControl_ExportDataCommand _DBControl_InsertRow _DBControl_LocalTimeCol _DBControl_MillisecondsCol _DBControl_ReasonCol _DBControl_ReloadData _DBControl_ResetSettingsTooltip _DBControl_SaveData _DBControl_UserCol _DBControl_UtcTimeCol	_RecipeViewer_Add_a_new_recipe_to_the_archive _RecipeViewer_DataValue_01 _RecipeViewer_DataValue_02 _RecipeViewer_Export_selected_recipe_values_to_csv_file _RecipeViewer_Group_01 _RecipeViewer_Group_02 _RecipeViewer_Import_recipe_values_from_csv_file _RecipeViewer_Read_the_data_values_from_device _RecipeViewer_Recipe_Commands _RecipeViewer_Recipe1 _RecipeViewer_Reload_all_Recipes_values_from_archive _RecipeViewer_Remove_the_current_recipe_from_archive _RecipeViewer_Save_all_recipe_changes_in_the_archive _RecipeViewer_Write_the_actual_data_values_to_device
Scheduler Viewer	Scheduler Viewer	
_SchedulerControl_AddCalendarItem _SchedulerControl_AddScheduleTime _SchedulerControl_Calendar _SchedulerControl_CalendarItem Type _SchedulerControl_Date _SchedulerControl_DateOff _SchedulerControl_DayMonth _SchedulerControl_DayOfWeek _SchedulerControl_DeleteCalendarItem _SchedulerControl_EditCalendarItem _SchedulerControl_EnableEvent _SchedulerControl_EndDate _SchedulerControl_EndTime _SchedulerControl_EveryDay _SchedulerControl_EveryFriday _SchedulerControl_EveryHour _SchedulerControl_EveryMinute _SchedulerControl_EveryMonday _SchedulerControl_EverySaturday _SchedulerControl_EverySunday _SchedulerControl_EveryThursday _SchedulerControl_EveryTuesday _SchedulerControl_EveryWednesday _SchedulerControl_Month _SchedulerControl_NewEventAccessLevel _SchedulerControl_NewEventAccessMask	CalendarDayType_Any CalendarDayType_Friday CalendarDayType_Monday CalendarDayType_Saturday CalendarDayType_Sunday CalendarDayType_Thursday CalendarDayType_Tuesday CalendarDayType_Wednesday CalendarItemType_DateRange CalendarItemType_SingleDate CalendarItemType_WeekNDate CalendarMonthType_Any CalendarMonthType_April CalendarMonthType_August CalendarMonthType_December CalendarMonthType_Even CalendarMonthType_February CalendarMonthType_January CalendarMonthType_July CalendarMonthType_June CalendarMonthType_March CalendarMonthType_May CalendarMonthType_November CalendarMonthType_October CalendarMonthType_Odd CalendarMonthType_September CalendarWeekType_Any CalendarWeekType_Fifth CalendarWeekType_First CalendarWeekType_Fourth CalendarWeekType_Last CalendarWeekType_Second CalendarWeekType_Third _SchedulerControl_MonthDayAny _First	

_SchedulerControl_NewEventAccessRole _SchedulerControl_NewEventCalendar _SchedulerControl_NewEventEnableTag _SchedulerControl_NewEventName _SchedulerControl_NewEventTag _SchedulerControl_NewEventTitle _SchedulerControl_NewEventType _SchedulerControl_NewEventValueOff _SchedulerControl_NewEventValueOn _SchedulerControl_NewEventWeeklyPlan _SchedulerControl_ReloadData _SchedulerControl_SchedulerName _SchedulerControl_SchedulerSave _SchedulerControl_SelectableRuntime _SchedulerControl_StartDate _SchedulerControl_StartTime _SchedulerControl_Time _SchedulerControl_TimeOff _SchedulerControl_Type;Type;ITA _Type _SchedulerControl_Type _SchedulerControl_WeeklyItemAddEvent _SchedulerControl_WeeklyItemEditFormCancel _SchedulerControl_WeeklyItemEditFormDelete _SchedulerControl_WeeklyItemEditFormEndTime _SchedulerControl_WeeklyItemEditFormOk _SchedulerControl_WeeklyItemEditFormStartTime _SchedulerControl_WeeklyPlan _SchedulerControl_WeekOfMonth _SchedulerControl_NewEventExceptions _SchedulerControl_DeleteCalendarItems	_SchedulerControl_MonthDayAny _Last _SchedulerControl_MonthDayAny _NoDate MonthDayAny_First MonthDayAny_Last MonthDayAny_NoDate
--	---

## Historical Data Viewers

Data Logger Viewer	Historian Viewer	System Log Viewer
_DataloggerViewer_ActualSettings _DataloggerViewer_All _DataloggerViewer_BestFit _DataloggerViewer_Column _DataloggerViewer_Column1 _DataloggerViewer_Column2 _DataloggerViewer_Column3 _DataloggerViewer_DateTimeEnd _DataloggerViewer_DateTimeStart _DataloggerViewer_Day _DataloggerViewer_EditSettings	_HistoricalViewer_ActualSettings _HistoricalViewer_All _HistoricalViewer_BestFit _HistoricalViewer_DateTimeEnd _HistoricalViewer_DateTimeStart _HistoricalViewer_Day _HistoricalViewer_dValue _HistoricalViewer_dValueBefore _HistoricalViewer_EditSettings _HistoricalViewer_ExportDataCommand _HistoricalViewer_Hour	_LogViewer_ActualSettings _LogViewer_App _LogViewer_BestFit _LogViewer_Class _LogViewer_Details _LogViewer_EditSettings _LogViewer_File _LogViewer_ForceNewLog _LogViewer_FullLogPath _LogViewer_HostName _LogViewer_Identity

