



# Movicon NExT

## 22.0 Tutorial

Ver.3.4.268



# Table of Contents

<b>1. GETTING STARTED WITH AUTOMATION PLATFORM.NEXT .....</b>	<b>1</b>
1.1. MOVICON.NEXT TUTORIAL INTRODUCTION .....	1
1.2. THE MOVICON.NEXT MODULES .....	1
1.3. ARCHITECTURE .....	2
<b>2. AUTOMATION PLATFORM.NEXT PROJECTS.....</b>	<b>5</b>
2.1. STARTING UP MOVICON.NEXT FOR THE FIRST TIME .....	5
2.2. CREATING A NEW PROJECT .....	6
<b>3. I/O DATA SERVER.....</b>	<b>13</b>
3.1. HOW TO CREATE TAGS.....	13
3.1.1. <i>Insert a new tag</i> .....	13
3.1.2. <i>Tag Properties</i> .....	14
3.2. ADDING COMMUNICATION DRIVERS .....	16
3.2.1. <i>Configuring the Driver</i> .....	17
3.3. TAG I/O PHYSICAL ADDRESSES.....	18
3.4. IMPORTING TAGS.....	20
<b>4. SYNOPTIC EDITING .....</b>	<b>23</b>
4.1. CREATE A SCREEN .....	23
4.2. INSERTING OBJECTS IN SCREENS .....	26
4.2.1. <i>Graphics Editor</i> .....	27
4.2.2. <i>Importing Drawings</i> .....	27
4.3. TOOLBOX.....	27
4.3.1. <i>Inserting objects from the Toolbox</i> .....	28
4.4. SYMBOLS LIBRARY .....	28
4.4.1. <i>Inserting symbols from the Symbol Gallery</i> .....	30
4.5. SCREEN NAVIGATION .....	30
4.6. EDITING GRAPHICAL OBJECTS .....	31
4.6.1. <i>Editing Objects</i> .....	32
4.7. ASSIGNING TAGS TO OBJECTS.....	33
4.7.1. <i>The Reference Tag</i> .....	33
4.8. TAG BIT AND ARRAY TAG POINTING .....	37
4.9. CREATING COMPOSED SYMBOLS .....	38
<b>5. DYNAMIC ANIMATION OBJECTS .....</b>	<b>41</b>
5.1. DYNAMIC ANIMATION SETTINGS .....	41
5.2. HOW TO GIVE OBJECTS COLOR ANIMATION .....	44
5.3. MORE DYNAMIC ANIMATION EXAMPLES .....	46
5.4. 3D ANIMATION GRAPHICS.....	50
5.5. USING MULTITOUCH .....	56
5.5.1. <i>Manipulating objects</i> .....	57
5.6. USING DISPLAY CONTROLS .....	59
5.7. USING COMMAND OBJECTS.....	60
5.8. STARTING UP PROJECTS IN RUNTIME.....	65
<b>6. ALARM MANAGEMENT.....</b>	<b>69</b>
6.1. INSERTING ALARMS .....	69
6.2. DISPLAYING ALARMS.....	74
6.3. CREATING ALARM SIMULATIONS .....	76
6.4. EVENT HISTORIAN LOG .....	78

6.5. HISTORIAN LOG VISUALIZATION.....	79
---------------------------------------	----

<b>7. HISTORIAN MANAGEMENT .....</b>	<b>81</b>
--------------------------------------	-----------

7.1. HISTORIANS MANAGEMENT .....	81
7.1.1. <i>Historians</i> .....	81
7.1.2. <i>Data Loggers</i> .....	81
7.2. INSERTING AND ASSOCIATING HISTORIAN PROTOTYPES .....	82
7.3. DISPLAYING HISTORIAN WITH DATA ANALYSIS .....	86

# 1. Getting Started with Automation Platform.NExT

## 1.1. Movicon.NExT Tutorial Introduction

Welcome to the Movicon.NExT Tutorial .

This tutorial aims to give a quick illustrative guide on the main Movicon.NExT automation platform techniques used in its main modules which include the Data Server for communication and the Movicon.NExT graphical interface.



© 2015 Progea srl - All Rights Reserved



By using the knowledge acquired by the end of this tutorial you should be able to configure the main supervision project resources with Movicon.NExT.  
Before reading this guide you should install the software using the setup procedure.  
All the information in this document is based on the assumption that:

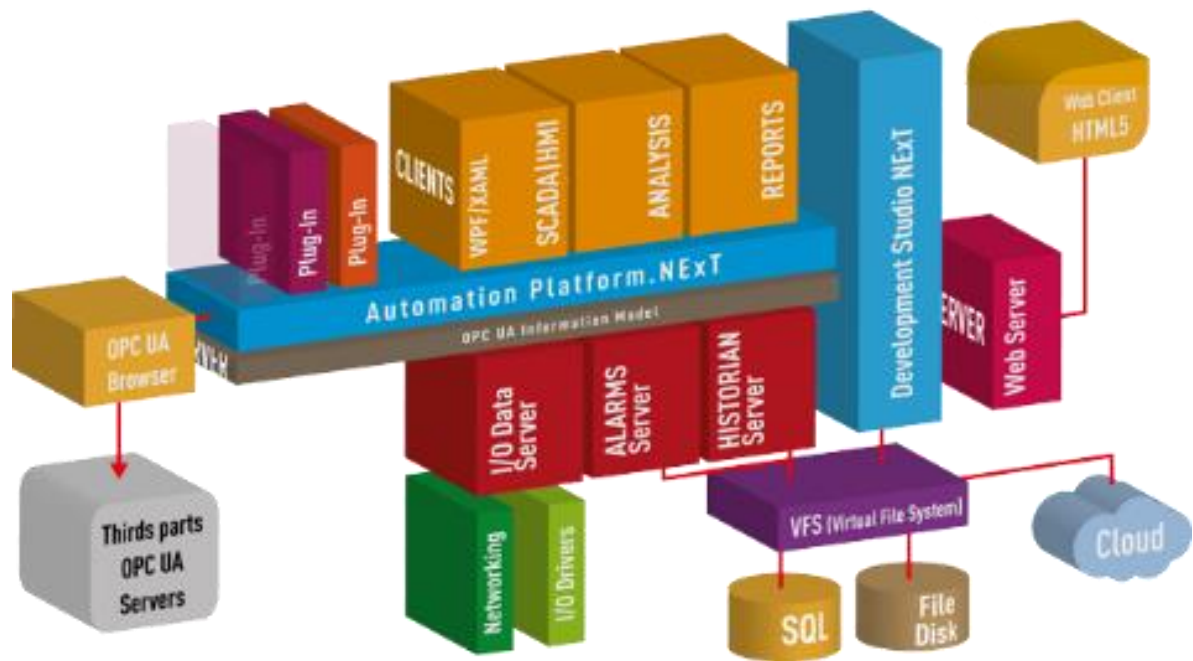
1. The operating system being used is Windows 8 64 bit or later versions
2. The user knows how to use Windows
3. The user has sufficient knowledge of automation systems, variable and PLC concepts

## 1.2. The Movicon.NExT Modules

Progea's Movicon.NExT is the most modern software technology available on the automation market. Its consists of an industrial framework developed by Progea that provides a suite of functional modules with ""plug-in" technology to give your application modularity and flexibility.

These platform's modules are available when installed with the setup procedures. If they are not installed, the platform will display only available modules within its context.

This technology enables platform expandability to integration of future function modules developed by Progea and third party modules as well.



*This diagram shows the platform with its main functional modules and their collocation within the system's architecture.*



Movicon.NExT is a modular and scalable platform. The modules can be made available in the platform if installed using the normal installation procedures. This tutorial presumes that all these function modules have been installed as necessary for supervision applications. These modules are:

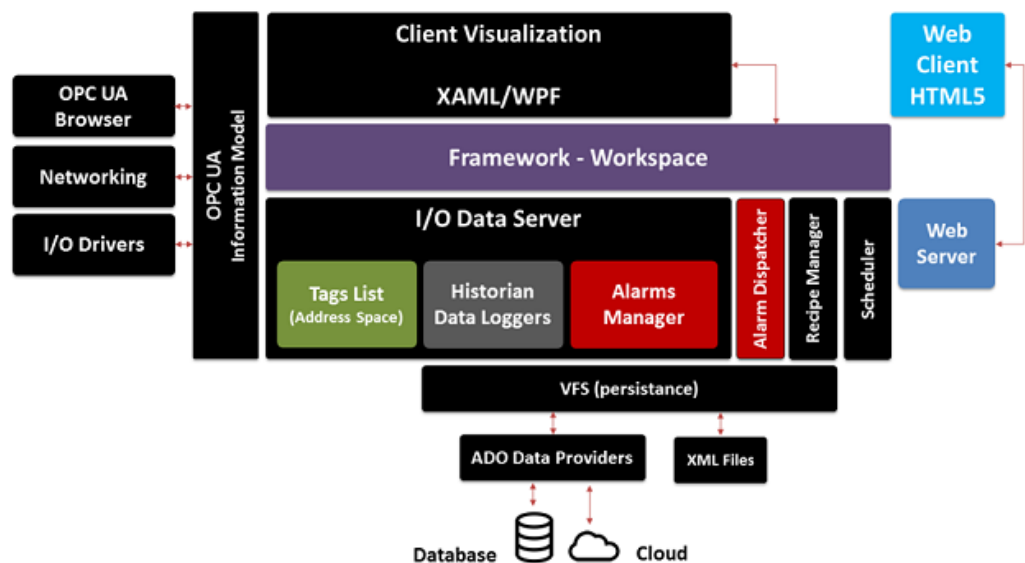
- **Data Server**, to communicate with the field and Tag definitions
- **Movicon.NExT**, for graphical interface
- **Alarm Manager**, for managing Alarms and Event Historians
- **Historian**, for recording data.

## 1.3. Architecture

Movicon.NExT uses a innovative and scalable architecture concept based on the most modern software technologies around today.

The architecture uses a Client - Server model based on .NET framework.  
The MoviconNExT technology added value can be seen in the following areas:

- Configuration ambient completely based on WPF
- Data Server modular with OPC UA information-based model
- HMI graphical interface based totally on WPF with indispensable XAML graphics quality
- .NET technology with openness to third party integration
- Built-in Web Server with HTML5 technology
- Powerful and performing Historian based on SQL Server and open to DB Providers
- Alarm Management and Downtime Analysis
- Great variety of built-in visualization, analysis and control tools.



*This diagram illustrates the fundamental concepts of the Platform.NExT Client-Server architecture*



The Movicon.NExT platform is a new concept designed to provide a new built-in solution experience in the world of evolutionary industrial automation and Enterprise Control Systems.

Progea is working on revolutionary concepts to enrich their product functionality with continuous implementation of new functions, models and graphical objects. Keep in touch with Progea at [www.progea.com](http://www.progea.com) !





## 2. Automation Platform.NExT Projects

### 2.1. Starting up Movicon.NExT for the first time

Movicon.NExT will open in Edit Mode when started up for the first time by displaying the "Startup Page" if not specified otherwise in the command line.



- **Edit Mode:** this is the system's operating mode that enables projects to be created, edited or modified mode.
- **Runtime mode:** this mode is used for running projects according to how they have been programmed.

#### The Startup Page

The Startup Page displays an interface screen in tile format to facilitate the procedure of setting the main commands used at system startup, such as:

1. Create a new project
2. Open existing project
3. Open a recent project
4. Use local or online documentation
5. Access to example projects, tutorial, on-line support

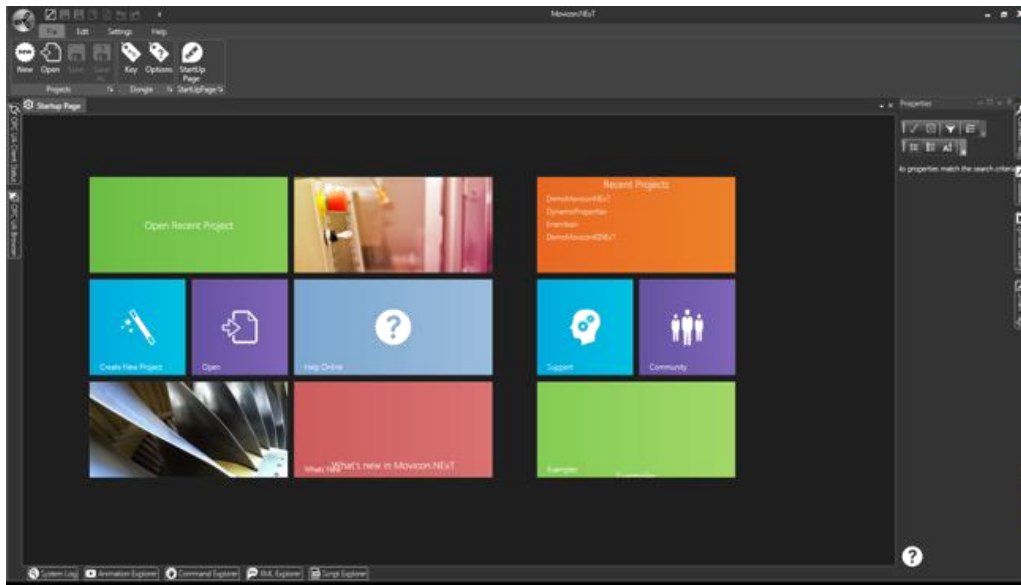
The workspace consists of a configuration window which we will go into more detail later. The user interface uses the 'Blend' style and positions all the mobile and floating windows in their original positions for default but is configurable as pleased there after.

The workspace contents can be configured manually at any time as pleased by the user. Modifications applied to the Workspace are persistent and will be automatically retained for the next startup.



**To restore all the default workspace settings, re-start**

Platform.NExT while keeping the 'CTRL' key pressed down. The workspace settings will return to their default values once restarted.



*The Movicon.NExT Startup Page with tiled interface for accessing the startup operations*

At this point the Startup Page Tile interface commands can be used for opening existing projects or creating new ones by using a wizard.

## 2.2. Creating a New Project

To create a new project simply use the **"New"** command available from the **"File"** menu, or use the appropriate icon from the **"Startup Page"** within the Workspace. The "New" button will launch the wizard to use as a guide to configure a new project.



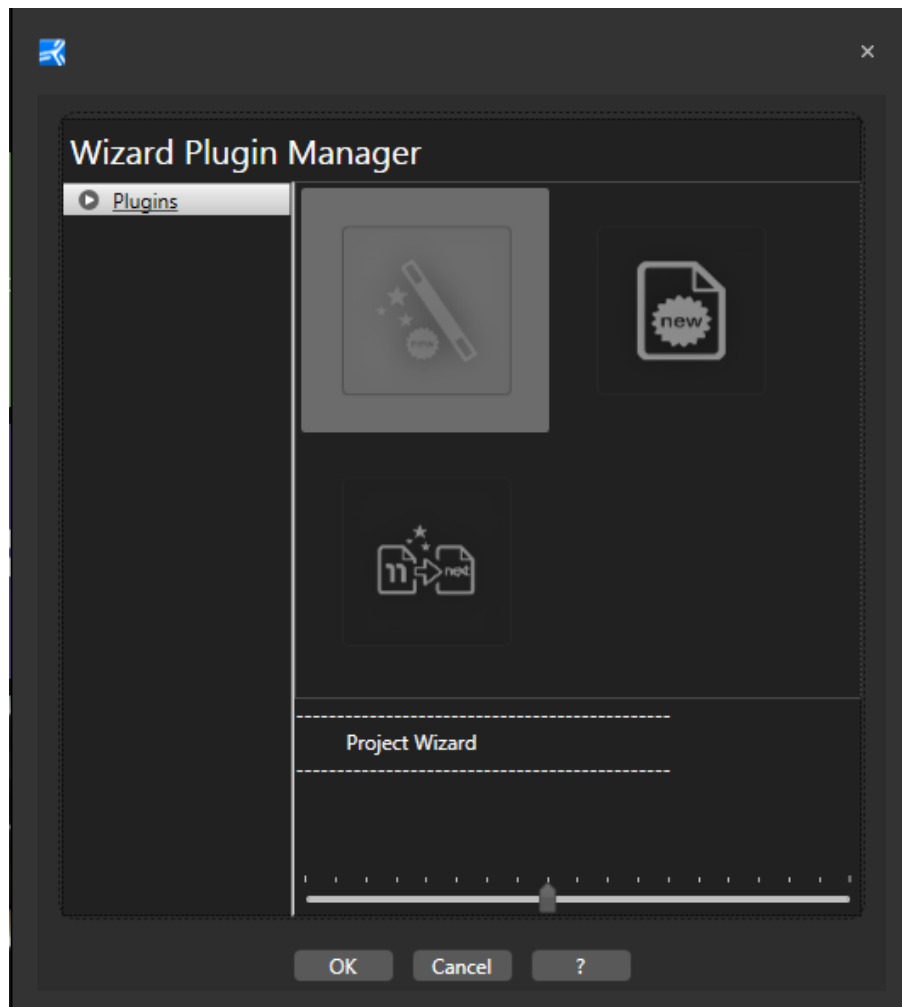
To open an existing project use the "Open" command and select the desired project file.

### The Wizard for creating New Projects

The "New" command uses a **wizard** through which a new project can be created and configured by following a simple and brief step-by-step guide. The wizard window follows 5 simple steps that will be described below:



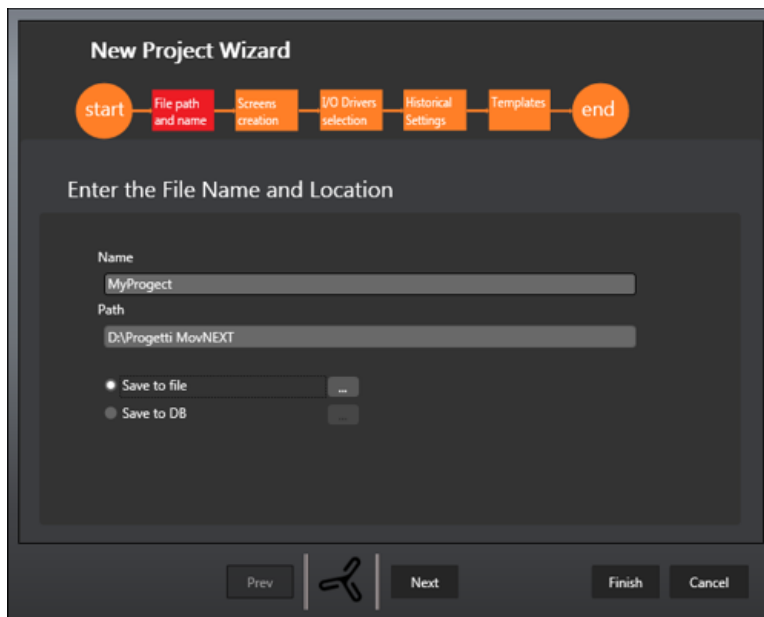
New 'empty' projects can be created or the wizard can be terminated at any moment to create a new project with those settings defined up to that moment.



*Window used for launching the Wizard or for creating a simple empty project.*

### **Step 1 : Project name and path**

The wizard's first window is used for setting the name and path of the project to be created. The project can be saved on file (in xml. format as proposed by default), or on database. When choosing the later you must select database type and which connection to use.