_DataloggerViewer_ExportDataCommand _DataloggerViewer_Hour _DataloggerViewer_LocalTimeCol _DataloggerViewer_MillisecondsCol _DataloggerViewer_Minute _DataloggerViewer_Month _DataloggerViewer_ReasonCol _DataloggerViewer_RefreshDataCommand _DataloggerViewer_ResetSettingsTooltip _DataloggerViewer_UserCol _DataloggerViewer_UtcTimeCol _DataloggerViewer_Week _DataloggerViewer_Year	_HistoricalViewer_Minute _HistoricalViewer_Month _HistoricalViewer_Name _HistoricalViewer_RecordDateTime _HistoricalViewer_RecordDateTimeMilliseconds _HistoricalViewer_RefreshDataCommand _HistoricalViewer_ResetSettingsTooltip _HistoricalViewer_ServerPicoSeconds _HistoricalViewer_ServerTimeStamp _HistoricalViewer_ServerTimeStampMilliseconds _HistoricalViewer_SourcePicoSeconds _HistoricalViewer_SourceTimeStamp _HistoricalViewer_SourceTimeStampMilliseconds _HistoricalViewer_Status _HistoricalViewer_UserName _HistoricalViewer_Value _HistoricalViewer_ValueBefore _HistoricalViewer_Week _HistoricalViewer_Year	_LogViewer_Item _LogViewer_JustFileName _LogViewer_Level _LogViewer_Line _LogViewer_LogFile _LogViewer_LogFile1 _LogViewer_Logger _LogViewer_Loggers _LogViewer_MachineName _LogViewer_Message _LogViewer_Method _LogViewer_NDC _LogViewer_RefreshData _LogViewer_ResetSettingsTooltip _LogViewer_SelectFirst _LogViewer_SelectLast _LogViewer_ShortLogFile _LogViewer_Thread _LogViewer_Throwable _LogViewer_TimeStamp _LogViewer_UserName
Event Trace Viewer		
_ServerHistory_ActualSettings _ServerHistory_AggregationType _ServerHistory_AllTitle _ServerHistory_BestFit _ServerHistory_ControlViewModel _ServerHistory_Day _ServerHistory_DayTitle _ServerHistory_EditSettings _ServerHistory_EndDate _ServerHistory_EndMaxReturnValues _ServerHistory_EndTime _ServerHistory_ExpandTitle _ServerHistory_FetchBtn _ServerHistory_Hour _ServerHistory_HourTitle _ServerHistory_LabelTitle _ServerHistory_LegendTitle _ServerHistory_MinTitle _ServerHistory_Minute _ServerHistory_Month _ServerHistory_MonthTitle _ServerHistory_NextBtn _ServerHistory_NextTitle _ServerHistory_PrevBtn _ServerHistory_PrevTitle _ServerHistory_PrintTitle _ServerHistory_ReadType _ServerHistory_ResetSettingsTooltip _ServerHistory_ServerPicoSeconds _ServerHistory_ServerTimestamp _ServerHistory_SourcePicoSeconds _ServerHistory_SourceTimestamp _ServerHistory_StartDate _ServerHistory_StartTime _ServerHistory_StatusCode _ServerHistory_StopBtn _ServerHistory_Value _ServerHistory_WeekTitle _ServerHistory_YearTitle		

## DB Connectors

Pivot Grid
_PivotGrid_ActualSettings
_PivotGrid_BestFit
_PivotGrid_CollapseAll
_PivotGrid_EditSettings
_PivotGrid_EventDateTime
_PivotGrid_EventDateTimeUtc
_PivotGrid_EventDetails
_PivotGrid_EventId
_PivotGrid_EventMessage
_PivotGrid_EventType
_PivotGrid_ExpandAll
_PivotGrid_ExportDataCommand
_PivotGrid_OID
_PivotGrid_ReloadData
_PivotGrid_ResetSettingsTooltip
_PivotGrid_SourceName
_PivotGrid_SourceNode

## Dialog Window

log-in User window
_UserDialog_CancelLabel
_UserDialog_CloseLabel
_UserDialog_OkLabel
_UserDocument_PasswordExpired
_UserEditorControl_AddNewRoleCommandDescription
_UserEditorControl_AddNewRoleCommandText
_UserEditorControl_AddNewRoleCommandTooltip
_UserEditorControl_AddNewUserCommandDescription
_UserEditorControl_AddNewUserCommandText
_UserEditorControl_AddNewUserCommandTooltip
_UserEditorControl_AdminRoleName
_UserEditorControl_DragAndDropHere
_UserEditorControl_EditMenuHeader
_UserEditorControl_GuestRoleName
_UserEditorControl_PageSplitterBottom
_UserEditorControl_PageSplitterTop
_UserEditorControl_PowerUserRoleName
_UserEditorControl_TreeViewHeader
_UserEditorControl_UserAndRoles
_UserEditorControl_UserAndRolesSharedRepo
_UserManagement_AccessMaskEditor
_UserManagement_CancelLabel
_UserManagement_LoginUserName
_UserManagement_LoginWindowTitle
_UserManagement_LogoutUserName
_UserManagement_OkLabel
_UserManagement_OldPassword
_UserManagement_Password
_UserManagement_PasswordConfirm
_UserManagement_PasswordLabel
_UserManagement_RequestedLevel
_UserManagement_RequestedRole
_UserManagement_RoleAccessLevel
_UserManagement_RoleAccessMask
_UserManagement_RoleCultureName
_UserManagement_RoleName
_UserManagement_RoleProperties
_UserManagement_UserAccessLevel
_UserManagement_UserAccessMask
_UserManagement_UserCultureName
_UserManagement_UserElectronicSignature

\_UserManagement\_UserEmail  
\_UserManagement\_UserForcePasswordChangeFirstLogin  
\_UserManagement\_UserMobile  
\_UserManagement\_UserName  
\_UserManagement\_UserNameLabel  
\_UserManagement\_UserPassword  
\_UserManagement\_UserPasswordConfirm  
\_UserManagement\_UserPasswordExpiresInDays  
\_UserManagement\_UserPhoneNumber  
\_UserManagement\_UserProperties