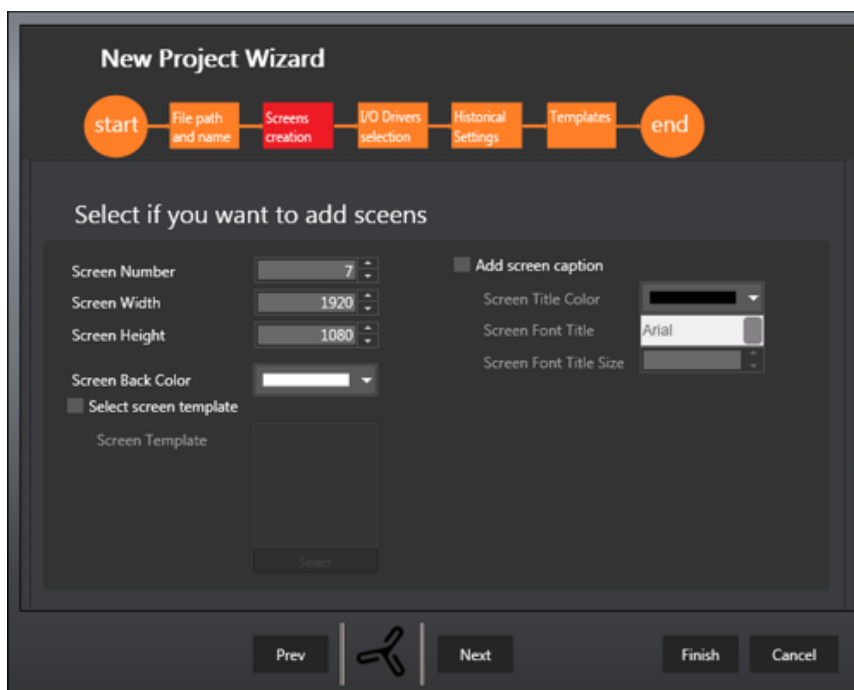
Window used for setting the project's name and path

## Step 2 : Creating Screens

The second wizard window asks you whether you wish to automatically create a certain number of screens in the project.

If you do not wish to create any screens, simply continue on to the next window by using the "next" button.

If you wish to create screens, you will need to set the required parameters as described below.



Window used for setting project screens

**Screen Number**

The number of screens to be added is entered in this field.

**Screen Width**

The Screen width in pixels is entered here for the screens to be added.

**Screen Height**

The Screen height in pixels is entered here for the screens to be added.

**Select Screen Template**

This check-box is used for enabling the selection of a Screen Template to be used for creating screens. A preview of the selected templates will be shown in the "screen Template" pane. The button underneath this preview window is used for opening the Template selection window.

**Screen Back Color**

If you do not select a template for the screens to be inserted, you can use this window to select the color to associate to the empty screen's background.

**Add Screen Caption**

When this checkbox is enabled, a Caption will be added at the top of every screen showing the screen name.

**Screen Title Color**

This field is used for selecting a color for the screen Title.

**Screen Font Title**

This field is used for selecting Font to be used in the Screen Title's text.

**Screen Font Title Size**

This field is used for selecting the font size to be used in the Screen Title's text.

**Step 3: Inserting Drivers**

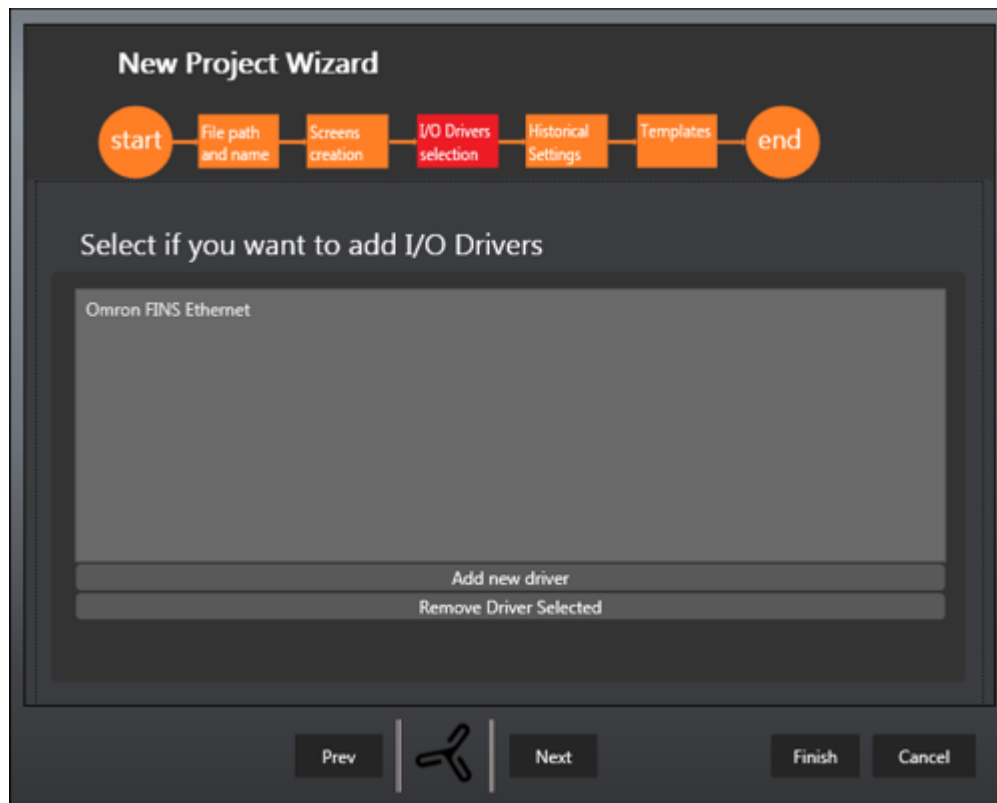
The third wizard window is used for adding one or more I/O Communication Drivers to the project.

If you do not wish to insert Drivers for the time being, you can always insert them during design mode. If this is the case press the "next" button to continue on to the next step.

If you wish to insert one or more I/O Drivers, use the "Add New Driver" to select the driver to install.



The Driver configuration settings are to be done later on in design mode after having created the new project.



*Window used for inserting I/O Drivers in the project*

You can insert all the drivers you need or remove them by using the appropriate command.

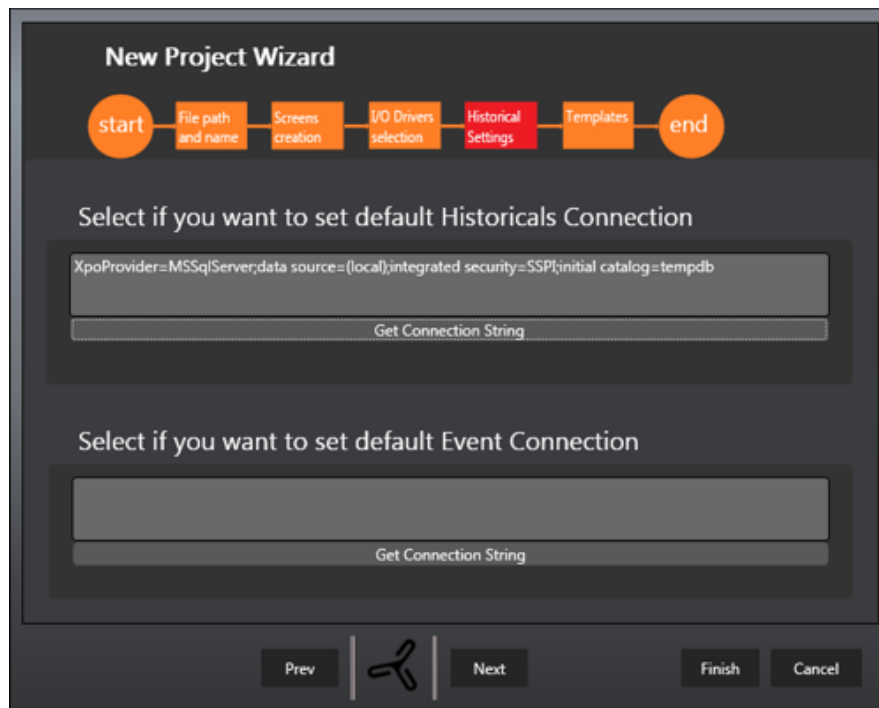
#### **Step 4: Configuring the DB connections for Historians**

The forth and last wizard window is used for setting the DB connections that will become the system's default settings to be used by the project's Historian engines (Event Log and Historian).

All the recording engines require a connection string definition in order for the project to connect to the relative databases for recording data.

If you do not intend to record data on the Historians you will not need to define a string. On the other hand you can also define a connection string for each individual project data recording configuration. In this case ignore this page and press the "next" button.

If you wish to provide access to databases with default connects for each recording engine you will need to apply the following configurations:



Window used for setting project Historian connections.

### Add Historians Connection

This field provides the command for inserting connection strings to the database for the Historian tables managed by the **Historian** module, whether being Data Loggers or Historian Prototypes.

A "**Get Connection String**" button has been provided to simplify connection procedures and through which a window will display for setting database type, table and authentication. The connection string will be generated automatically (see the chapter on Historian Management for more information on this argument)

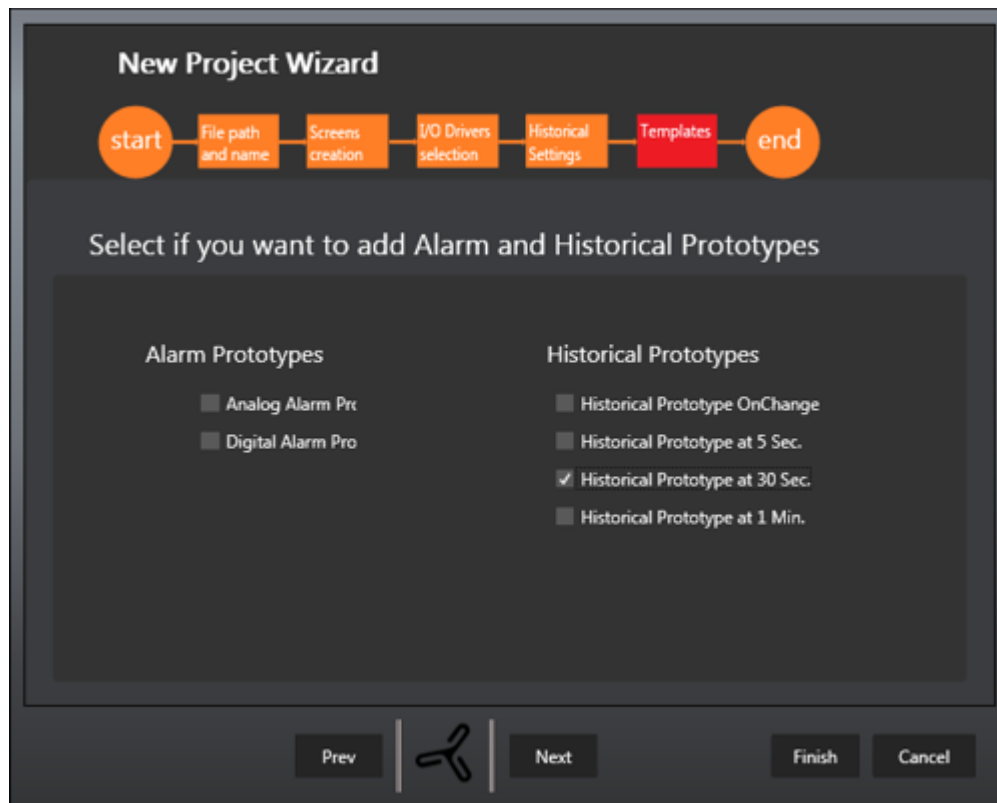
### Add Event Connection

This field provided the command for inserting the database connection string for data tables managed by the **Alarm Manager** module relating to the Alarm events and system messages.

A "Get Connection String" button has been provided to simplify connection procedures and through which a window will display for setting database type, table and authentication. The connection string will be generated automatically (see the chapter on Alarm Management for more information on this argument)

## Step 5 : Inserting Template models

The last wizard window is used for selecting template models to insert in the project that can then be sued as models to configure for generating alarms or data recordings. If you do not wish to create one now, ignore this page and press the "next" button. Template models can always be created later on while in design mode. If you wish to create Template models proceed as described below.



Window used for setting project Alarms and DataLoggers

### Alarm Prototypes

You can select and add Digital and Analog Alarm prototypes (template) to the project. the Digital Alarm template only has one activation threshold while the Analog Alarm has four intervention thresholds.

During design mode you will be able to associate Tags and other properties to the Alarm Prototype.

### Historian Prototype

You can select and add Data Recording prototypes (templates) in the project using the proposed models:

- Record on change
- Record every 5 secs.
- Record every 30 secs.
- Record every 1 min.

During design mode you will be able to change the recording times, add other new models and associate tags and other properties.

### End of Wizard

The last window enables you to confirm the previously defined settings. Upon confirmation a new project will be created allowing you to **continue on with designing the project by adding, modifying or deleting any of the desired configurations.** You can always return back to any of the previous wizard steps to change those settings you have already defined by using the "prev." button.



## 3. I/O Data Server

### 3.1. How to create Tags

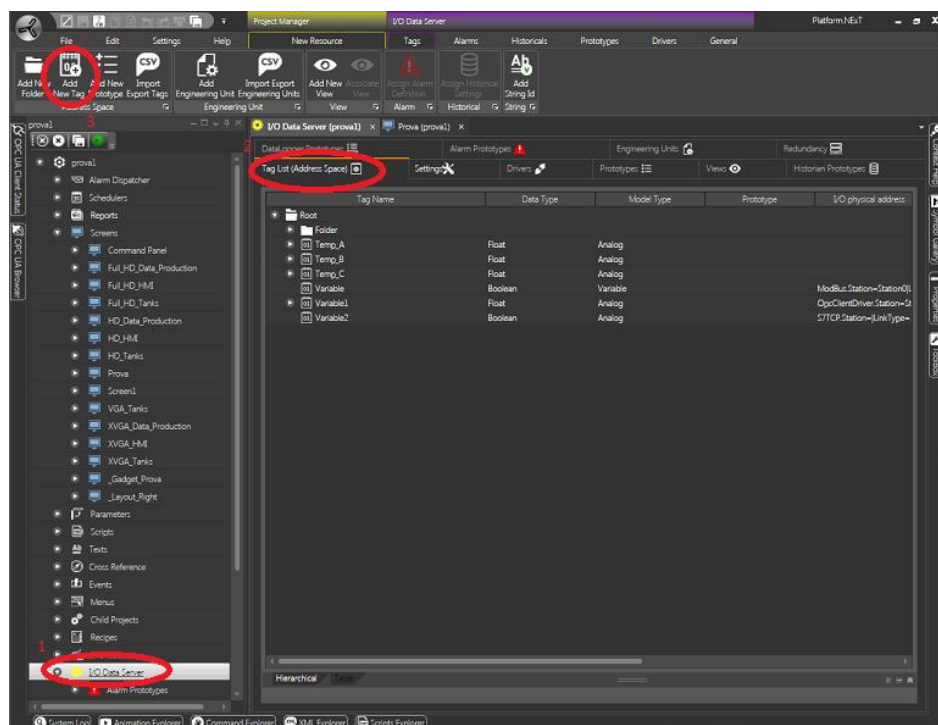
The project Tags are the basic elements of dynamic information which connect to PLC tags or field devices by using communication drivers to then connect to the supervision project's objects and functions.

Project tags are defined on the platform's server side and added to the I/O Data Server module's resource defined as "**Address Space (Tag List)**" located in the project's tree structure.

#### 3.1.1. Insert a new tag

To create and insert a new tag in the project proceed as follows:

1. Select the **I/O Data Server module** from the left hand side of the workspace in the Project Explorer window.
2. Open the "I/O Data Server" group's "**Address Space (Tag List)**" resource with a double click in the Project Window to display the list of project tags within the workspace.
3. Use the "**Add New Tag**" command from the "I/O Data Server-Address Space" ribbon to add a tag to the list.



The list of Tags inserted in the project will display when opening the I/O Data Server's Address Space resource.

The list can be displayed in hierarchy mode (tree structure) or Grid mode.

4. When activating the add new tag command a window will display with the new tag's properties to be set. This window should contain at least the 3 main tag parameters:
- Tag Name
  - Data Type
  - Physical Address (if available)

**Tag Properties**

Name: Variable

Description:

Model Type: Analog

Data Type: Float

Engineering Unit:

I/O physical address: Demo.Station=Station0|LinkType=

☒ Retentive

Array Dimension: 0

Initial Value:

OK Cancel

*Tag property settings window*

The Tag property window can be accessed whenever needed by double clicking on the variable selected in the Address Space.

### 3.1.2. Tag Properties

The Properties Window is used for modifying the various tag properties:

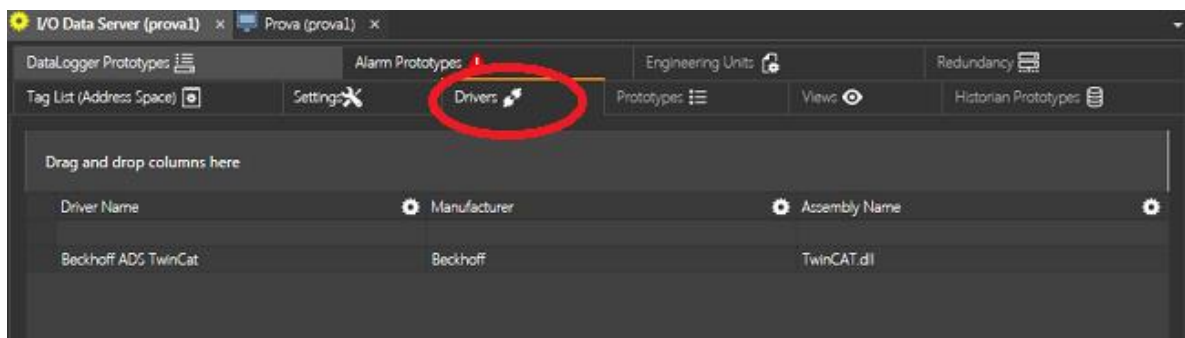
<b>Name</b>	Name of the tag. The tag must have its own unique name within the group it belongs to
<b>Description</b>	The tag description string can be entered as pleased (field not mandatory)
<b>Model Type</b>	Data model type definable as analogical value, method, enumerator type or other type for example.
<b>Data Type</b>	Tag data type field is only available when selecting certain Model Types such as an "Analog" Model Type. In this case the data type can be defined as Boolean, Byte, Int32, or Float.
<b>Prototype</b>	This field is used for selecting the prototype to define the tag with. This field is only available if "Object Type" has been selected as "Model type". The desired Prototype to be selected must be defined in the "Prototypes" resource beforehand in order for it to be selected within this property.
<b>Engineering Unit</b>	The engineering unit to be associated to the tag. The engineering unit must be defined in the appropriate resource beforehand in order for it to be selected within this property. (field not mandatory)
<b>I/O Physical Address</b>	I/O physical address of the memory area of the device to be connected to (e.g. PLC). In order to connect to device you will need to set up at least one communication driver in the project beforehand: for further information on how to insert a link please refer to paragraph on "Tag I/O Physical Addresses"
<b>Retentive</b>	This check box is used for enabling the variable's retentivity so that its value will be saved on file for use in the next project startup.
<b>Array Dimension</b>	This field is used for defining the number of Array elements the Tag must have if set as Array type in the Data Type property. When leaving this field set at zero the variable will not be considered as an Array type but only as one single variable type.
<b>Initial Value</b>	This field is used for specifying which value the variable is to be initialized with at each project startup (field not mandatory)

## 3.2. Adding Communication Drivers

The Automation Platform.NExT platform manages realtime communication with field devices using the OPC UA standard technology, through using Communication Drivers that manage proprietary protocols towards PLCs and the most popular field networks. In this Tutorial we will teach how to implement Communication Drivers and put them into action.