# 15. Command Menu

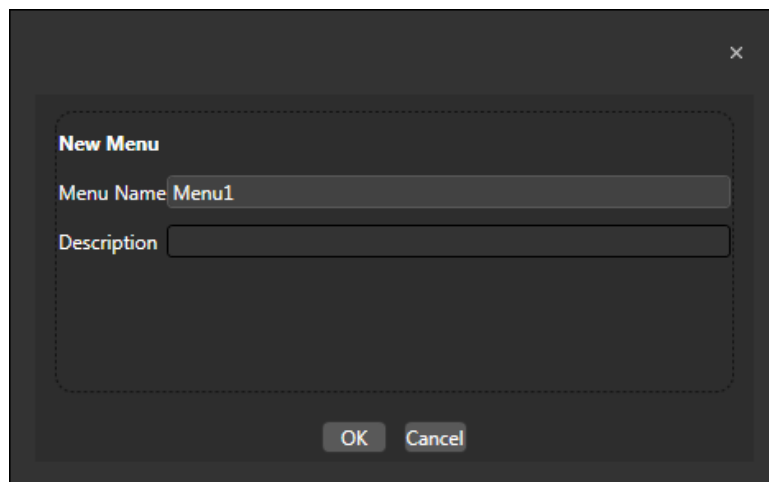
## 15.1. Project Menus

Popup menus that activate when clicked on can be created in Movicon and organized in tree structures with sub-menus. Custom menu bars can also be created and associated to screen pages. These menus can be used for activating the same commands that can be activated with buttons. These commands are available from the Movicon "Command List".

### Inserting a new Menu

In order to create a new Menu to appear in the "Menus" node within the Project Explorer window's tree structure, you need to use the Menus Resource node's contextual command icon or the "Menus" command from the "New Resources - New Resources" Ribbon. In order to rename a Menu, you will need to use the desired Menu Resource's command icon.

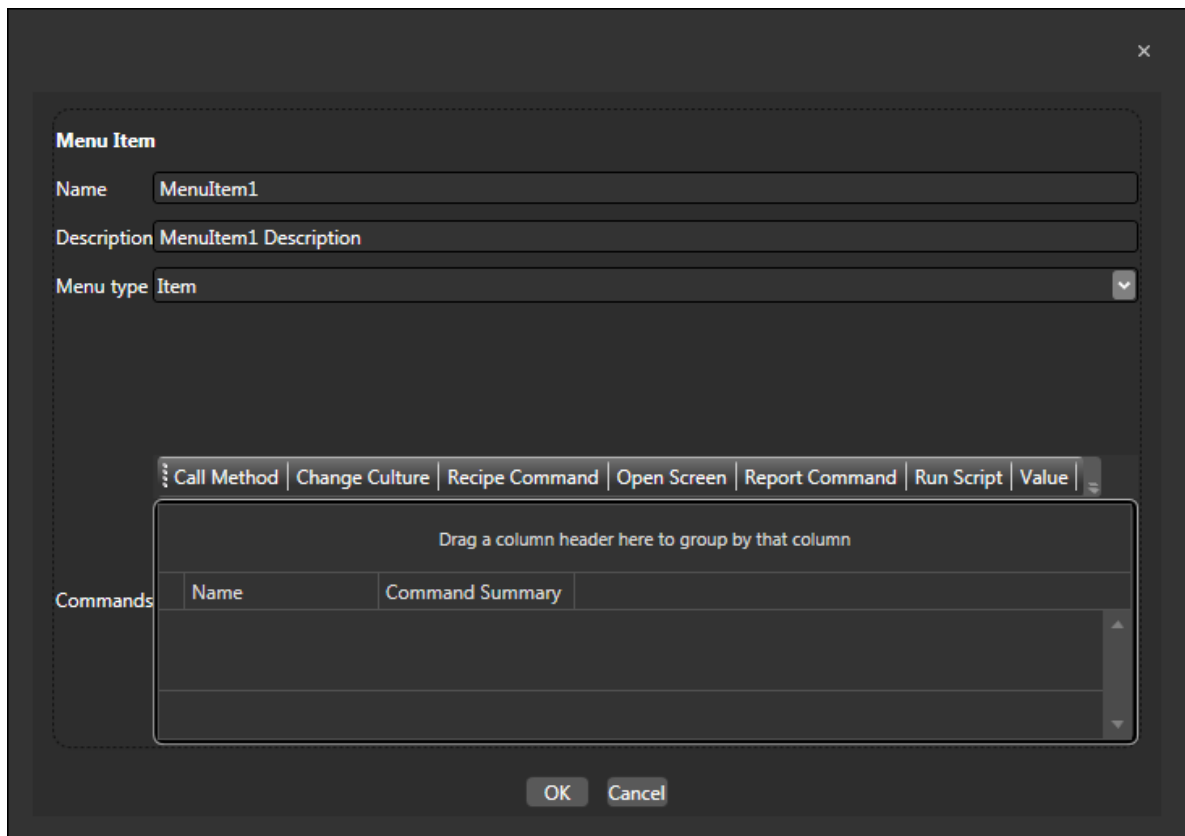
When a new Menu is added, it will automatically open in the workspace. To open an already existing Menu simply double click on the resource in the Project Explorer window or use the desired Menu resource's contextual command icon. When double clicking on the Menu's name within the workspace, a dialog window will open enabling you to enter a description for the Menu. This description will not be shown in runtime.



*Window used for entering a menu description*

### Inserting a Menu item

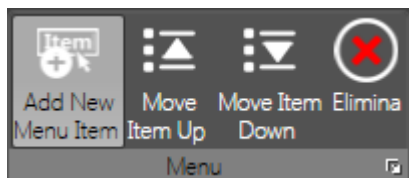
Each individual project Menu can be composed of one or more Items as menu components. These Items are displayed in the Menu's tree structure in the workspace and can be inserted and configured to compose menu and eventually structured into sub-menus. In order to add new Items, you need to use the right mouse button or the "Add New Menu Item" command from the "Menu Manager - Menu" Ribbon. When adding a new item, a dialog window will automatically open to define its settings. This same window can also be opened by double clicking on an already existing item to edit it.



Window used for setting the Menu Item properties

## Commands for editing Menu Items

The commands for editing menu items are available from the "Menu Manager - Menu" Ribbon.



Ribbon containing the menu edit commands.

### Add New Menu Item

This command is used to insert a new Item in the Menu Resource being edited. Once the new item has been created a pop-up window will display containing the menu item's properties (described in previous paragraph).