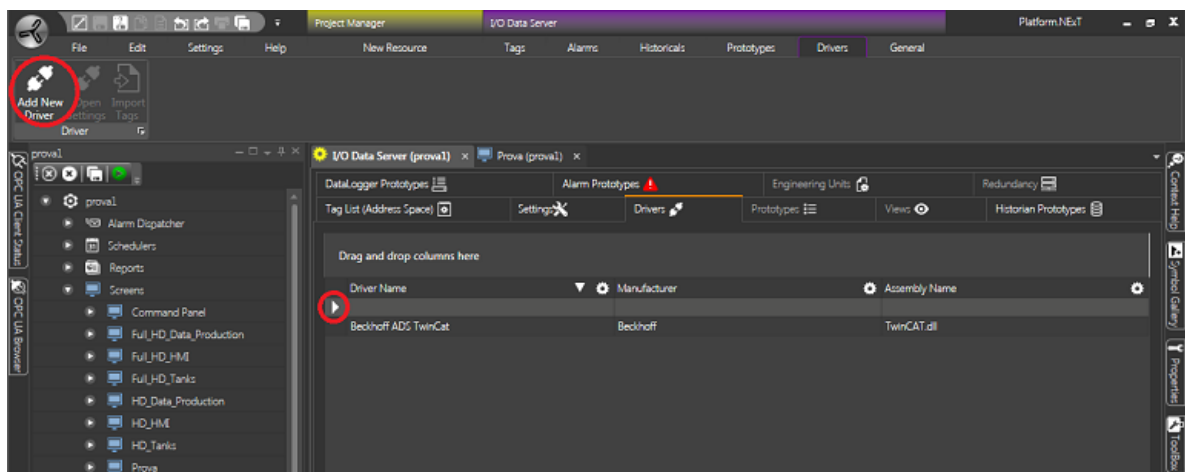
The Communication Drivers are components managed by the I/O Data Server module, and can be added to the project and configured using the **I/O Data Server's I/O Drivers** window'.

To add a new Communication Driver to the project select the project's I/O Data Server and open the "I/O Drivers " resource.



The I/O Data Server's "I/O Drivers" resource contains the list of Drivers that have been installed in the project.

If Communication Drivers have not yet been installed in the project, an "Install Driver" button will show in the centre of the "Drivers" window. By using this button or the "Add New Driver" command from the "Server Address Space" Ribbon a Driver selection window will open:



Window used for selecting the Driver to be installed in the project. Note that the drivers are listed by manufacturer whereby one folder may contain a number of drivers.

To add a new driver in the project proceed as follows:

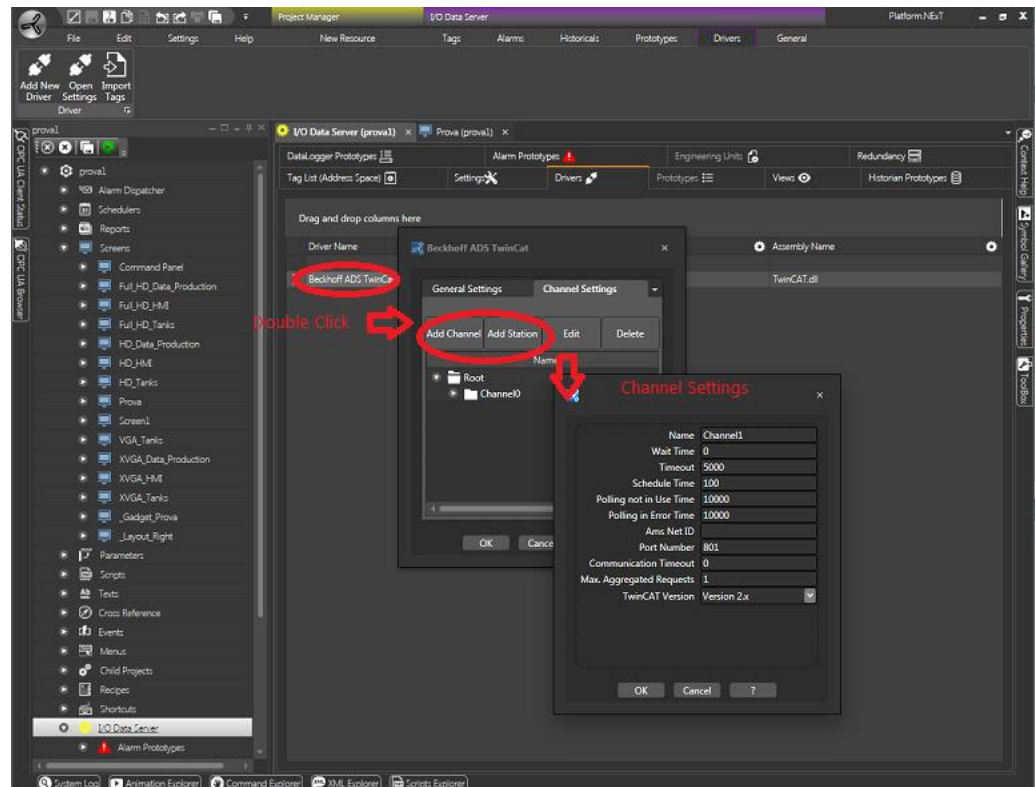
1. Select the "Add new Driver" command

2. Select the driver desired from the those listed in the Driver selection window and confirm with OK.

### 3.2.1. Configuring the Driver

After having inserted a Driver in the project's I/O Data Server module, you can now configure it by defining the communication parameters relating to the protocol type and physical device used.

Double click on the name of the Driver to open the Communication Driver Settings window.



The Driver's settings window presents three setting levels:

1. **General** Settings : defines the main parameters needed for driver communication.
2. **Channels** Settings : each driver can provide communication through one or more channels. At least one channel must be defined.
3. **Stations** Settings : each driver's communication channel can be defined with one or more stations. At least one station must be defined for each channel.

Some of the parameters presented in the three setting forms are the same for all the Drivers, while others are specific to each individual Driver depending on the protocol type used by that Driver. **Therefore please refer to the documentation of each individual driver.**

### 3.3. Tag I/O Physical Addresses

After having inserted the Tag and inserted and configured the Driver you can now proceed with linking the Tag to the field device's physical address.

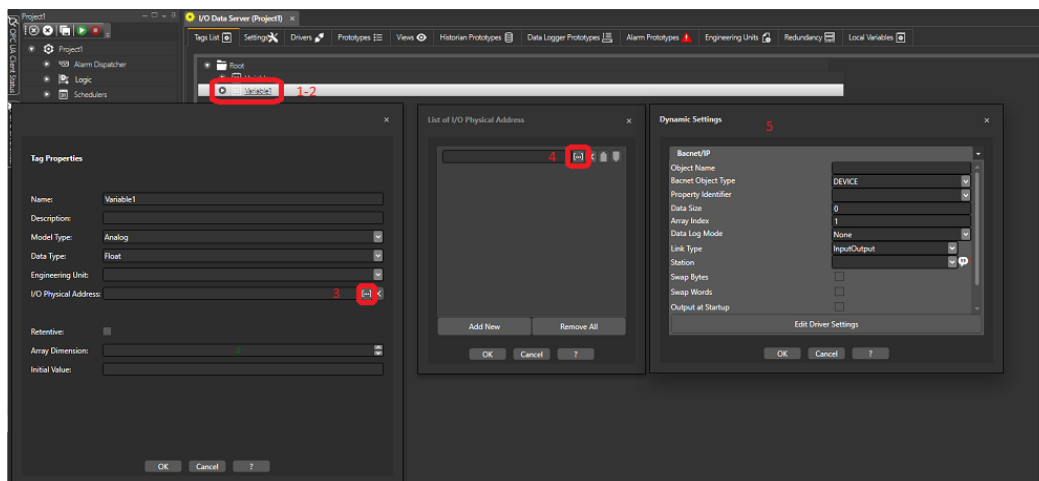
Each Tag can in fact be configured with the physical address of the device to be connected during project runtime through its properties. The Tag property used for doing this is called the 'I/O Physical Address' property.



Before inserting a dynamic link in a Tag you must insert and configure a Communication Driver in the project as described in the paragraph on "Adding Communication Driver". When doing this make sure that the driver is defined with the Channel and Station settings as well.

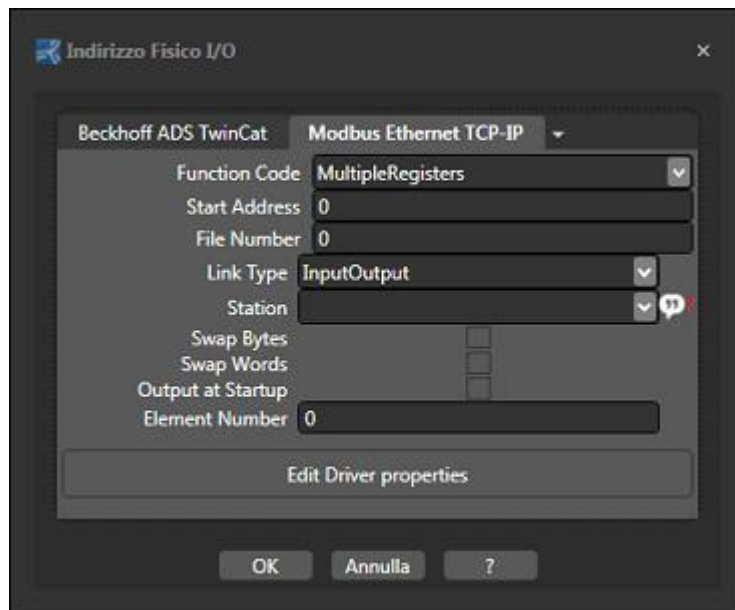
At this point proceed as follows:

1. Select a previously inserted **Tag** in the project, or create a new one.
2. Open the Tag's property settings window with F4 (right click -> edit-> Properties).
3. Select the "I/O Physical Address" property and access its settings using the button on the right.
4. Use the "**Dynamic Settings**" configuration window to set the physical field address for the desired Communication Driver.



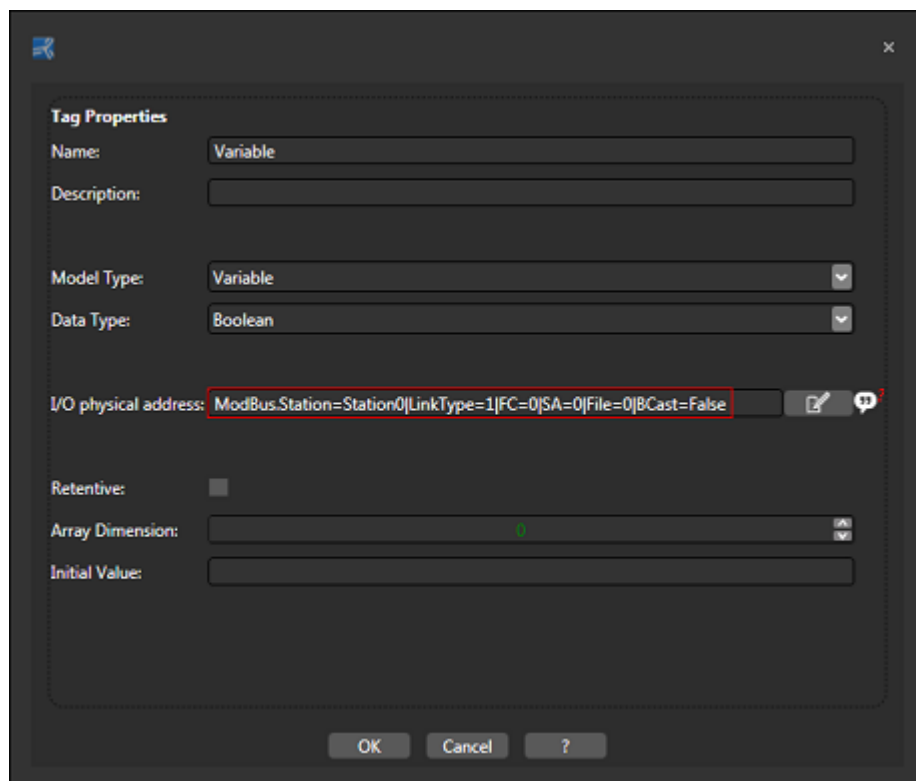
*Dynamic Tag link settings as physical field device address.*

5. The list of drivers installed in the project is displayed at the top of the "Dynamic Settings" window shown in separate Tabs. Select the desired Driver to populate the "Dynamic Settings" window with its related fields to be set.
6. For instance if you select the "Modbus Ethernet TCP-IP" Driver, the parameters that will show for defining their settings are typical of Modbus, such as "Function Code", Start Address" and "Link Type".



*Dynamic Tag link settings window for the Modbus TCP-IP Driver.*

7. After having selected the connection parameters to the field device memory address, confirm with "OK". The Driver connection string will be shown in the Tag's "I/O Physical Address" property. This connection string summarizes the settings of the various parameters defined in the "Dynamic Settings" window:



*A tag's property setting window with the "I/O Physical Address" field*



The conditional variable accepts tag types: Boolean, SByte, Byte, Int16, UInt16, Int32 and UInt32



Note: The Driver connection string can be directly typed in the Tag's "I/O Physical Address" as long as great care is taken inserting the syntax correctly. For example:  
`ModbusTCP.Station=Station0|LinkType=1|FC=2|SA=354|File=0|BCast=False`

After having configured the Tag's "I/O Physical Address" correctly, the Data Server will then be able to establish communication with field device during project runtime to read data from the PLC and update the Tag with it in real time. Data can be read or write, or write only according to the settings defined in the Tag's "I/O Physical Address" property.

### Gateway Server

As from Movicon.NEXt 3.1 version it will be possible to assign a Tag with a list of several dynamic addresses so that it can connect to several different drivers at the same time.



The dynamic links to be associated to the same variable must be links to different Drivers. It is not possible to insert two links of the same driver to the same variable.

The dynamic links associated to one single Tag can be of any type such as Input, Output or Input/Output for example. However, in cases where more than one dynamic Input/Output link has been associated to the one same tag, different devices might perform input operations at the same time. If this happens, the last task performed will be the one that will value the tag.



It is not possible to import the same variable using the importer of two different drivers in order to have both links in the same Tag. The second importer may overwrite the already existing variable and therefore the initial dynamic link may be replaced by the new one.

## 3.4. Importing Tags

Generally the automatic tag import functions are preferred when already defined by the PLC programmer or by configured data of the device to be communicated with. Automatic Tag importing saves time and there is no risk of transcription errors.

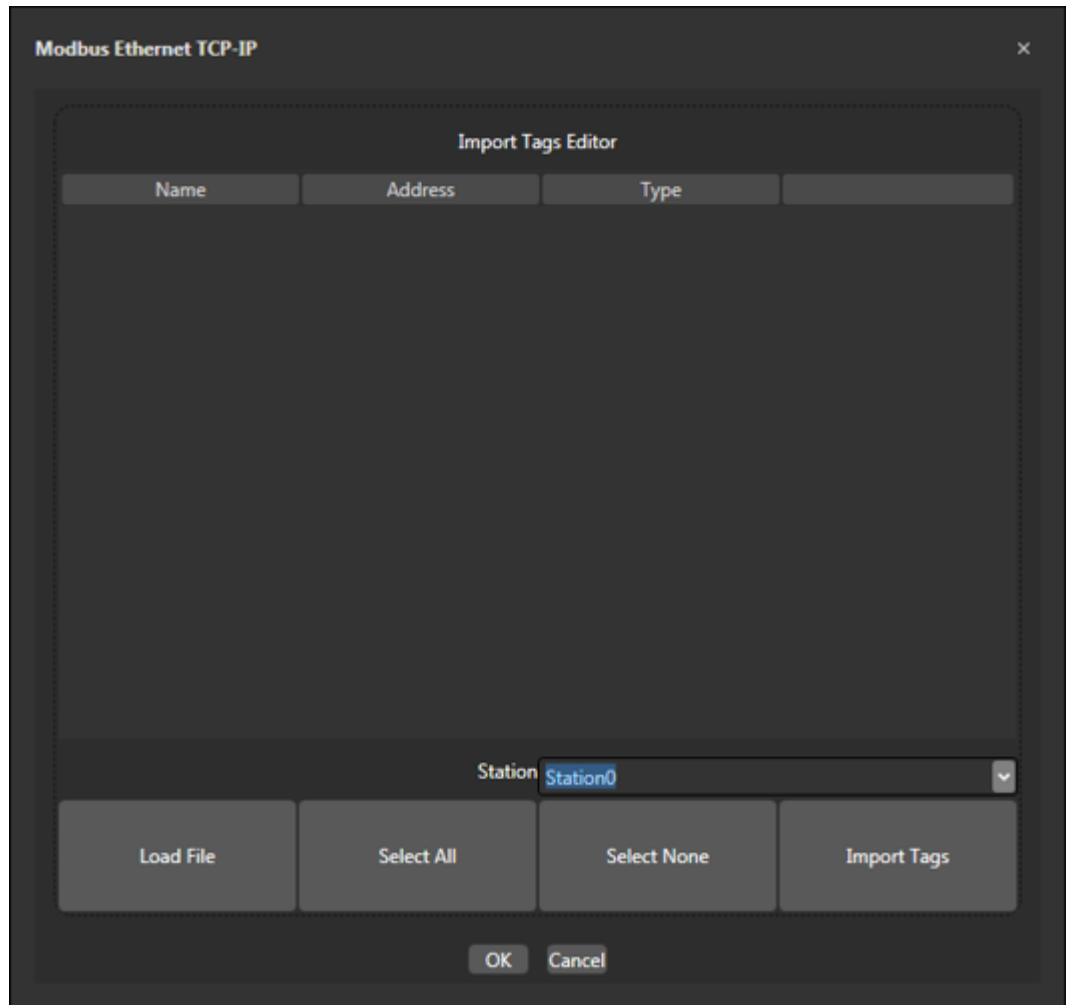


To use the automatic Tag import feature, you will need to access the **Driver di comunicazione** settings. Each driver has its relating "importer" that selected the source file of the data to be imported and interprets the data according to device type and its characteristics to automatically generate the list of Tags that should consequently be defined with **Name**, **data type** and **physical address**.

1. To proceed with importing tags automatically you must first configure the Driver as described previously.



2. Select the desired driver from the list of Drivers installed in the project and then right click->Import TAGS.
3. This window will display:



1. From this import window select the Driver's reference station using the "Station" combo box.
2. Select the data source using the "Load File" button. All the available tag names will display on the list within the window.
3. Select "All" or multi-select the data to be imported
4. Confirm with OK.

At this point, the projects' I/O Data Server's Address Space will be populated with the list of Tags imported with the operation.



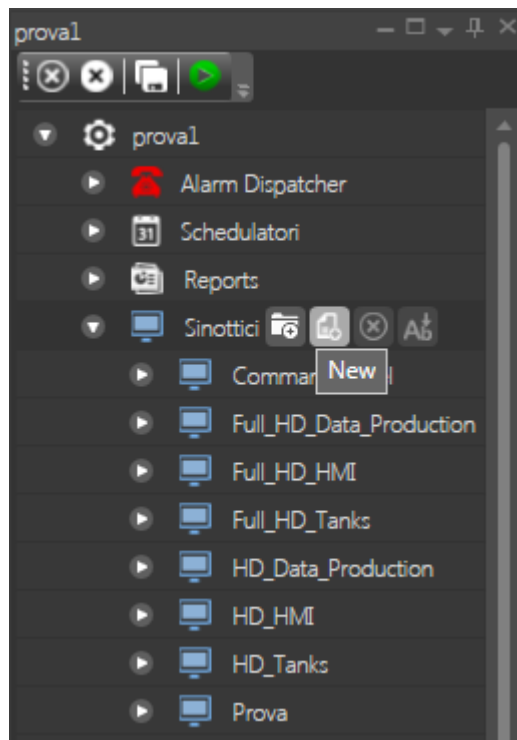
## 4. Synoptic Editing

### 4.1. Create a Screen

The project's graphical interface is managed using the Platform.NexT's main **Movicon.NExT** module.

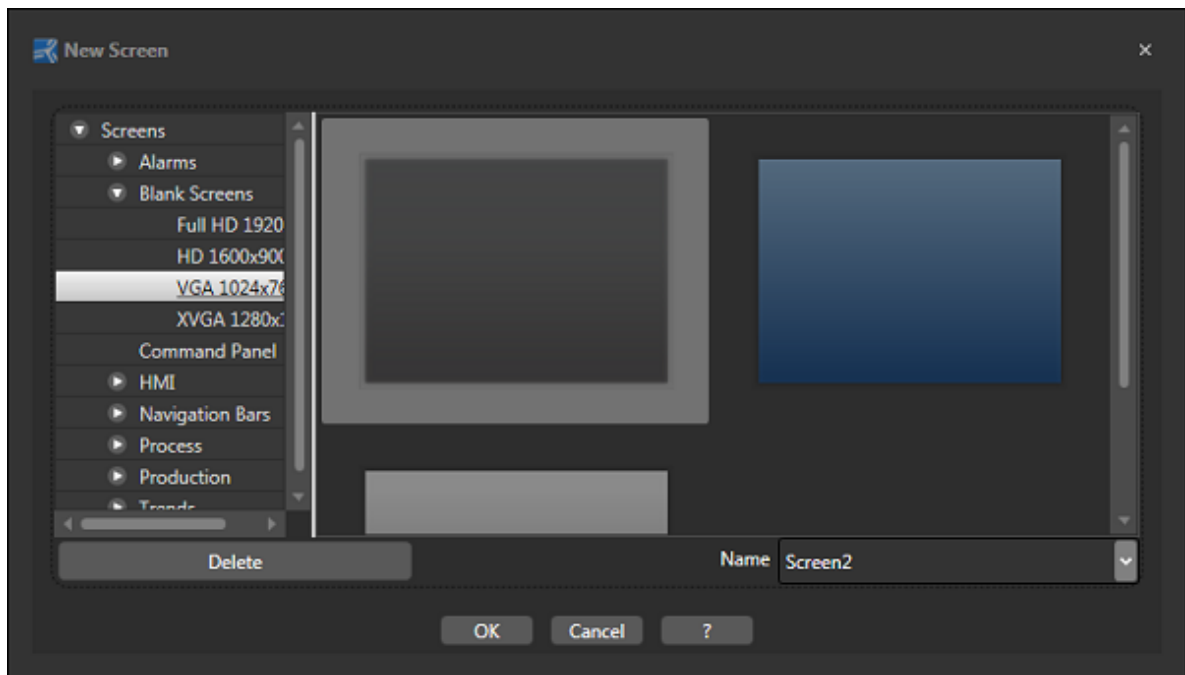
**The screens** are fundamental resources used for creating graphical project screens. To add a new Screen to the the project proceed as follows:

1. Go to the Project Explorer Window and select the **"Screen"** folder from the tree structure. Keep the mouse pointer on the resource to display the tooltip with the enabled **"New Folder"** and **"New"** commands. The **"New Folder"** command is used for adding folder within which to insert the screens. The folders are only needed for organizing screens into groups. The **"New"** command is used for inserting the new Screen.



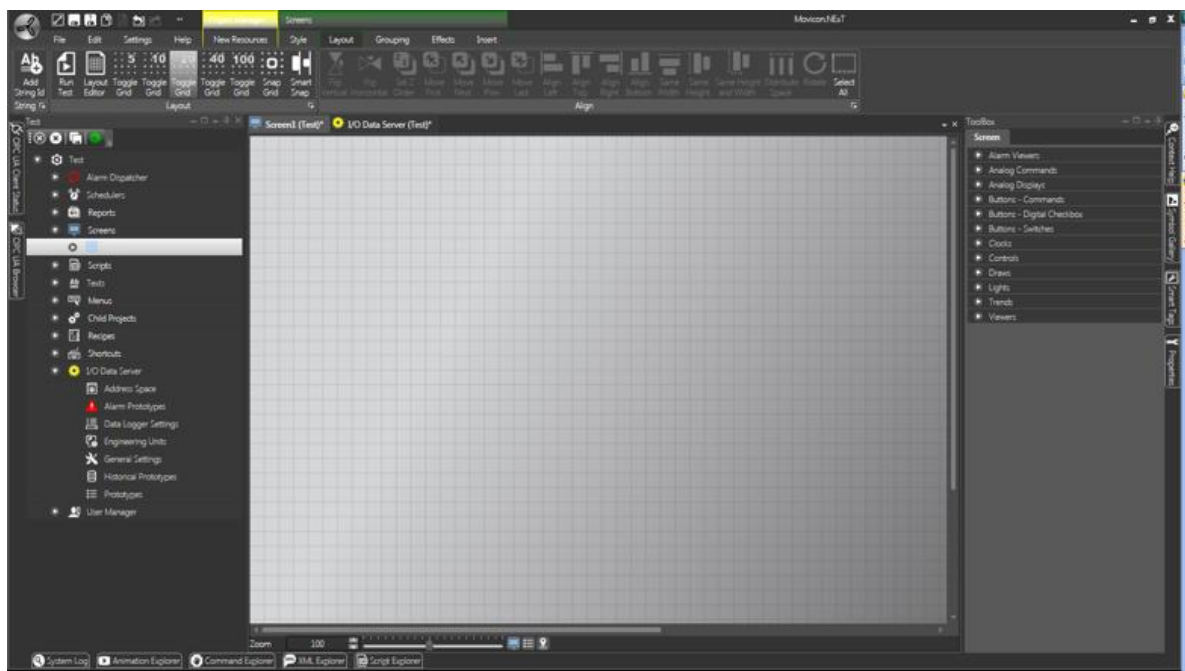
*A new Screen is inserted by using Project Explorer Window.*

2. Once the "New" command has been pressed a window will appear enabling you to select a Create a Screen to use. As an alternative to inserting screens with the stand default settings Templates offer a selection of ready-to-use screens whose background colors and sizes can be customized as preferred. The list of Templates is made available when installing Movicon and contains a limited number of different screens that can to be added with new Screen Template created and saved by the programmer.



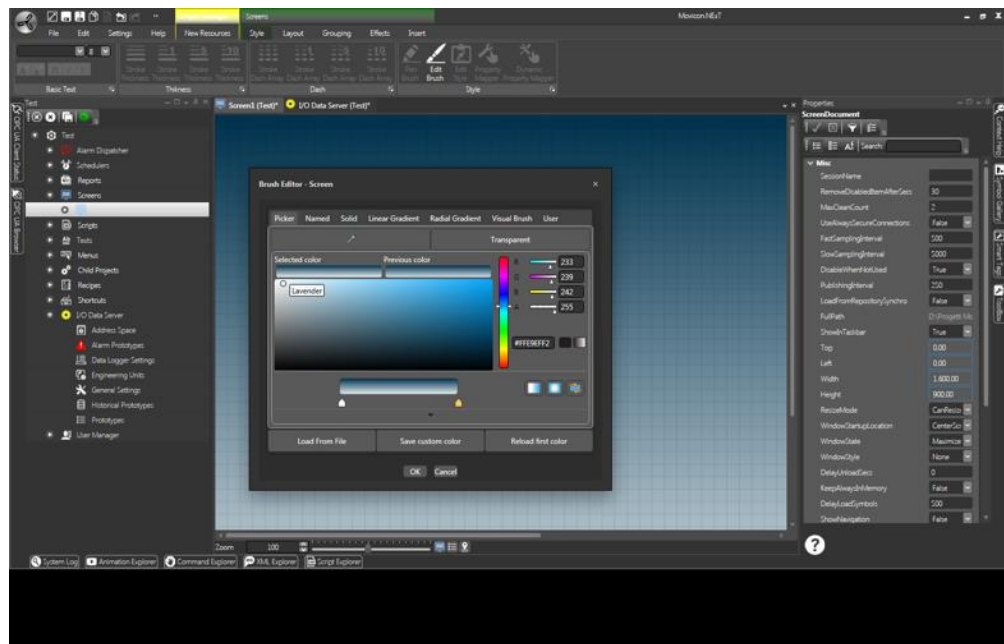
Window used for selecting a Screen Template to use for creating a new Screen.

3. Select a Screen Template from those available, enter a name for the screen in the appropriate field and confirm with "OK" to insert the new screen. Upon doing this the screen will also automatically open within the Movicon Workspace.



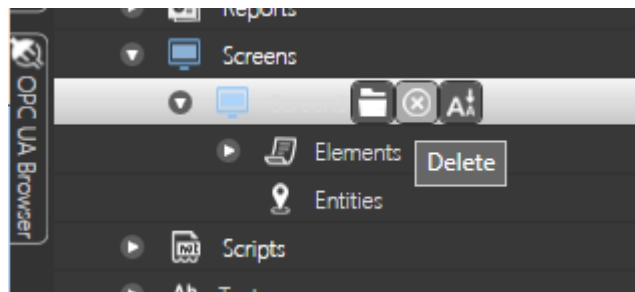
New Screen opened in the Movicon workspace.

4. Use the Properties Window to modify the screen's properties. Some properties can also be modified directly by using the commands in the Ribbons. For example, by clicking the **"Edit Brush"** command from the "Screen Manager - Style" Ribbon a colour selection window will open for the screen's background:



Window used for selecting a screen background colour.

- Once a screen has been inserted in the project it can be deleted (Delete), have its name changed (Rename) and opened (Open) in the Workspace using the appropriate commands which display as tooltips when using the mouse to select the name of the screen from Project Explorer Window.

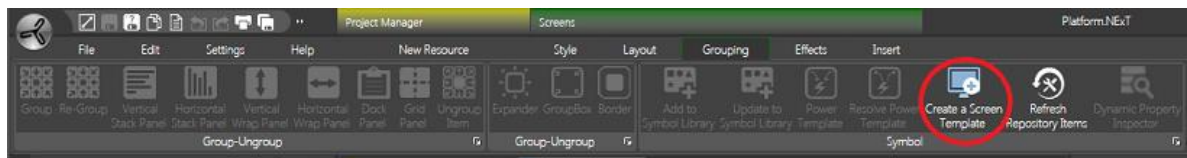


The Open, Delete and Rename screen commands can be activated within the Project Explorer Window.

## Screen Templates

The Screen Templates are screen models that are used for creating new project screens. To create a Screen Template proceed as follows:

- Open a Screen in the Movicon Workspace.
- Configure as pleased, by giving it a different background color, size in pixels or company logo.
- Assign a name indicating screen template type to be created.
- Activate the **"Create a Screen Template"** command from the "Screen Manager - Grouping - Symbol" ribbon or textual menu which appears when right clicking on screen.



To create a Screen Template simply open the desired screen in the Workspace and activate the "Create a Screen Template" command.

- When the create screen command accomplish its task successfully Movicon will show the "Screen Template has been created succesfully!" message. From this point onwards it will appear available for selecting along with the other screen templates on the list when inserting a new screen.

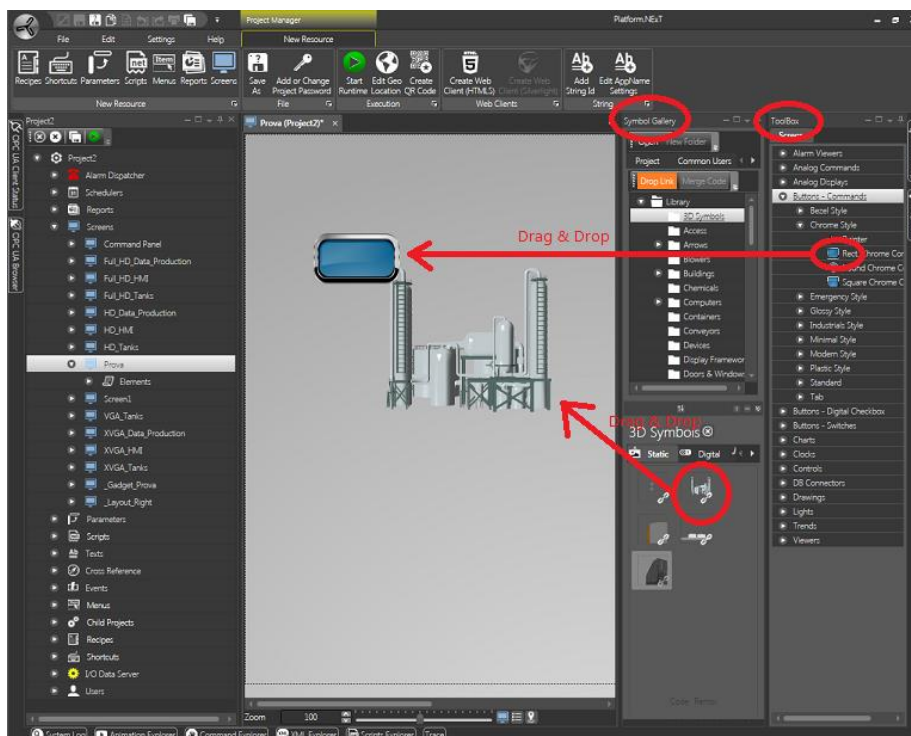
Screen Templates are saved in this folder:

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

Use the files within this folder for **deleting or renaming Screen Templates**.

## 4.2. Inserting Objects in Screens

When a screen is opened in the Movicon Workspace, the "Screen Manager" Ribbon is activated along with the "**Toolbox**" and "**Symbol Gallery**" windows. These two windows can be left hidden or docked in the foreground as preferred.



The windows used for inserting objects and symbols are made available when opening a Screen in the Workspace



Objects and symbols can be dragged from the Toolbox and Symbol Library window and dropped on the screen using the "Drag & Drop" techniques.

#### 4.2.1. Graphics Editor

The Movicon graphics editor can be used as pleased to edit graphics by using the basic shaped provided in the Toolbox "Shapes" category.

The drawings can be configured as pleased, using their properties to define colors, gradients, stiles, transparency along with others.

The drawings you create can then be grouped in Symbols and added to the Symbol Library as we see later on ahead.

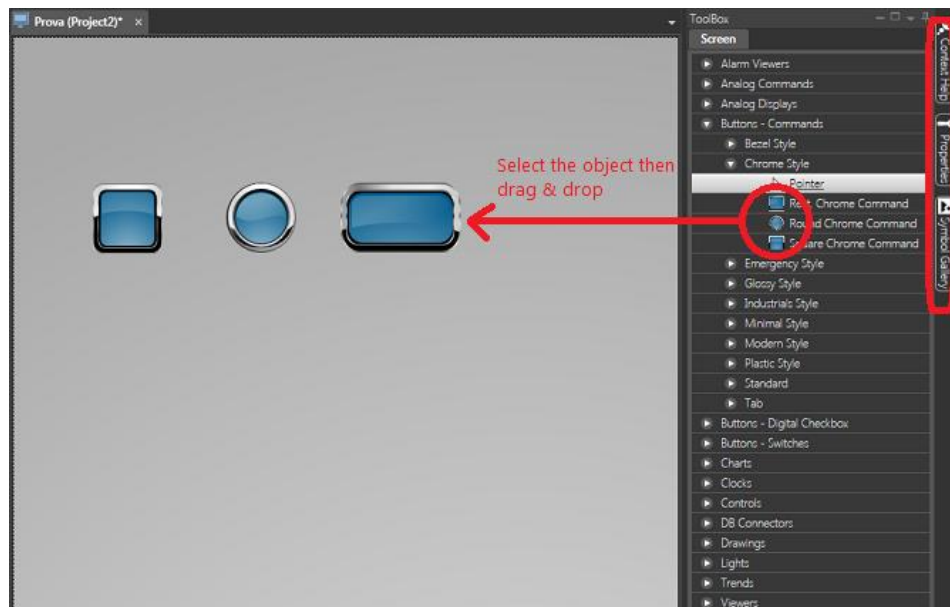
#### 4.2.2. Importing Drawings

The graphical designs can also be imported from external using the standard XAML format.

When using the "Import XAML" command from the "Insert" group ribbon, you will be able to select a XAML graphics file and turn it into a Movicon symbol with all the effects on your screen and insert it for later use in the symbol library.

### 4.3. 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.

### 4.3.1. Inserting objects from the Toolbox

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

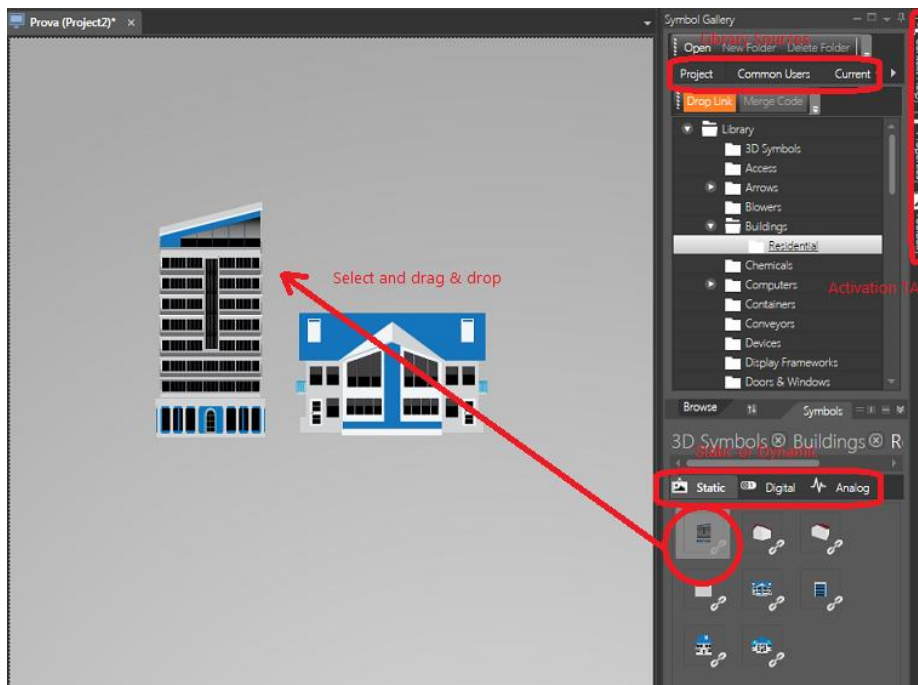
1. Open the screen of interest within the Movicon workspace
2. Activate the Toolbox if not already opened in foreground
3. Open the object group desired by clicking on the icon to expand the group
4. Select the object you which to insert on screen
5. Position mouse pointer with attached object onto the 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.