### Move Item Up

This command is used to move the selected menu item one position up in the Item list of the menu being edited.

### Move Item Down

This command is used to move the selected menu item one position down in the Item list of the menu being edited.

### Test Menu

This is used to test how the menu being edited will display by showing an preview of the results obtained.

## 15.2. Menu Properties

The Menu Item properties are:

The screenshot shows a 'Menu Item' configuration window. It includes input fields for Name, Description, Menu type (a dropdown), Enable Tag (a dropdown), and Mark Tag (a dropdown). Below these is a 'Commands' section with a tabbed interface. The tabs are 'Call Method', 'Change Culture', 'Recipe Command', 'Open Screen', 'Report Command', 'Run Script', and 'Value'. The 'Call Method' tab is selected. Below the tabs is a table with two columns: 'Name' and 'Command Summary'. Above the table is a text prompt: 'Drag a column header here to group by that column'. At the bottom of the dialog are 'OK' and 'Cancel' buttons.

### Name

Represents the name associated to the menu Item being edited. The menu Items support the Change Language function, therefore item names can be represented by fixed text or an "ID\_String" which must be declared in the string table. In this case the item name will change in runtime according to the language activated.

### Description

Represents a description associated to the menu item by the design engineer. This description will not be shown in runtime.

### Menu Type

Represents the item type to be inserted in the menu. This property can have the following values:

- **Item:** indicates that one or more commands can be associated to the item and can contain sub-items
- **Separator:** indicates that the item will be a simple separator between the menu items. It will be represented as a line.
- 

### Enable Tag

Allows you to select the variable that triggers the activation of "Menu Item". If the variable is reset to "zero", the menu item will appear inactive and therefore unavailable. Otherwise If the variable has a value different than "zero" the menu item will appear active and therefore available for executing the command.

If you do not specify this option by leaving it blank, Movicon will consider the item as enabled.

**Mark Tag**

This is used to select the name of the variable that will determine the display of the check mark (✓) next to the menu Item. If the variable is set to a logic state different from "zero", the check mark appear will appear next to the item. If you do not specify this option, the check mark will not display.

**Image**

Allows you to select an image to associate to the Item menu. The image default size is 32x32 px.

**Commands**

This section is used for defining a list of one to several commands that will be executed when the menu item is activated. The editable Command List is the same one available for the Button objects.

## 15.3. Menu Activation in Runtime

The Menu resources can be activated only in screen menu bar format or as contextual pop-up menus associated to objects or symbols on screen in runtime.

**Screen Menu Bar**

To activate a Menu as a customized Screen Menu Bar simply define the Menu with the same of the Screen you wish to activate. In this way the menu bar will appear at the stop of the Screen when it opens.



If the Screen has been inserted inside a folder (or in a nested folder), the Menu must be inserted inside a folder with the same name.



Defining a Menu the "Main" name, Movicon will activate for default this menu as a menu bar for all the Screens in the project that have not yet been defined with their own menu being a menu with the same name.

**Contextual Menus for symbols and objects**

To activate a Menu as a contextual pop-up of an object or symbol, object's "MenuName" property will need to be set with the name of the menu to be displayed. In order to view this menu in Runtime simply right click on the object.

# 16. Shortcut

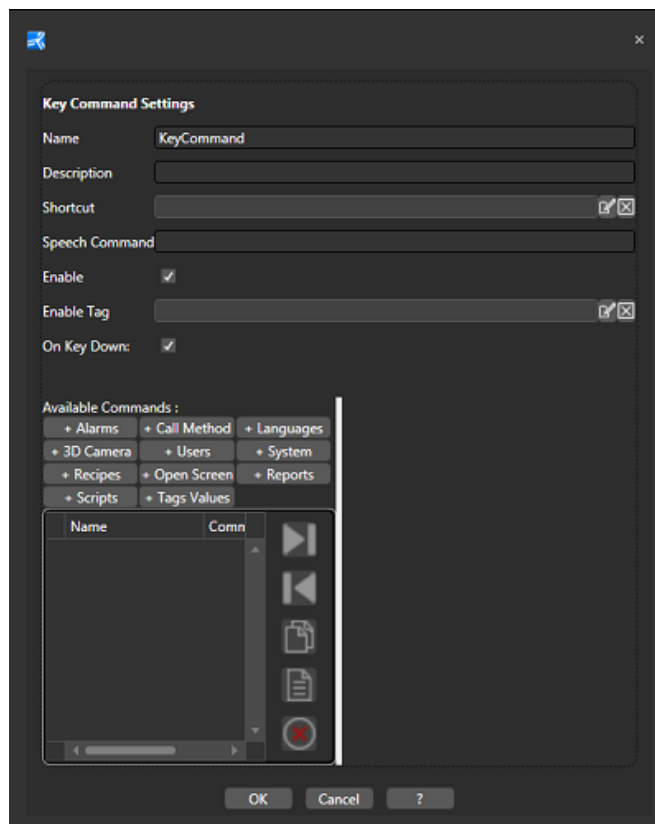
## 16.1. Project Shortcuts

When using Movicon you can execute a series of commands simply by using the keyboard. This is done by associating the command to keys or combination keys as shortcuts to commands in function with the screen currently active at that moment. These shortcuts can be used for activating the identical command that can also be activated with those buttons available in the Movicon "Command List". "Commands List"

### Inserting new shortcuts

To create a new shortcut, which will then appear as a sub-note of the "Shortcuts" node in the Project Explorer window tree structure, you can use the contextual Shortcuts resource node's command icon, or use the "Shortcuts" command from the "New Resources - New Resources" Ribbon. To rename a Shortcut use the rename command icon at the side of the Shortcut Resource desired.