## 4.4. Symbols Library

The Movicon Symbol Library contains ready to use predefined and configured static and dynamic symbols. This library can be accessed from the screens 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.





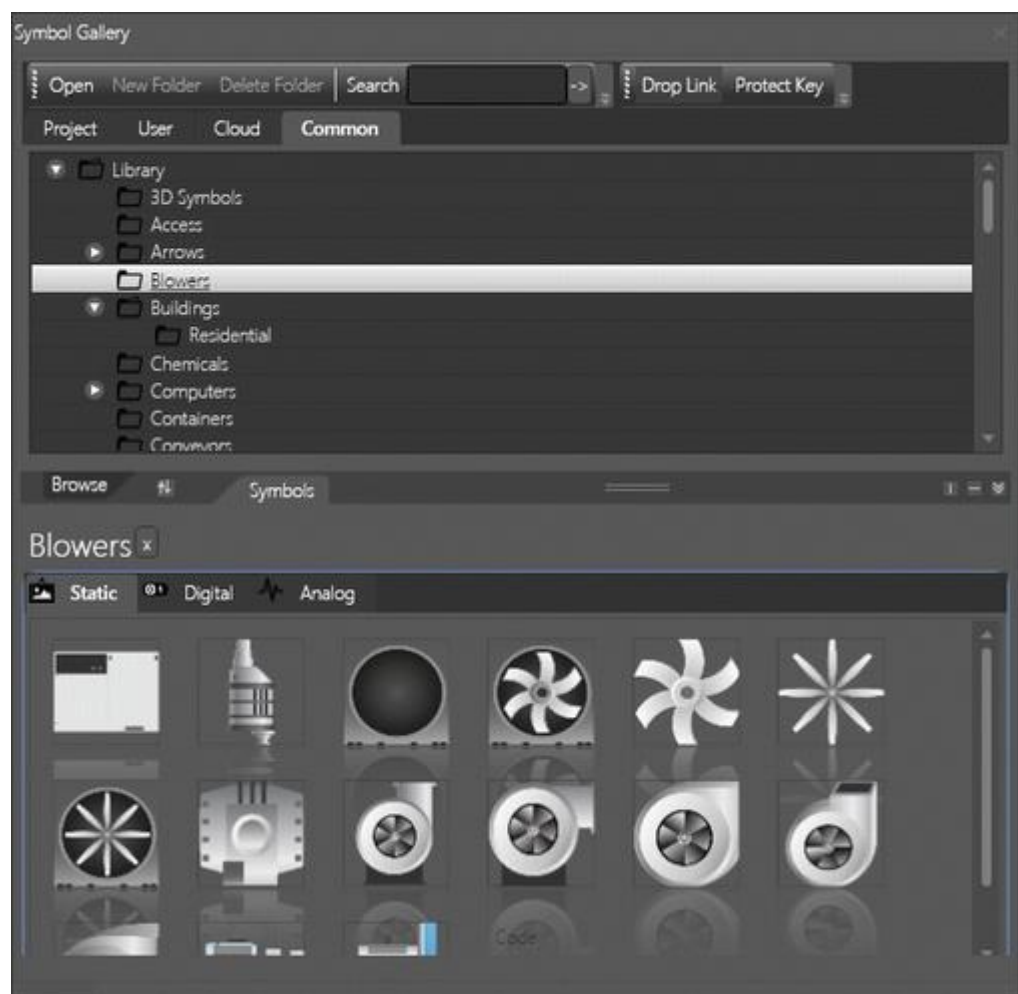
The symbols refer to different sources and are selected through the source related tabs:

- **System:** these are symbols installed with the Movicon.NExT module
- **System Cloud:** these are Progea symbols available in the Cloud
- **User:** these are symbols which can be freely defined by the user to reside in system folders as determined by the user or in the cloud.
- **Project:** are symbols relating to the project residing in the project folders.



The symbols are XAML-based and the library is protected and encrypted for protection purposes by default.

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.*

To insert an object or symbol on screen according to the window from which it is selected proceed as follows.

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

To insert a symbol using the Movicon Symbol Gallery follow these quick simple steps:

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

## 4.5. Screen Navigation

Navigating the various project screens can be established by inserting a command object from the toolbox on screen and assigning it the "change page" command, or by using the system commands without having to configure any programmed commands.

### Using Command Objects

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.

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

### Using System Commands

Movicon projects are predisposed for executing screen page navigation without obliging the programmer to insert navigation buttons (nevertheless always available). 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.



At the Startup of the project can be defined the "Main Page", in this way you can open the screen dedicated as the "Main page", containing the change page commands that the programmer may have Defined.

### 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.

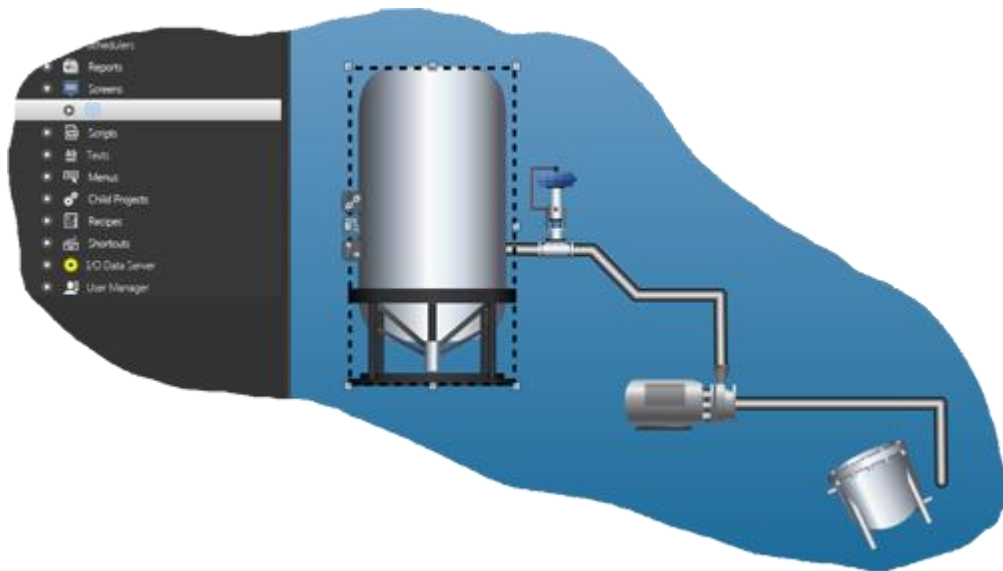
The system Toolbar commands in runtime are listed below starting from the left:

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

## 4.6. Editing Graphical Objects

After having inserted an object or symbol on screen it can then be resided and positioned as pleased and configured through its **Proprieties**.

In order to do this the object must first be selected with the mouse. When the object is selected it will become outlined with a rectangle consisting of eight small squares which are used for re-sizing the object using the standard Windows techniques.



*Objects and symbols selected on screen.*



Movicon permits multiple object selections, grouping of complex symbols, rotation, alignment, and overlap commands along with many other applicable features.



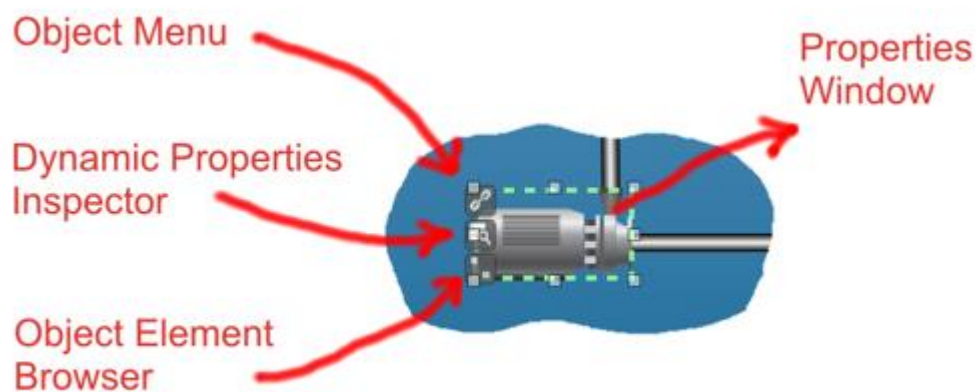
Intersection or points can be added to or removed from the basic drawing shapes, such as polygons, polylines and pipelines, using these commands:

- **CTRL+Click = Add point or intersection**
- **SHIFT+Click = Remove point or intersection**

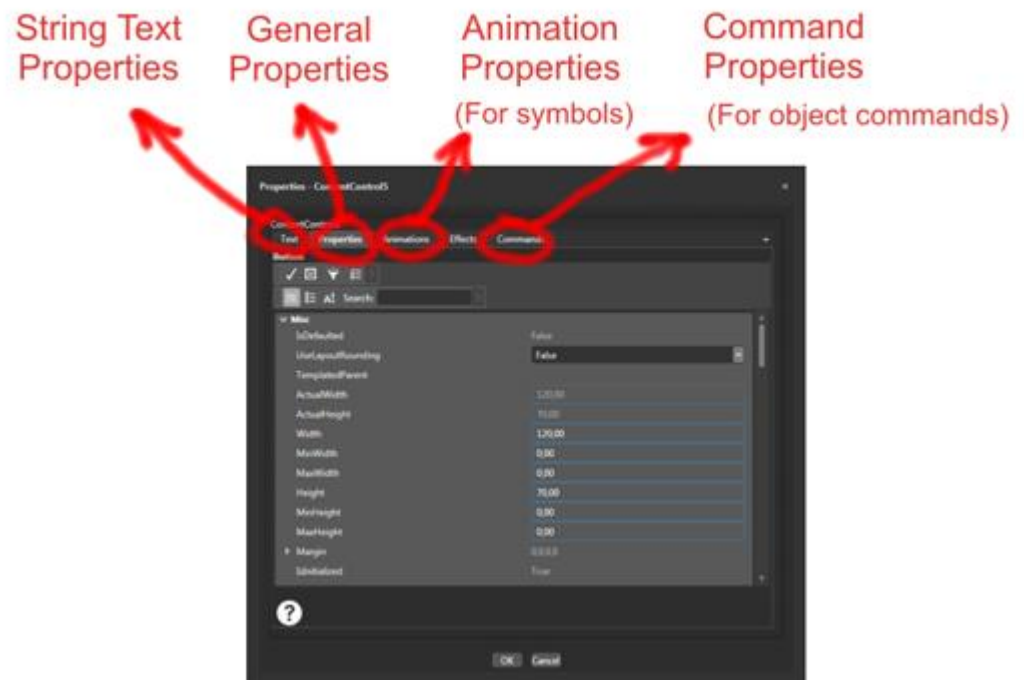
#### 4.6.1. Editing Objects

Each object or symbol can be configured as pleased by means of using the many style options and features offered by the platform.

The main configurations are accessible through the Contextual Menu or the Properties Window.



When double clicking an object, the General Properties window will display through which the object's properties can be defined by selecting the tab corresponding to each property group.



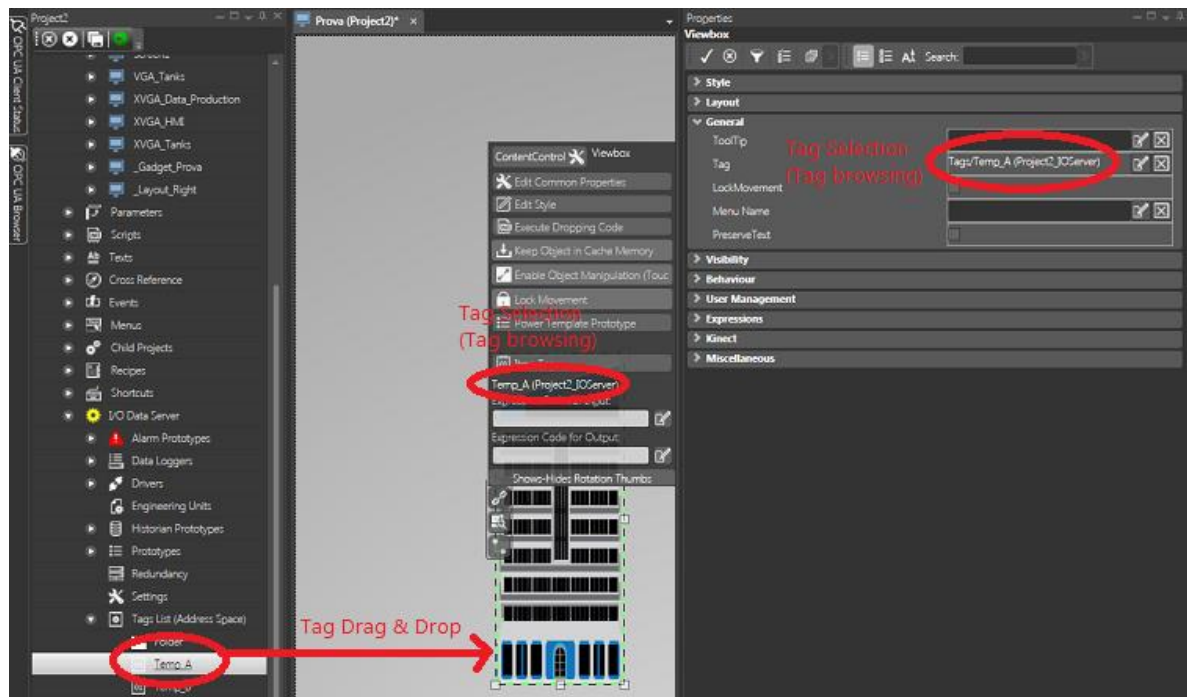
Not all the object's numerous property configuration possibilities will be described in this Tutorial but only those which are essential and helpful to convey the fundamental concepts using just a few clicks of the mouse.

## 4.7. 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.

### 4.7.1. 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

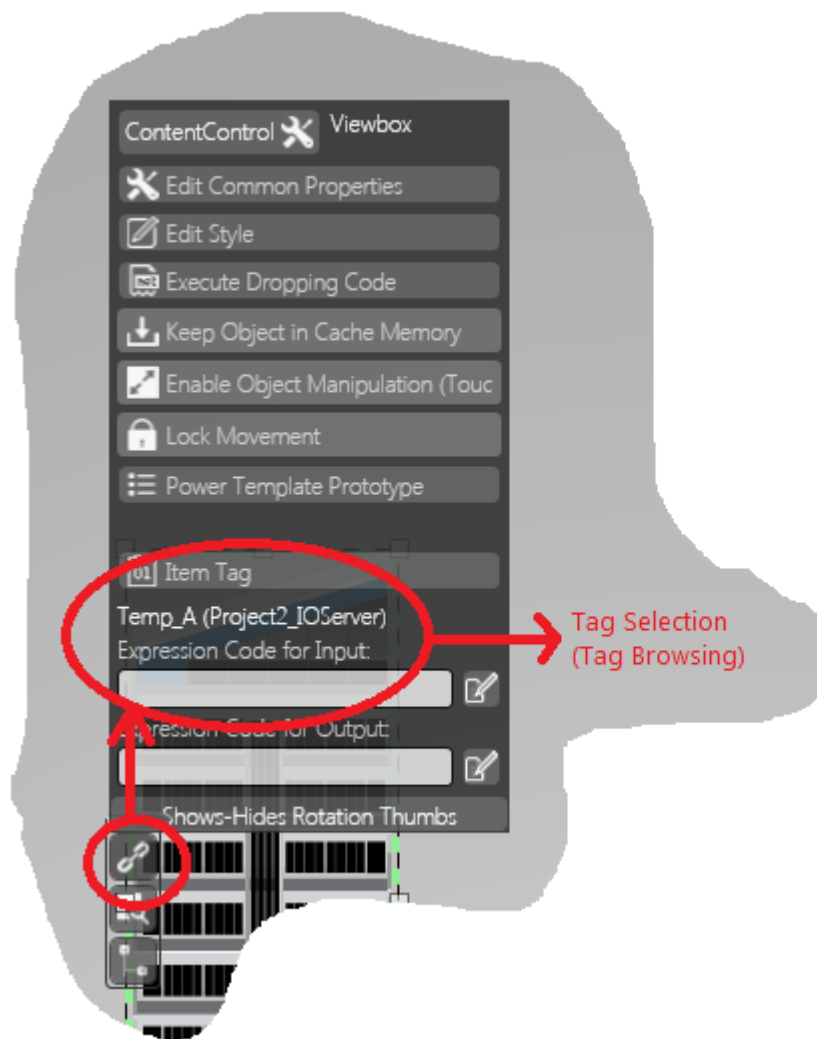
A Tag can be assigned to an object by simply dragging it from the list to the object using the Drag & Drop techniques.  
In order to do this proceed as follows:

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

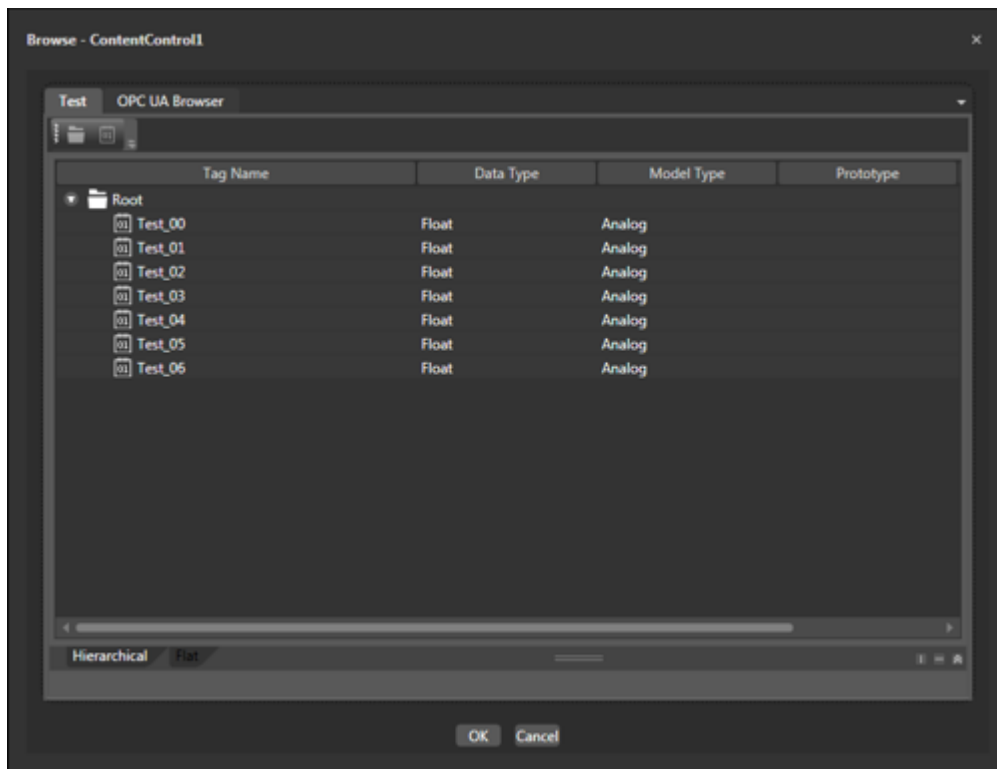
The Tag Browser window can also be used for selecting a Tag to associate to an object. The Tag Browser window can be accessed through the object's tooltip.  
To select a Tag using the Tag Browser window proceed as follows:

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.*

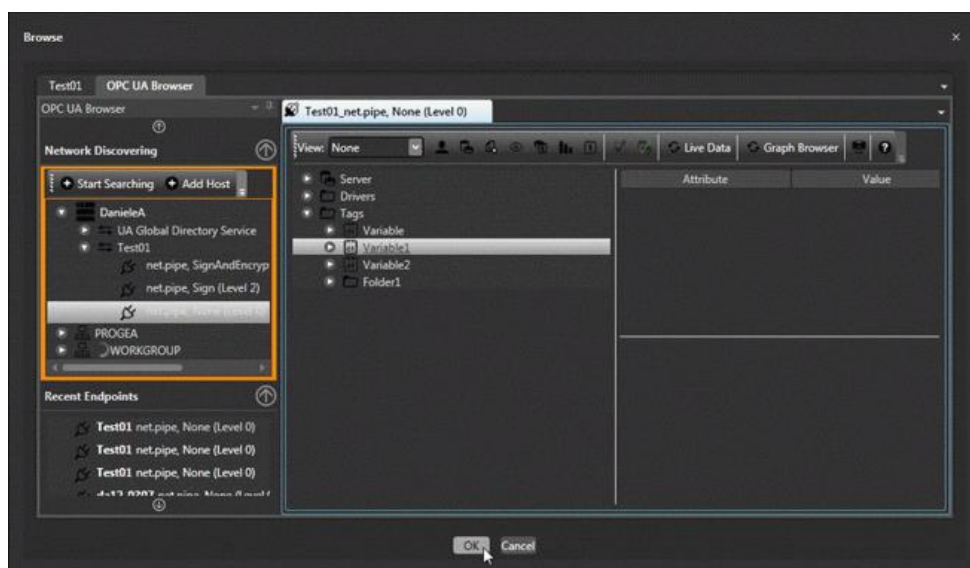
4. Select the Tag desired from the Tag Browser window and confirm with OK to associate the Tag with the object.



*An object Tag Browser window*

### Selecting an Item directly from a third party OPC UA server

It is also possible to select the **"OPC UA Browser"** tab from the Tag Browser window (can be reached from any object with a right click -> "Item Tag "), through which 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.*

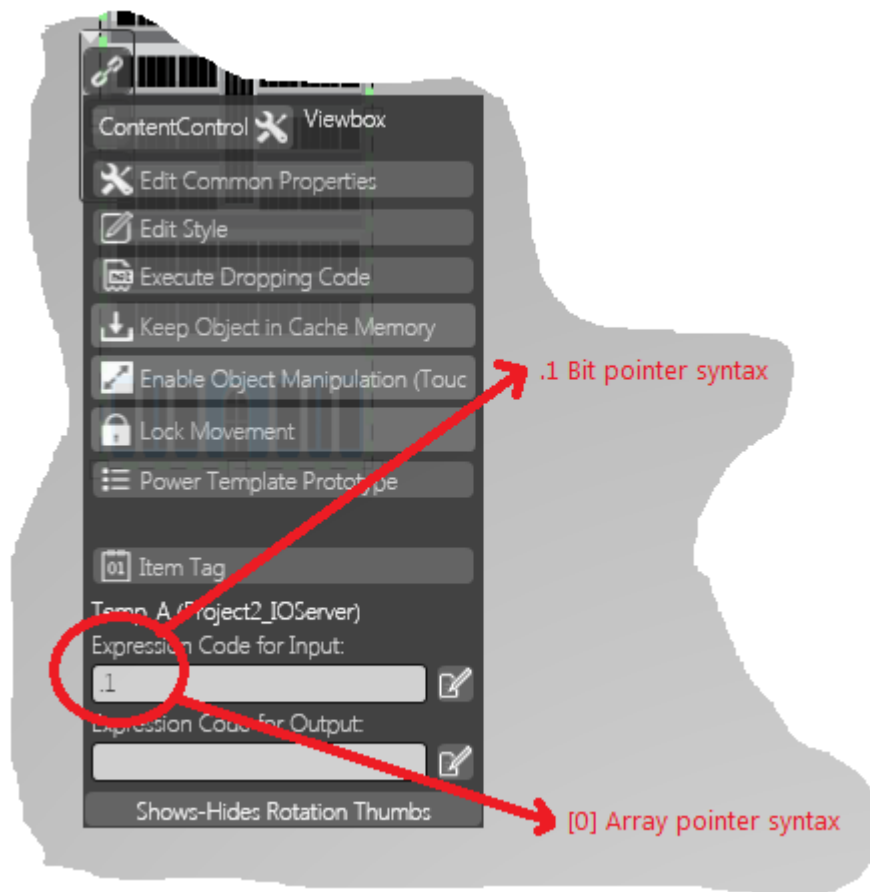


## 4.8. Tag Bit and Array Tag Pointing

In certain cases it may be necessary to display or set one individual bit of a Word or Dword type tag.

To do this proceed as follows:

1. Open a screen and insert a object, for example a Display
2. Assign the Display object with the Tag to be monitored
3. Open the object's command Tooltip and insert the "1" syntax in "Input Expression Code:" field to read/write the tag's 1 bit.
4. In project Runtime the Display will display/write only the Tag's 1 bit.



*Use the object's Tooltip "Expression:" field for pointing to individual Tag bits or array elements*

Very much of this procedure is the same used for pointing to individual array tag elements. In this case, after having associated the array variable to the object you must insert the number of items to be pointed to in the "Expression:" field by specifying the number enclosed within square brackets. For example to point to the first array element, being the 0 element, you must insert the "[0]" syntax. Proceed in this way:

1. Open a screen and insert an object, for example a Display object.
2. Assign the Display object with the Tag to be monitored.

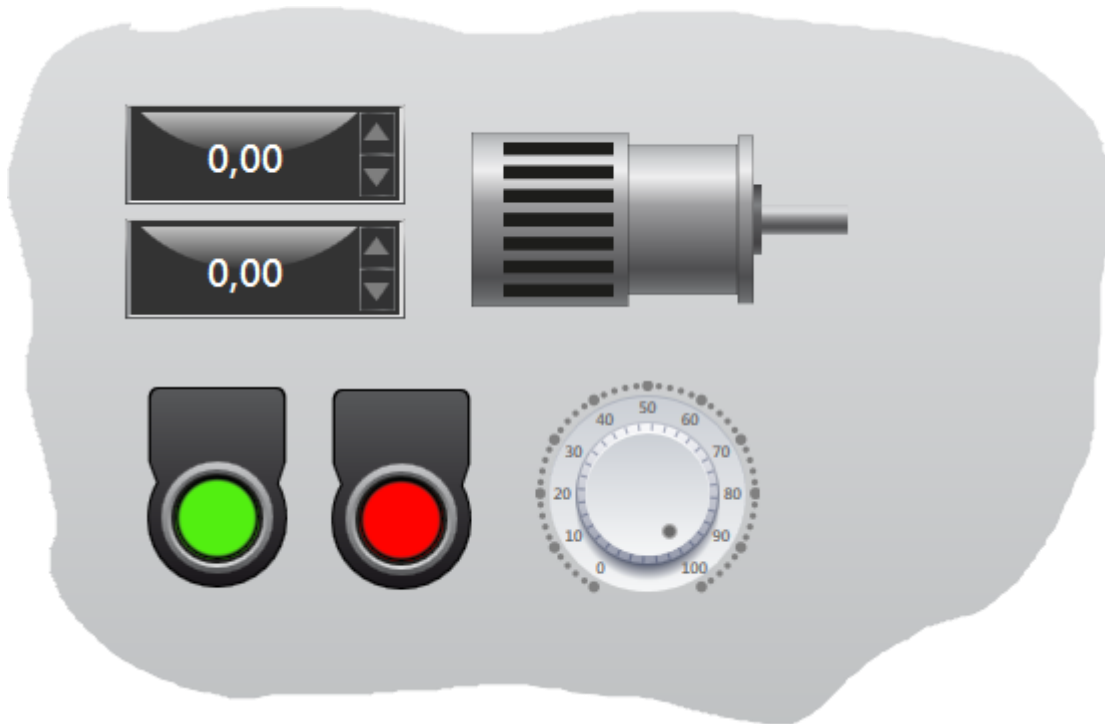
3. Open the object's command Tooltip and insert the "[0]" syntax in the "Input Expression Code:" field to read/write the 0 element of the Array Tag
4. The Display will display/write the Array tag's 0 element in project Runtime.

## 4.9. Creating Composed Symbols

The Movicon Controls and Objects can be grouped together different elements to create a new Symbol.

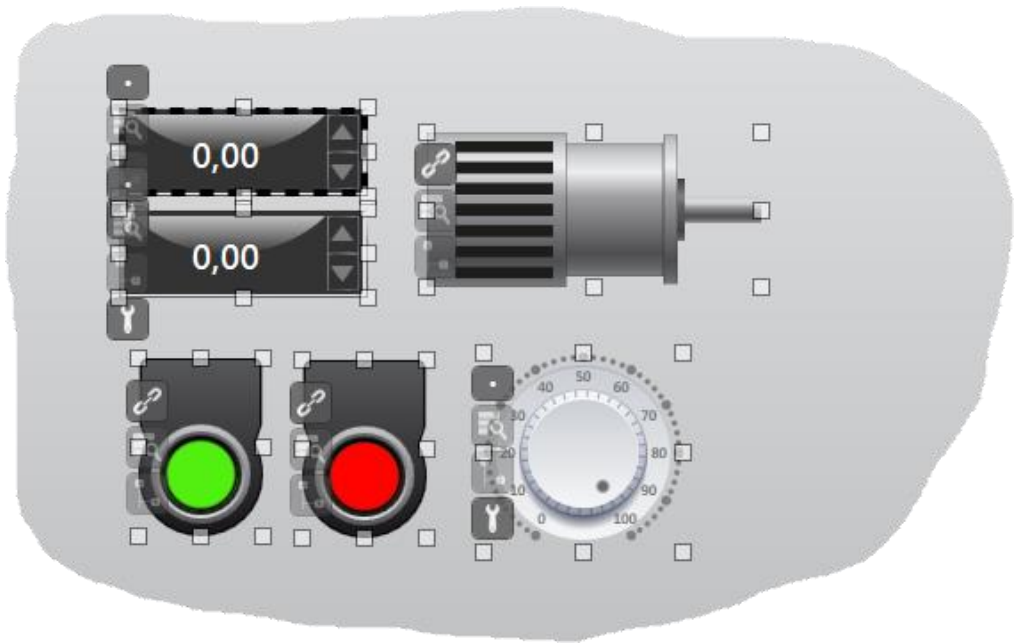
To create a composed symbol proceed as follows:

1. Add the Objects and Controls you wish to use for creating a Symbol keeping in mind that Symbols can be composed of elements from both the Toolbox and Symbol Gallery. In addition a Symbol can have other composed Symbols as elements.



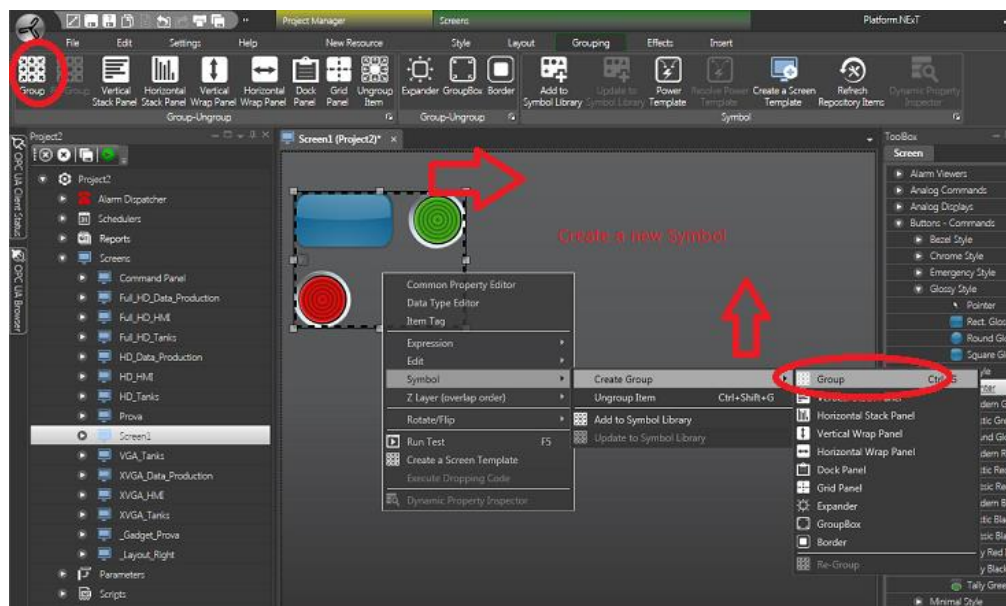
*These individual elements will be used to make a Symbol.*

2. Once the individual elements have been configured and sized as needed, select them all before applying the create Symbol command.



Before creating the a Symbol you must select the elements to compose it with.

4. The create Symbol command can be executed from the "Screen Manager - Grouping" Ribbon or from the contextual menu which opens when right clicking the selection of elements ("Symbol - Create Group" (menu) ). In this case use the "Group" command.



The Group command used to create a Symbol can be activated either from the Ribbon or the selected objects' contextual menu.

5. Once created the symbol can the be added to the Movicon "Symbol Gallery" by using the "Add to Symbol Library" command.



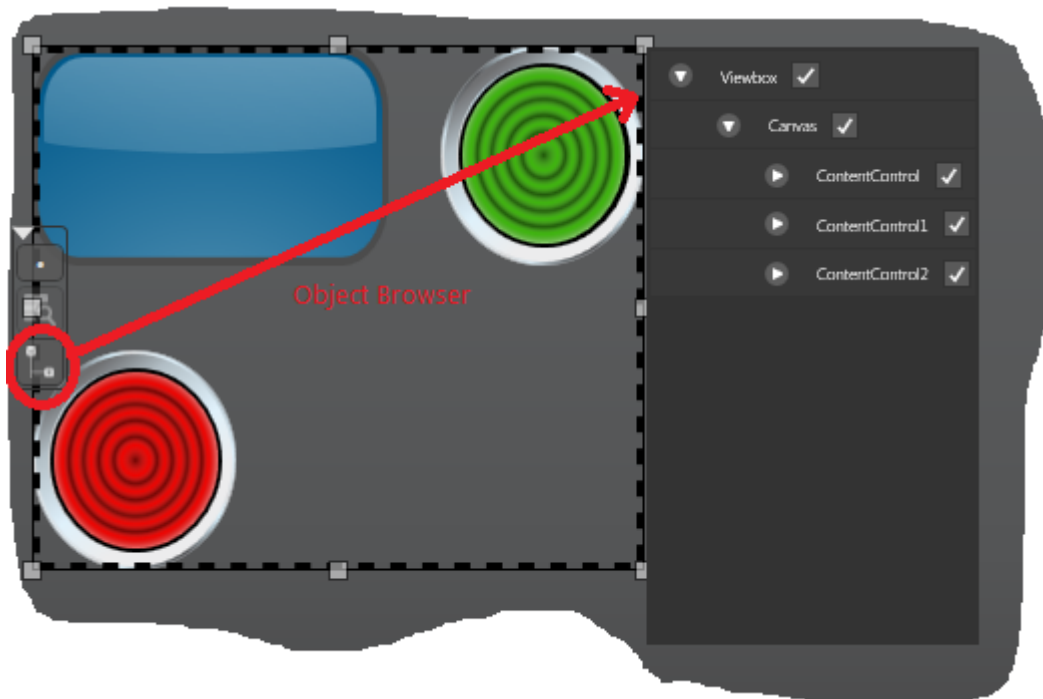
In addition to the "Group" command, which creates the Symbol keeping the same sizes and positions of the elements, there are other creation

commands such as "Vertical Stack Panel" and "Horizontal Stack Panel", which create a Symbol modifying the position and size of each element it is composed of to be disposed vertically or horizontally.

Once composed symbols can also be decomposed anytime afterwards using the "Ungroup" command and then recomposed back to its original formation using the "Re-group" command.

### Inspecting and selecting symbol components

Composed symbols can be inspected using commands predisposed for highlighting the components' structures and selecting each individual element, for example to modify the properties each Composed Symbol element without having to ungroup it to do so.



*The composed Symbol elements can be inspected using a predisposed window for this function or the tree structure from the Project Window.*

By using the command's small rapid activation button located on the symbol's border a menu will display showing the symbol's element structure as shown in the above image.

The selection of each item on the list will graphically highlight the select item making it possible for you to set its properties individually and independently from the symbol as a whole.

Symbol inspection can also be implemented from the screen's tree structure ("Elements" Group) in the Project Explorer Window.

## 5. Dynamic Animation Objects

### 5.1. Dynamic Animation Settings

The Movicon Object and Symbol Dynamic Animations, being the association of variable (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 no Tag has been defined for each animation, the **reference Tag** associated to the object will be used object will use (see paragraph on "Assigning Tags to Objects").

#### Animation Settings Window

The animation settings window can be accessed through the object's command Tooltip or by double clicking on the object. Once opened this window will be modal type. To configure an object's animations proceed as follows:

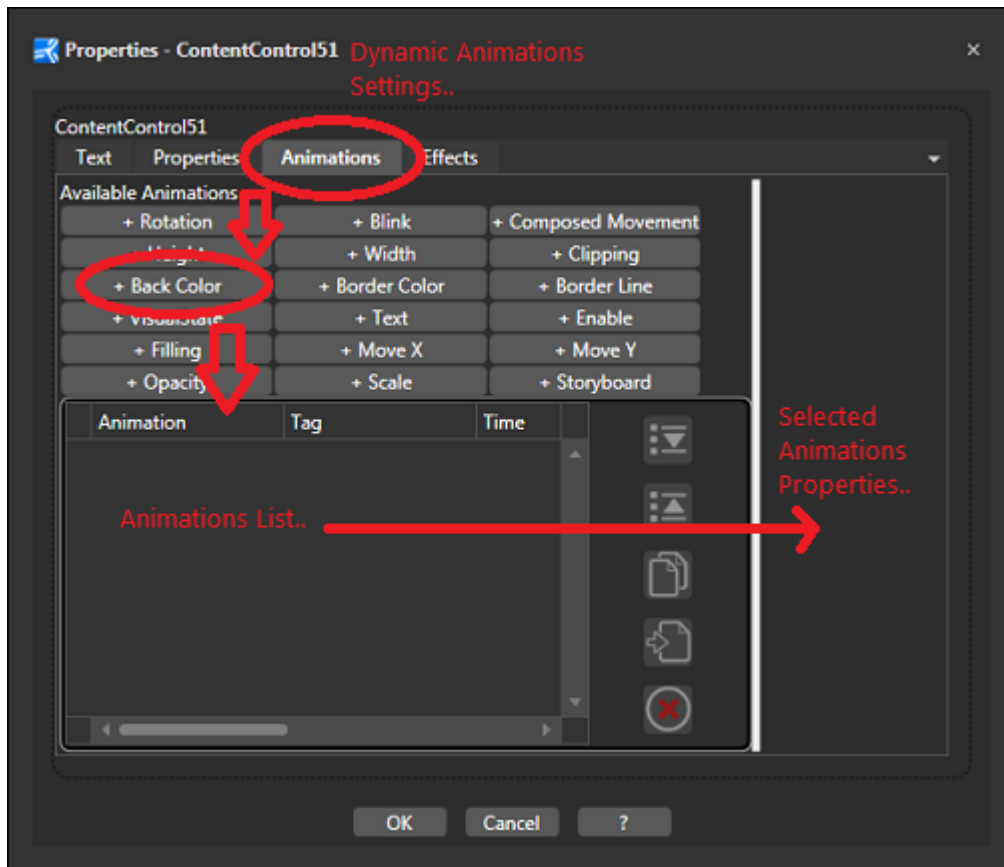
1. Open a screen and select the graphical object desired.
2. Activate the command Tooltip using the usual small button and then click on the "**Edit Common Properties**" item.