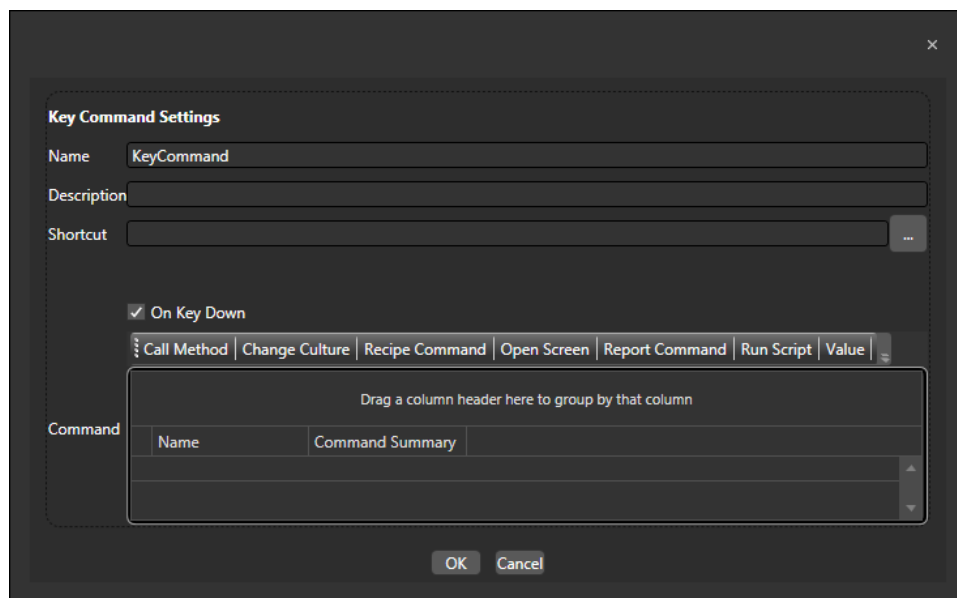
When a new shortcut is added it will automatically open in the workspace. To open an already existing shortcut simply double click on the resource within the Project Explorer window or use the desired shortcut resource command icon. The new shortcut can be given a description using the dialog window which opens by double clicking on the shortcut's name within the workspace. This description will not show in runtime.



Window used for inserting a description for a shortcut.

## Inserting shortcut command keys

Each of the project's shortcuts can be composed of one or more Key Commands. The Key Commands are displayed in the tree structure of the Shortcut displayed in the workspace where it is possible to insert the keys and configure them using the appropriate commands. To add new Key Commands use the right mouse key or the "Add New Key Command" command from the "Menu Manager - Menu" Ribbon. When adding a new Key Command a dialog window will automatically open displaying its settings. This window can also be opened with a double click on an existing key to modify its settings.



*A Shortcut Key Command property setting window*

## 16.2. Shortcut Properties

The properties of a Shortcut key command are as follows:

### **Name**

Represents the name associated to the Key Command. A shortcut can actually contain a list of several Key Commands.

### **Description**

Represents an eventual description that can be associated to the Shortcut key command by the programmer. This description will not be shown in runtime.

### **Shortcut**

Represents the combination keys (or key) requested for activating the command, or command list. This property is used for specifying the key to be used by clicking in the text box and using the keyboard to enter it manually. As an alternative you can also use the "..." button to the right that once clicked will open a pop-up window where the desired combo keys to be associated to the shortcut can be entered.

### **Speech Command**

specifies the "Voice Command", to be used at runtime, to execute the accelerator.

**Enabled**

Enable / disable the accelerator in development.

**Enable TAG**

Allows you to select the variable that triggers the activation of "Menu Item". If the variable is reset to "zero", the menu item will appear inactive and therefore unavailable. Otherwise If the variable has a value different than "zero" the menu item will appear active and therefore available to the execution of the command.

If you do not specify this option, leaving the option empty, Movicon will consider the item always enabled.

**On Key Down**

This property indicates whether the command associated to the created combo keys is to be executed when the key is pressed down (key Down) or released (Key Up).

## 16.3. Shortcut Activation in Runtime

The shortcut resources can be activated directly from the screen containing them in runtime. To activate the shortcuts of a specific Screen simply enter the Shortcut with the same name of the screen you wish to activate. In this way when the Screen opens, the corresponding Shortcut will activate to allow command execution upon pressing its corresponding keyboard keys.



If the Screen has been inserted in a folder (or in a nested folder), the Shortcut must be inserted in the folder with the same name.



When defining a Shortcut with the name of "Main", Movicon will activate this Shortcut for default for all the project Screens that have not been defined with their own Shortcut, that is one with the same screen name.





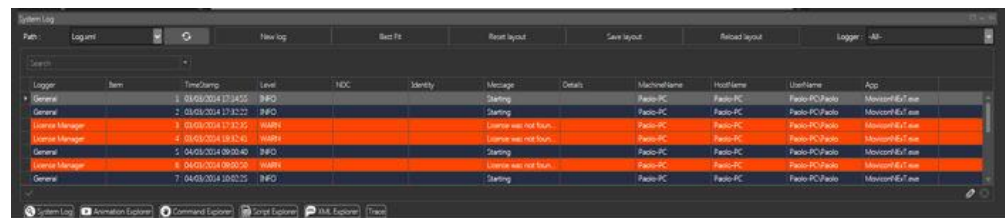
# 17. Diagnostic and Log

## 17.1. System Log Window

The **System Log** window that reports information and error messages generated during project editing or runtime is located on the bottom border of the Movicon.NexT workspace.

This window is accessible during design engineering mode by using the Tabs located on the bottom border. It can also be opened during project Runtime with the "**SHIFT+F2**" keys.

In addition the "system log viewer" object is also available from the toolbox for inserting on screen.



The data shown in the "System log" window is saved on file. This file is created in:

C:\ProgramData\Progea\Movicon.NEX.T.Log\Log.xml.