*The Animation Settings Window opens when clicking the "Common Property Editor" button from the object's Tooltip.*

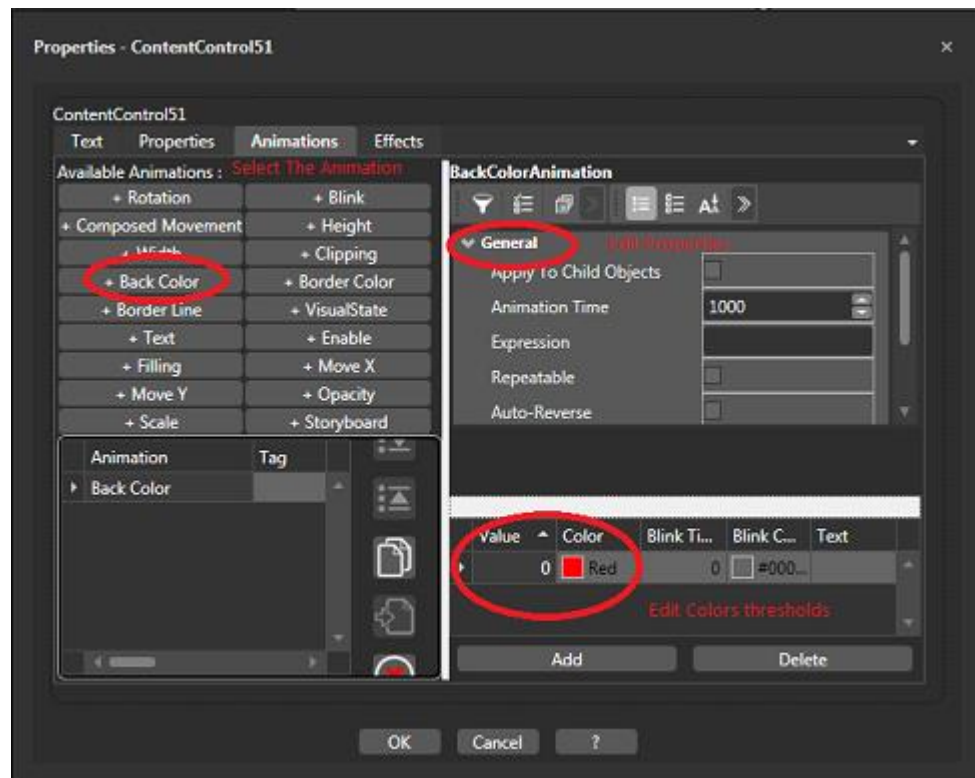
3. The "Common Property Editor" window has a selection of tags containing different types of settings. To access the animation settings, select the "**Animations**" Tab.

- 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

- After having added the Animation desired you can now proceed in configuring its parameters by double clicking on the row in 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 t animate the back color in different ways with different Tags.*



Window showing Animation parameters.

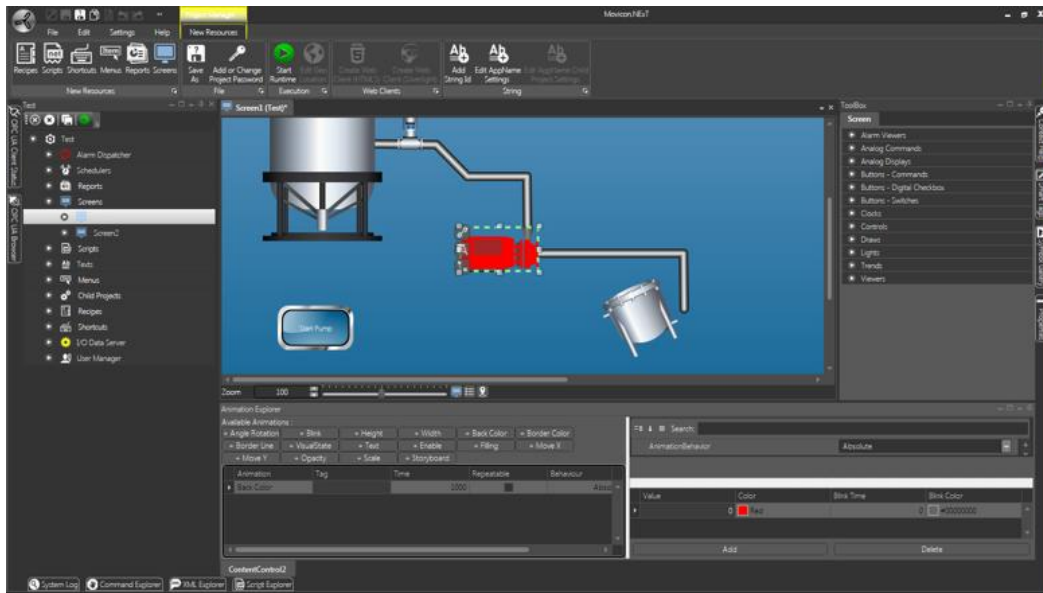
6. Some of the parameters of the various Animations remain the same while others change according to the Animation type. Once the Animation parameters have been configured, close the window with OK to confirm settings.

### Animation Explorer Window

The object's Dynamic Animations can also be edited 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, meaning that it automatically updates 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, meaning that it can be hidden, docked or positioned in foreground.



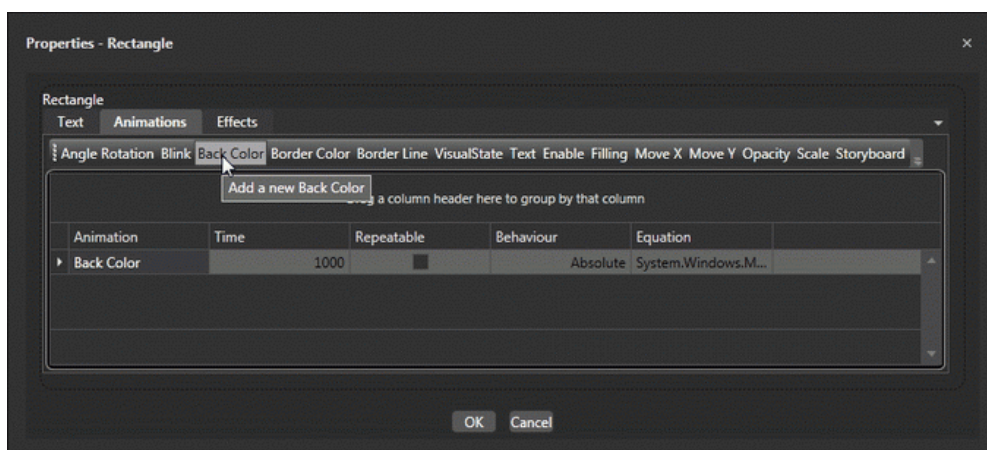


The "Animation Explorer" window is accessible from the Tabs located on the bottom border of the workspace.

## 5.2. How to give objects Color Animation

This lesson will give step by step guidance on how to set animation properties for colouring the object's background in function with the **"Variable1"** tag previously entered in the project's Address Space.

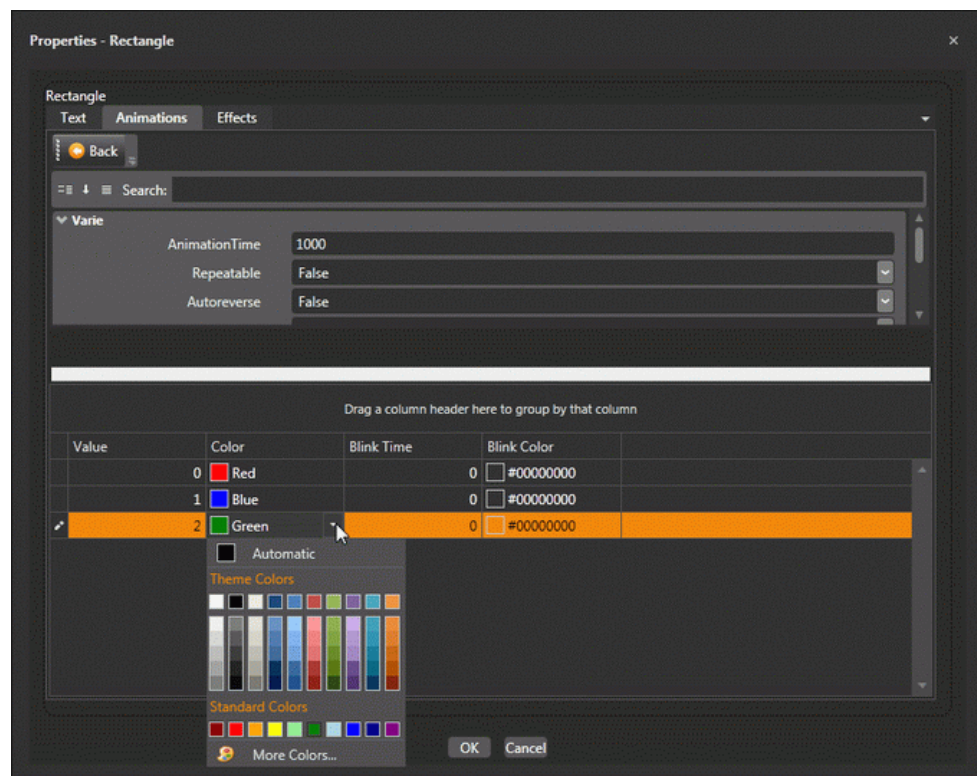
1. Open a screen and insert an object such as a Rectangle.
2. Select the object and open the command Tooltip then click on the "Common Property Editor" button.
3. In the dialog window that opens you will see a series of tabs which contain different types of settings. In our case, select the "Animations" Tab.
4. Click the "Back Color" item in the "Animations" window to add the animation to be managed. This animation item will be added to the list ready for configuring.



Animation Settings Window with the Back Color Animation added to the list.

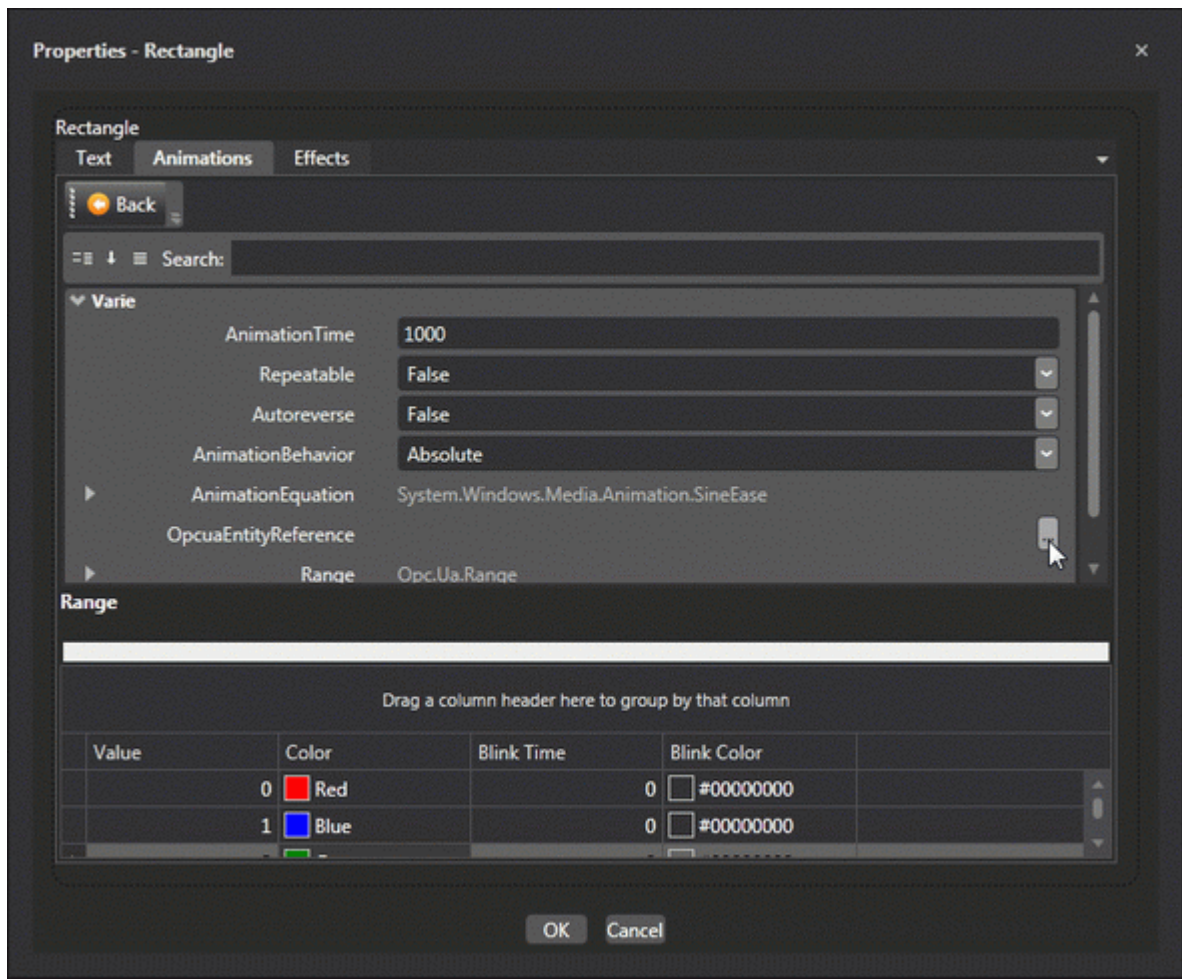


5. Double click the row on the list containing the "Back Color" animation to update window with its parameters.
6. The window displays divided into two parts. The top part contains the Animation's general properties and the bottom part shows a list to the inserted threshold colors. Only one threshold set to the "0" value has been inserted for default.
7. To add other color thresholds click at the bottom of the window showing the first threshold and then use the "INS" and "CANC" keys from the keyboard to add or delete other thresholds. In our case add two new thresholds.
8. After having added the new thresholds configure their activation values, the color to be used and blinking effect if need be. In this example set the three thresholds with the 0, 1, and 2 values and set them with the red, blue and green colors.



*Each threshold color can be changed by editing the table fields.*

9. Once you have finished defining the color thresholds you must assign the refer tag which is done directly in the Animation's properties using the "..." browse button in the "Tag" field to open the tag browser window to select the tag desired. In this case, select the "Variable1" tag which should have been created in the project's Address Space beforehand. This can be done at any time and before configuring the intervention thresholds.
10. Make sure that the "Variable1" tag's "Writable" property is enable so that the tag can be written in runtime.



*The reference tag is inserted in the Animation's Tag property*

11. If you do not specify a tag in the Animation's Tag field, Movicon will use the one associated to the object for managing the Animation (see paragraph on "Assigning Tags to Objects"). If no Tag has been associated to the object or inserted in the 'Tag' property, this animation will have no effect in runtime.
12. At this point you can now test run to see if the Animation works by starting the project up in runtime Starting UP Projects in Runtime and modifying the "Variable1" value. The object's colour should change according to which value, from 0 to 2, has been obtained.

## 5.3. More Dynamic Animation Examples

In addition to the back colours you can also apply other Animations to the Objects. Those available are:

<b>Angle</b>	Object's angle rotates in function with the Tag value
<b>Rotation</b>	
<b>Blink</b>	Blinking effect (visible yes, no, with gradual opacity)
<b>Back Color</b>	Back colour based on the Tag's threshold values
<b>Border Color</b>	Color of the border based on Tag's threshold values
<b>Border Line</b>	mobile line animation around the border of basic shape element.

<b>Visual State</b>	Enables control's "VisualState" visualization (pressed, normal, released, etc)
<b>Text</b>	Dynamic text within object in function with Tag value
<b>Enable</b>	Enables object's dynamics in function with Tag value
<b>Filling</b>	Object's dynamic filling in function with Tag value
<b>Move X</b>	Object's horizontal movement in function with Tag value
<b>Move Y</b>	Object's vertical movement in function with Tag value
<b>Opacity</b>	Object transparency (opacity) in function with Tag value
<b>Scale</b>	Object's scale in function with Tag value
<b>Storyboard</b>	Executed storyboards existing in XAML objects

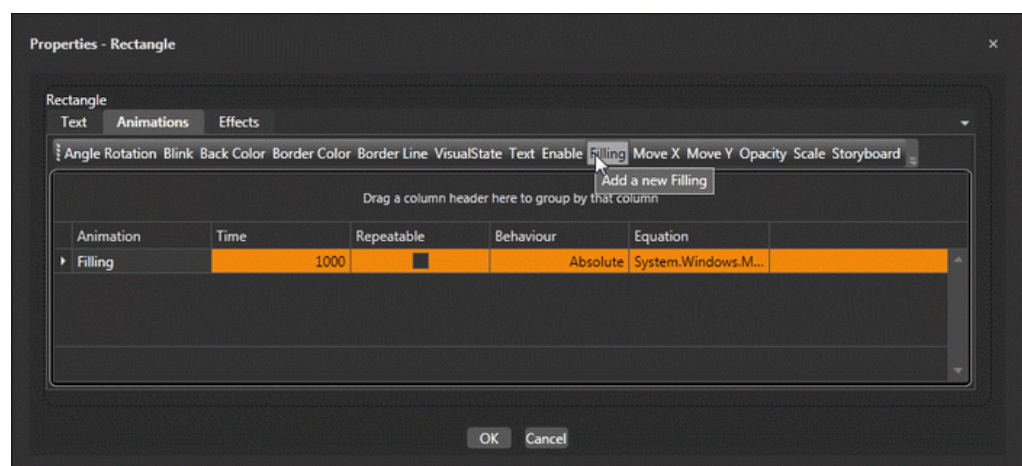
The various types of animation are set in the same way unless certain type of Animation have specific parameters of their own.

Below are described step by step procedures for setting a Rectangle object with two Animation types: Filling and Move X .

### "Filling" animation

To create the object's gradual filling animation proceed as follows:

1. Open a screen and insert the rectangle object.
2. Select the object to open the command Tooltip and click on the "Common Property Editor" button.
3. In the dialog window that opens you will see some tabs through which different types of settings can be defined. Select the "Animations" tab.
4. Click on the "Filling" item in the "Animations" window to add an animation to the list for configuring.