The information saved in the Log file is displayed as follows:

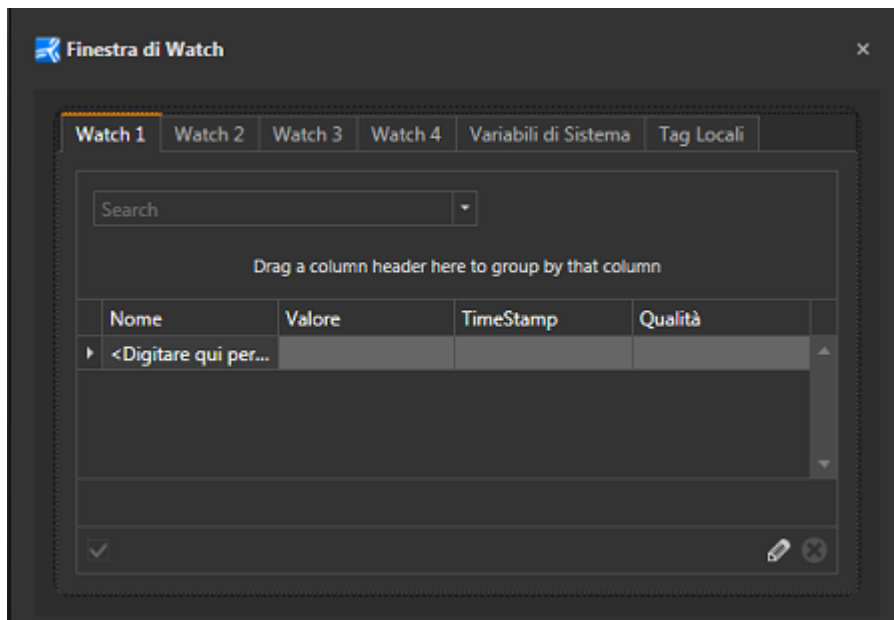
<b>Logger</b>	Module from which message derives, for example Server if from Server side or General if a project message.
<b>Item</b>	Generated message is shown by incremental ordinal number of occurrence.
<b>TimeStamp</b>	Date and time when message was generated.
<b>Level</b>	Indicates the type of message, INFO if it is a simple informational message, ERROR if it is an error
<b>NDC</b>	Work in progress
<b>Identity</b>	identified the user who executed the process that generated the error.
<b>Message</b>	Error message text.
<b>MachineName</b>	Name of PC in which message was collected.
<b>HostName</b>	Name of PC in which message was generated.

<b>UserName</b>	Name of Windows user which Movicon started up with.
<b>App</b>	Application which generated error (MoviconNExT.exe, MovNExTServer.exe...).

In addition to messages the System Log window also has some fields that can be used for filtering events for the "Logger" column or for searching texts.

## 17.2. Watch Window(Tags)

The Watch Window can only be activated within the project in Runtime mode using the "SHIFT+F5" command if the project is previously started up from development mode. If the project is started up directly in Runtime mode, it will not be possible to open the Watch Window.



The Watch Window is used to monitor and modify project variables whether they be server variables or local client variables. The following tabs will be shown in the window:

- Four Watch tabs (Watch 1, Watch 2 and so forth) from which it will be possible to select the Server variables that have been defined in the project.
- A SystemVariables tab containing a list of the system variables.
- A LocalVariables tab containing a list of the Local Client variables.

When the Watch Window is closed, the settings performed in the Watch tab will be saved in the Isolated Storage of the Windows user. In this way it will be possible to keep the list of tags inserted in the Watch tabs for when the window is opened next time.

### Limitations

Method variables are not handled in the Watch Window.

## 17.3. Server (OPC UA Browser) Diagnostics Window

When starting up the Server side from the development environment manually, using the "Start Server" command, a Server OPC UA Browser window will automatically open showing various information. The information shown in this window includes Server diagnostics, list of set alarms, list of Tags that the Server has published. In addition, by using the window's 'Live Data' section you will be able to view real Time Tag values that can be modified if Tag is writable.

The Server's diagnostics window is in fact an "OPC UA Browser" window that can also be used to connect to an OPC UA Server to retrieve information.

## 17.4. Client Status (OPC UA Client Status) Window

The "OPC UA Client Status" window shows information inherent to the project's Client side. This window can be opened during Runtime mode using the "CTRL+SHIFT+1" combination keys. This window shows which variables are in use in the project and allows you to monitor messages from the Server.

## 17.5. Graphical Object Diagnostics

A graphical object shows the user information relating to the associated variable. If there is a problem with the variable's quality, the object is highlighted with a red border. In order to find out the reason for this, right click on the object's border to access the 'Connection Status' menu.

This menu shows the following information:

- **"Monitored Item"**: complete name of variable
- **"id"**: name of variable
- **"Value"**: variable value, if invalid, will be set = null **"Quality"**: communication quality can be Good, Uncertain, BadTimeout, BadOutOfRange...For the complete list please refer to the documentation relating to the OPC UA specifications.
- **"Timestamp"**: date of value update
- **"Relative Path"**: relative path
- **"Last message"**: last message sent by server connection.
- **"Session"**: connection session towards server.



# 18. Generic Information

## 18.1. Tips and Tricks

The following commands and procedures can be used while editing projects.

### **How to delete variable associations to objects**

Go to object's command menu and click the button used for opening the selection window while keeping the ALT key pressed down.

### **How to display or hide the icon menu on the object's border**

The object's icon menu can be displayed or hidden by using the little triangle button located on the actual menu's border. In addition it can be set as open or closed for default using the task ribbon "Screens - Layout - Show-Hide Adorner Expander" .

### **How to change the object's rotation barycenter**

The object's rotation refers to its barycenter which can be changed by activating the small round red button located at the center for default by using the appropriate item from the object's Command Menu.

### **How to select an elements of a group of elements belonging to a 3D model**

First the "Enable 3D Edit" command must be enabled from the object's command menu. After which click on element to select it or use SHIFT+Click to select the group of elements (predefined in the model). The selected elements will highlight.

### **How to activate a Camera in runtime**

During edit mode camera points can be saved with custom names using the "3D Camera Position" command from the object's command menu. As a consequence, it will be possible to activate the memorized camera displays by right clicking on the object: a menu will appear showing a list of the camera points saved.

### **How to move the Camera of a 3D object in runtime**

Click and drag the object using the left mouse key.

### **How to use the Zoom of a 3D object in runtime**

Click and drag the object using the right mouse key.

### **How to put prototype structure members into a particular order**

Use the "Move Member Up" and "Move Member Down" commands from the Ribbon.

### **How to add/remove points to/from a Polyline**

SHIFT+Click to add or remove an intersection point.

**How to delete an image associated to an object or screen**

The image is considered to be part of the object's brush feature and therefore is simply deleted by re-assigning the object with another back color.

**How to Zoom a screen in NON Touch-Screen systems**

Keep the CTRL key pressed down and use the mouse wheel to zoom in and out

**How to select a grouped symbol components quickly**

Use the select components command from the symbol's command menu: a menu will display showing the structure of the grouped symbol's components. Simply select the desired component from this menu. The selected component will then show highlighted. The component can also be selected by clicking on it while keeping the "CTRL + SHIFT" keys pressed down.

**How to assign multiple variable selections to Screen objects**

After having opened a screen select the desired object from the ToolBox, eg. a display of gauge object, then select the variables to be used from Tag list located in the project's tree structure (use Ctrl+Click for multi-selecting) and then drag the variables onto the screen. At this point Movicon will generate the same number of objects of the type selected and then open a window for distributing the objects within the space provided. At this point each object inserted will be associated to a variable from those selected.

**How to find out if a variable is associated to an object in runtime**

While in runtime right click the object to display a menu through which a window can be opened containing information about the object including which tag has been associated, its value, quality and timestamp.

**How to open the Common Property Window**

The object's Common Property Window can be opened using the relevant command from the menu that is activated on the object's adorer or from the menu that appears when right clicking on the object.

**How to open the Tag Browser Window**

To open the Tag Browser window for selecting a variable to associate to the object simply double click the object while pressing down the SHIFT key, or use the command provided for this purpose in the object's command menu.

**How to open the System Log window during runtime mode**

Use the SHIFT+F2 command during runtime

**How to open the Client Status window during runtime**

Use the SHIFT+F1 command during runtime

**How to open  
the Cross  
Reference  
window during  
runtime**

Use the SHIFT+F3 command during runtime

**How to open  
the Watch  
window during  
runtime**

Use the SHIFT+F5 command during runtime

**How to reduce  
an object  
proportionately**

Keep the SHIFT key pressed down and drag the border of the object to resize it. The object will be resized proportionately in height and width.

## 18.2. How to create WPF objects in Platform.NExT

This example shows you how to create a WPF UserControl object within the Visual Studio 2012/2013 development environment and how to use this object in Platform Next.

This example includes the use of these folders:

- MyWpfUserControl1: contains the Visual Studio solution used for creating the User Control
- ProgramData\_ToolBox: contains the file for adding the User Control to the Platform Next Toolbox
- PlatformNext\_Project1: contains an example of a project that uses the User Control.

### How to create the User Control

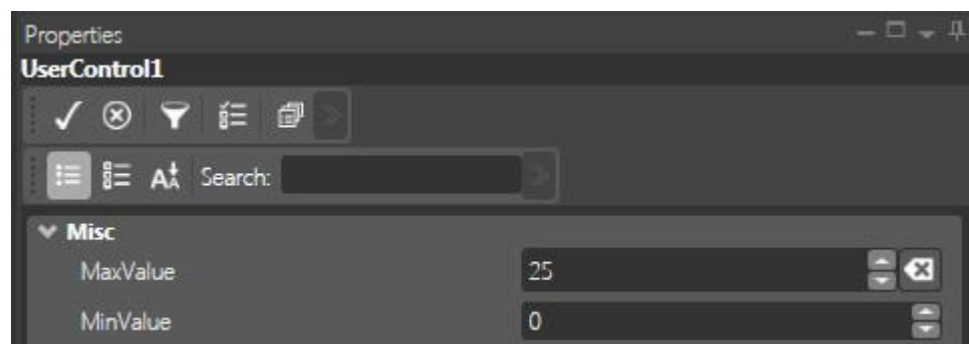
Use Visual Studio 2012 or an equivalent to create a new "WPF User Control Library" project in which to insert the desired controls and functionalities.

To expose the properties to use in Platform Next or to combine a project tag you will need to define the "DependencyProperty" in the user control.

The "DoubleValue", "MaxValue" and " MinValue" have been declared in the example solution.

The "DoubleValue" will be bound to the Platform Next project's variable as described in point 2.

The "MaxValue" and " MinValue" are static properties and will be exposed in the "instantiated" object property list on screen.



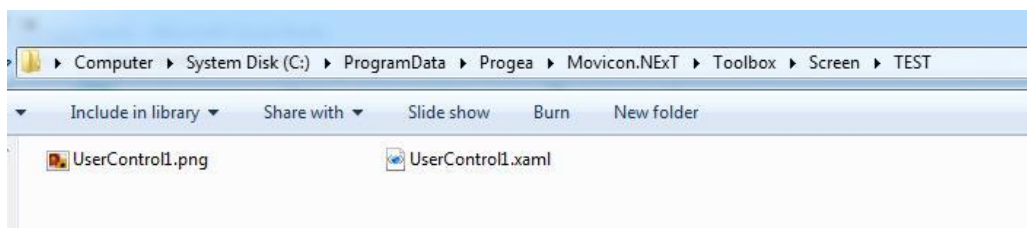
The file that is generated by the user control compilation, which is **"MyWpfUserControl1.dll"** in this example, must be copied to the "Toolbox" sub folder of the Platform Next installation path "C:\Program Files\Progea\Automation Platform.NExT\Toolbox".

### How to insert the User Control in the development system's Toolbox

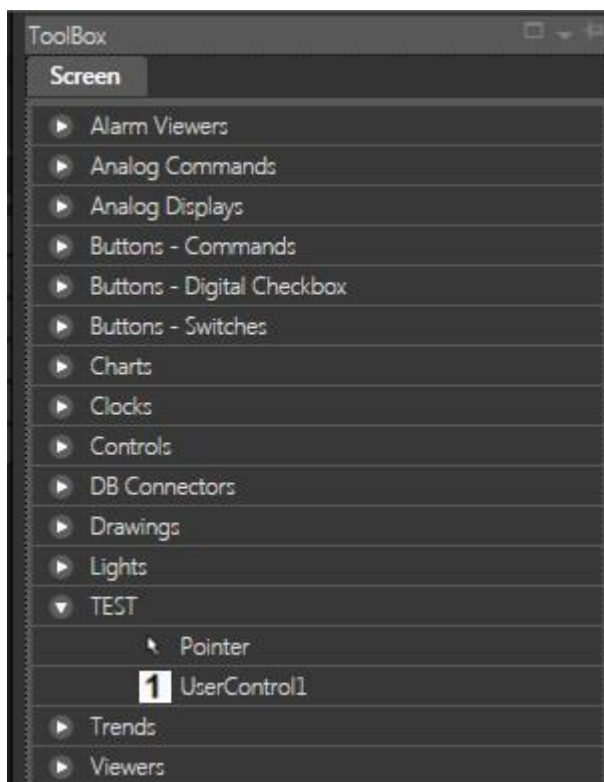
In order for Platform Next to handle the control that has just been created and inserted in the "ToolBox" subfolder, you will need to define a XAML file like the one in the "ProgramData\_ToolBox" folder provided with the example.

The XAML file is also combined with a PNG file that will contain the icon used in the Platform Next toolbox. The XAML and PNG files must be copied to a folder named as pleased in "C:\ProgramData\Progea\Movicon.NExT\Toolbox\Screen". The name of the folder created will be the name of the library in the toolbox.

For example, If we use "TEST" as the folder's name, the XAML and PNG files will be in "C:\ProgramData\Progea\Movicon.NExT\Toolbox\Screen\TEST"



This 'TEST' folder will also be displayed in the toolbox within the development environment as shown in the image below:



The XAML file definition, as shown in the code below,

```
<tr:UserControl1 xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
```



```

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
Tag="ProblematicXamlWriter"
xmlns:tr="clr-namespace:MyWpfUserControl1;assembly=MyWpfUserControl1"
Width="250" Height="60" Background="{x:Null}"
xmlns:mynsp="clr-namespace:Converters;assembly=Converters">

    <tr:UserControl1.Resources>
        <!-- to force interoperability on webclient -->
        <Style TargetType="{x:Type ListView}" x:Key="ConnectionStringBind"
/>

        <!-- to use our stringtodouble converter -->
        <mynsp:UpDownValueConverter x:Key="UpDownValueConverter" />
    </tr:UserControl1.Resources>

    <tr:UserControl1.DoubleValue>
        <Binding Path="Value" Converter="{StaticResource
UpDownValueConverter}" Mode="TwoWay" ValidatesOnDataErrors="True"
UpdateSourceTrigger="PropertyChanged"></Binding>
    </tr:UserControl1.DoubleValue>
</tr:UserControl1>

```

must be kept but you will need to consider that:

```
xmlns:tr="clr-namespace:MyWpfUserControl1;assembly=MyWpfUserControl1"
```

reports the definition of the namespace and the assembly created in the User Control using Visual Studio;

```
Width="250" Height="60"
```

report the default width and height values of the User Control inserted in a screen;

```
xmlns:mynsp="clr-namespace:Converters;assembly=Converters"
```

consents the use of the Converter that has already been defined in Platform Next for transforming data of different types. This example indicates the use of a bidirectional double data string converter

```
(<mynsp:UpDownValueConverter x:Key="UpDownValueConverter" />);
```

```
<Style TargetType="{x:Type ListView}" x:Key="ConnectionStringBind" />
```

necessary if the User Control has to be supported by webclient HTML5 as well

```

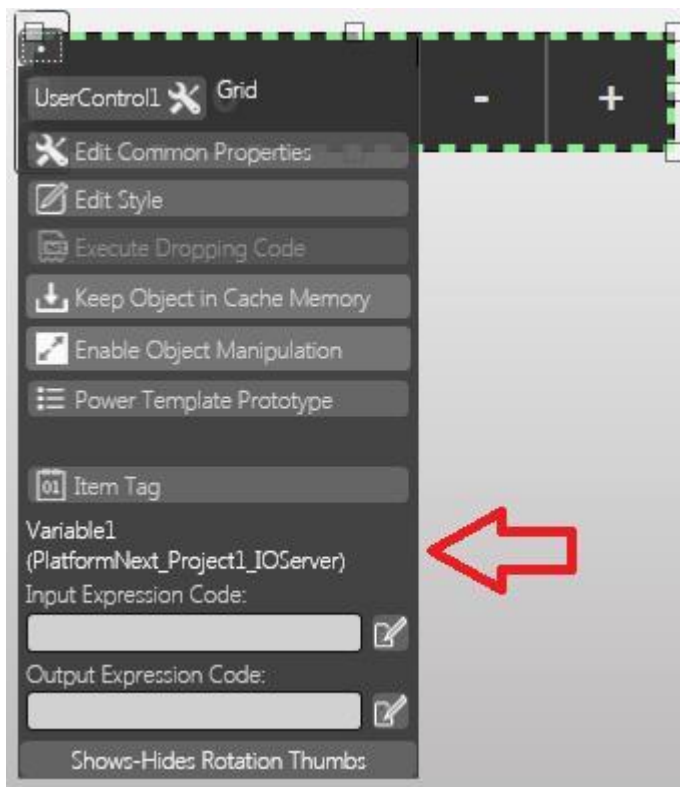
<tr:UserControl1.DoubleValue>
<Binding Path="Value" Converter="{StaticResource UpDownValueConverter}"
Mode="TwoWay" ValidatesOnDataErrors="True"
UpdateSourceTrigger="PropertyChanged"></Binding>
</tr:UserControl1.DoubleValue>

```

Defines the binding between the "DoubleValue" property exposed by the User Control and the "Value" value exposed by Platform Next containing the variable value combined with the object, as shown below.



Note that the "Value" value is always handled as string.



### Using the User Control in a project

Once the assembly file has been copied to "C:\Program Files\Progea\Automation Platform.NExT\Toolbox" and the XAML and PNG files have been inserted in the folder in "C:\ProgramData\Progea\Movicon.NExT\Toolbox\Screen", it will be possible to insert the User Control in a screen by simply dragging it from the toolbox.

The "PlatformNext\_Project1" folder provided with this example contains a Platform Next project that uses the User Control as described in point 1. The user control along with a standard display have been inserted in "Screen1" and have both been linked to the same variable to allow the behavior of the variable's value to be monitored.