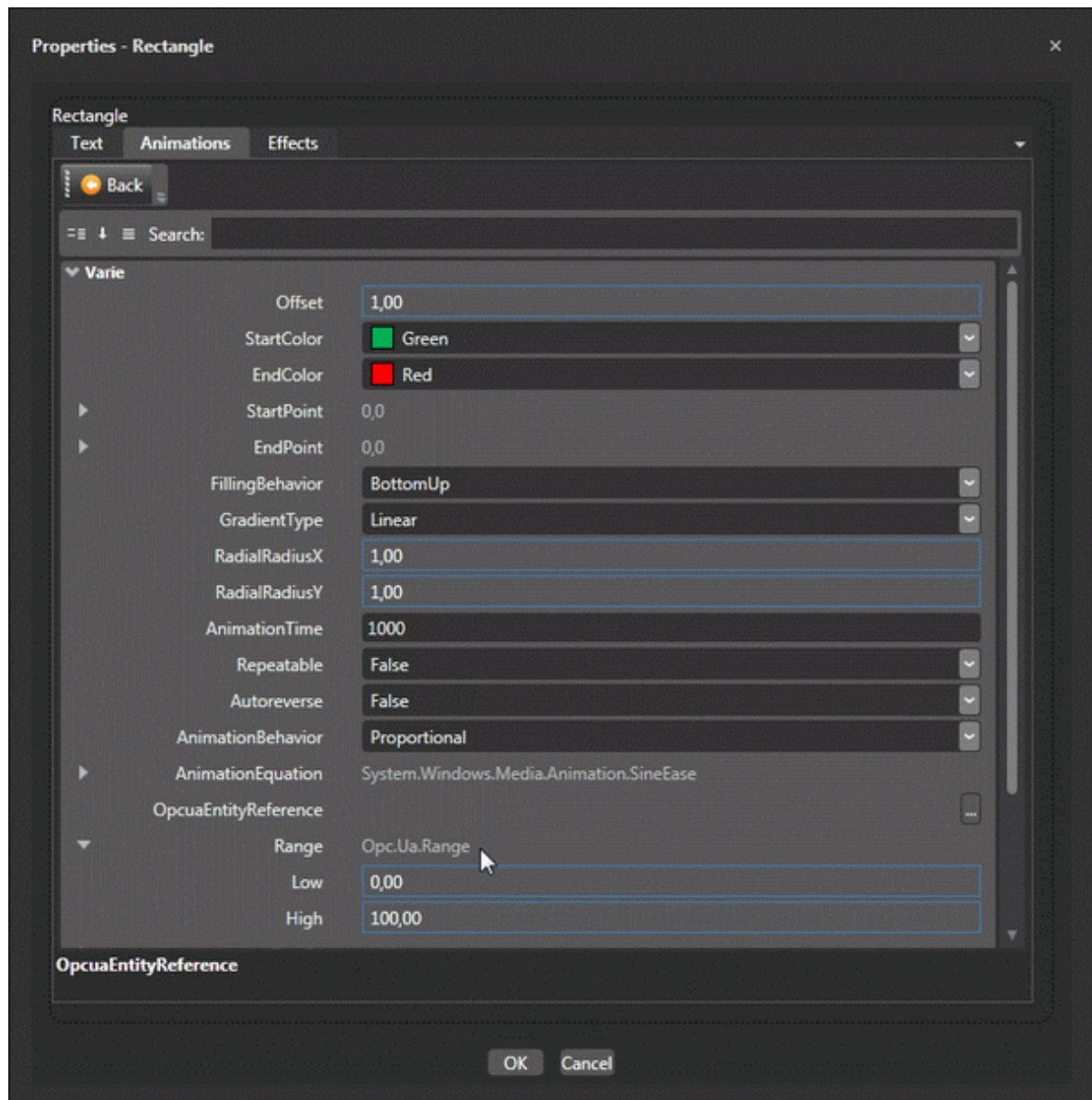


*Animation Setting Window showing the Filling animation added to the list.*

5. Double click on the list row showing the added Animation to update the window with the "Filling" animation settings.
6. This window can then be used to define the settings as required. For example, set the Range property with the value to manage the filling from '0' to '100' ("Range - Low" = 0, "Range - High" = 100).



7. Define the filling colors using the two "StartColor" and "EndColor" properties. If you set a different color in each one of these two properties, the rectangle's filling animation will gradually change from one color to the next.



*Animation Filling parameter setting Window.*

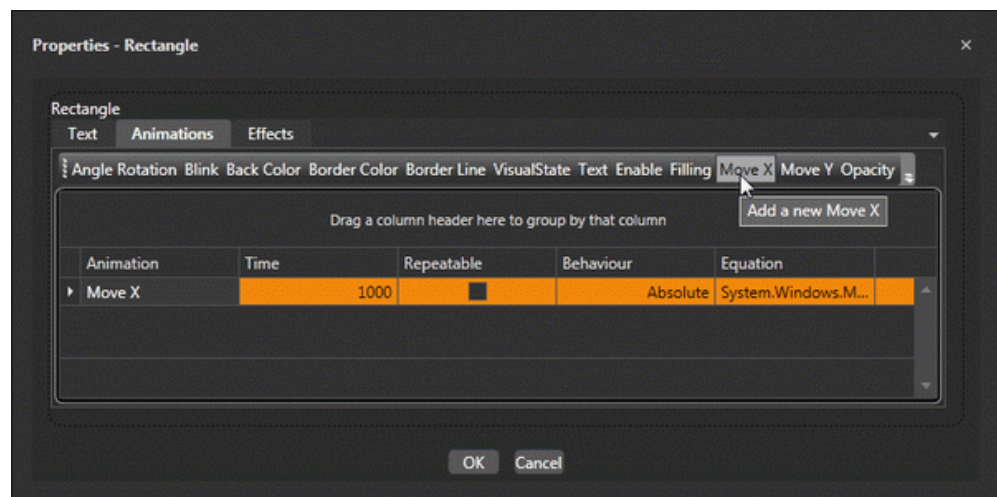
8. Once you have finished setting the animation parameters you must now assign a reference tag. To set this tag directly within the Animation properties use the "Tag" field. Click the "..." Browse button to the right to open the tag browser window and select the tag desired, e.g. "Variable1", that should have been previously created in the project's Address Space. Tags can be assigned at any moment even before configuring the Animation's parameters.
9. If you do not specify a tag in the Animation's "Tag" field, Movicon will use the one associated to the object for managing the Animation (see paragraph: Assigning Tags to Objects"). If no tag has been associated to the object and no tag has been inserted in the "Tag" property, the Animation will not be activated in Runtime.

10. Once you have completed the above steps check to see if the animation works by starting the project up in runtime and modify the value of the "Variable1". The object's filling should pass from 0% to 100% according to the value obtained.

### "Move X" Animation

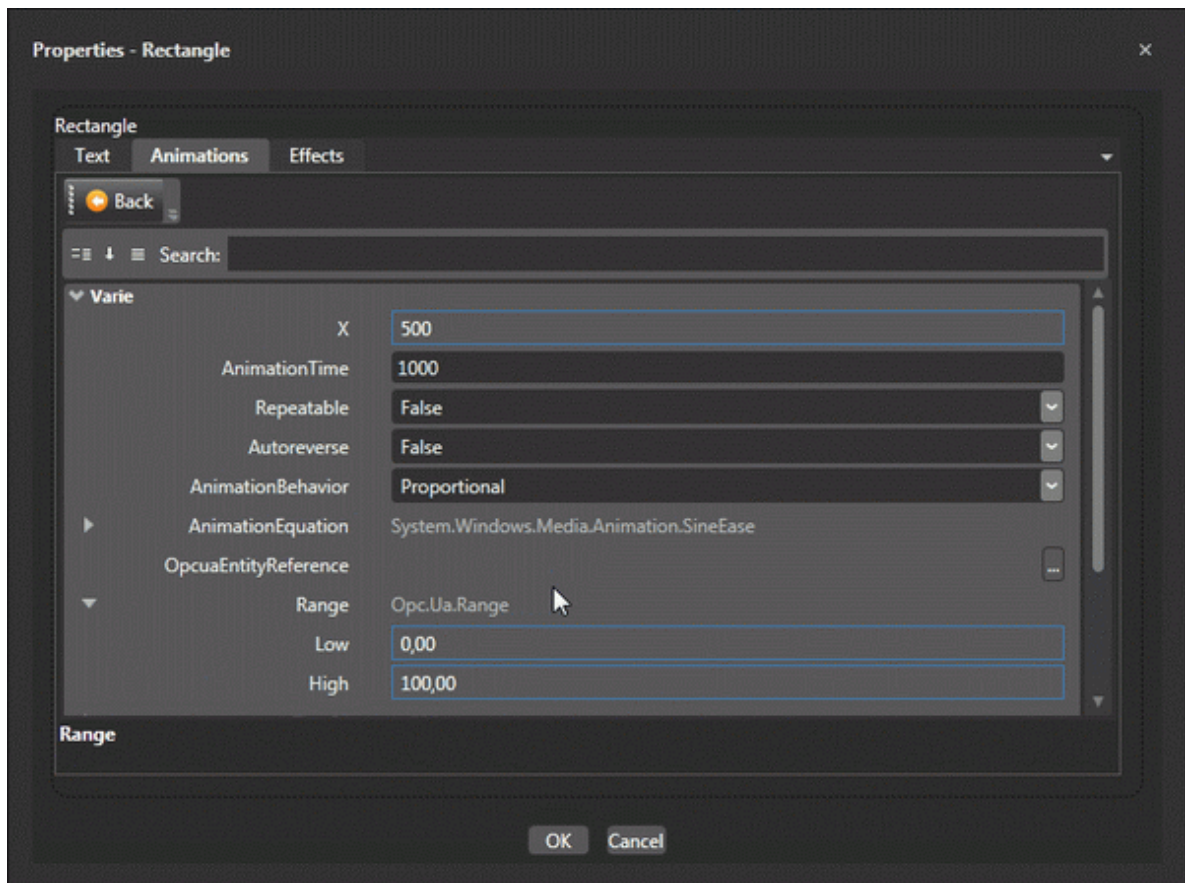
To create the object's horizontal movement proceed as follows:

1. Open a screen and insert the Rectangle object.
2. Selection the object to open the command Tooltip and click on the "Common Property Editor" button..
3. In the dialog window that opens you will see some tabs through which different types of settings can be defined. Select the "Animations" tab.
4. Click on the "Move X" item in the "Animations" window to add an animation to the list for configuring.



*Animation Settings Window with the Move X animation added to the list.*

5. Double click on the list row showing the added Animation to update the window with the "Move X" animation settings.
6. This window can now be used for defining the settings required. For example, enter 500 in the "X" property used to define the rectangle's maximum movement in pixels.
7. The "Low" and "High" fields in the "Range" group indicate the minimum and maximum values that the reference tag can obtain. The tag's maximum value corresponds to the rectangle's maximum movement. Insert "Range - Low" = 0 and "Range - High" = 100. In this way the tag's "0" value will correspond to the rectangle's initial position and the tag's 100 values will correspond the rectangle's movement of 500 pixels.



The Move X animation parameter setting window.

8. Once you have finished setting the animation parameters you must now assign a reference tag. To set this tag directly within the Animation properties use the "Tag" field. Click the "..." Browse button to the right to open the tag browser window and select the tag desired, e.g. "Variable1", that should have been previously created in the project's Address Space. Tags can be assigned at any moment even before configuring the Animation's parameters.
9. If you do not specify a tag in the Animation's "Tag" field, Movicon will use the one associated to the object for managing the Animation (see paragraph: Assigning Tags to Objects). If no tag has been associated to the object and no tag has been inserted in the "Tag" property, the Animation will not be activated in Runtime.
10. One you have completed the above steps check to see if the animation works by starting the project up in runtime and modify the value of the "Variable1". The object should move horizontally along 500 pixels starting from the initial position.

## 5.4. 3D Animation Graphics

Movicon.NExT uses XAML graphics with WPF technology which enables you to insert and use objects with 3D graphics. These objects can be animated by using individual 3D symbol components and varying the objects observation angle (camera).



Movicon.NExT has not yet been designed to edit or create 3D graphics scenes or models Movicon's job is to insert these objects on screens and provide

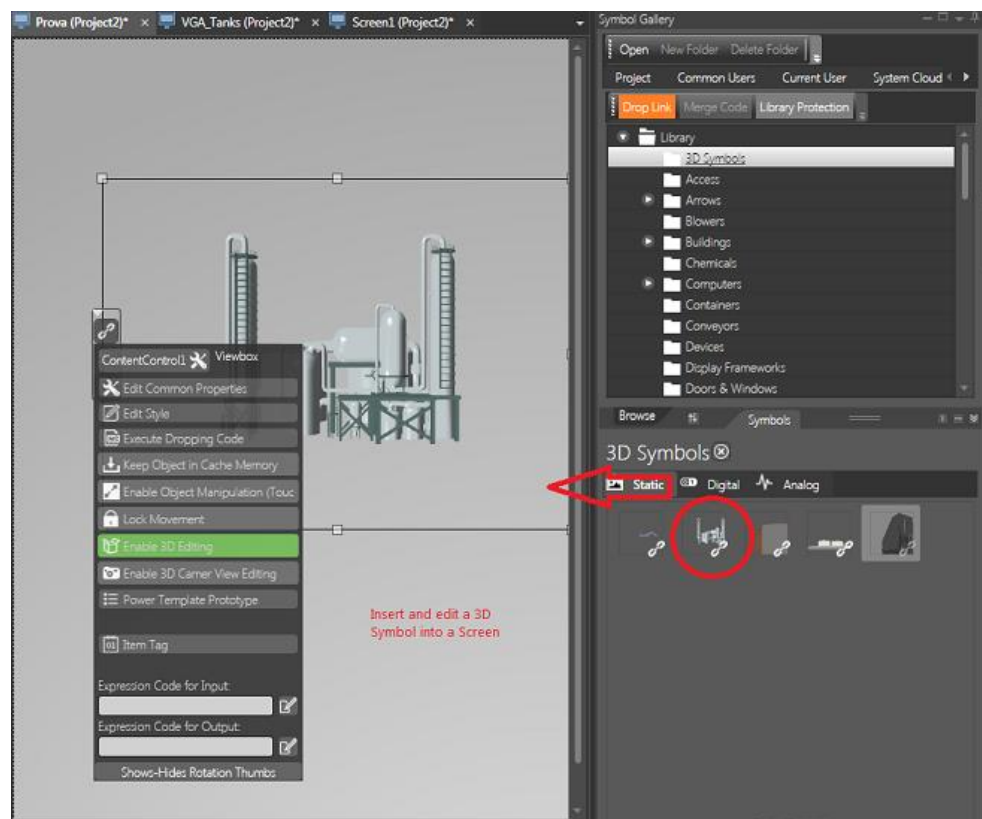
realtime graphical animation by connecting to the dynamic Tags to the animation type.

The Movicon symbol library contains some 3D graphical models, but the user should provide their own 3D models (generally created with Cad 3D or simulation software) to import and insert in screens as preferred.

The 3D format supported by Movicon is XAML and which defines 2D and 3D vector graphics.

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 mouser pointer on an area within screen and click to release and insert object. Size and position the object as pleased.
4. Select the object and access the commands menus 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*

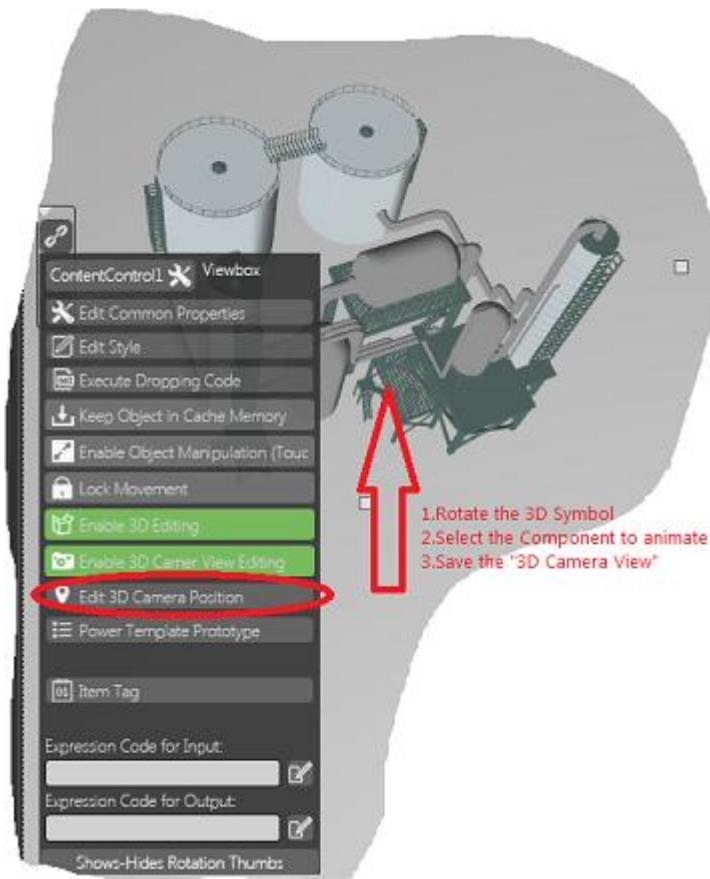
5. Note that when clicking the various 3D symbol components, the selected symbol's parts will highlight with a colour and throbbing effect to show which parts of the 3D symbol



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.

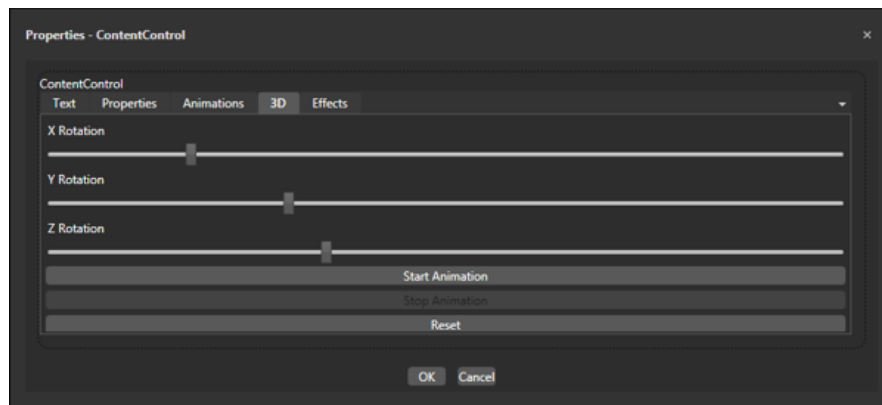
6. The 3D object can also be rotated also while being edited for either adding any other components to it or for recording different events viewed (different 3D camera scene positions), which can then be set in runtime by the user with a right mouse key click.
  - To use the rotation and recording functions of eventual views, use the "**Enable 3D Camera View**" and "**Edit 3D Camera Position**" commands made available after activating the "**3D Edit**" item described previously.



*When selecting a symbol component it will animate with a throbbing effect.*

- In addition when the whole object has been selected (and not just one of its components) you can use the X, Y, Z rotation cursors from the Common Property Editor's "3D" tab's settings to rotate the object or perform a simple rotation simulation.

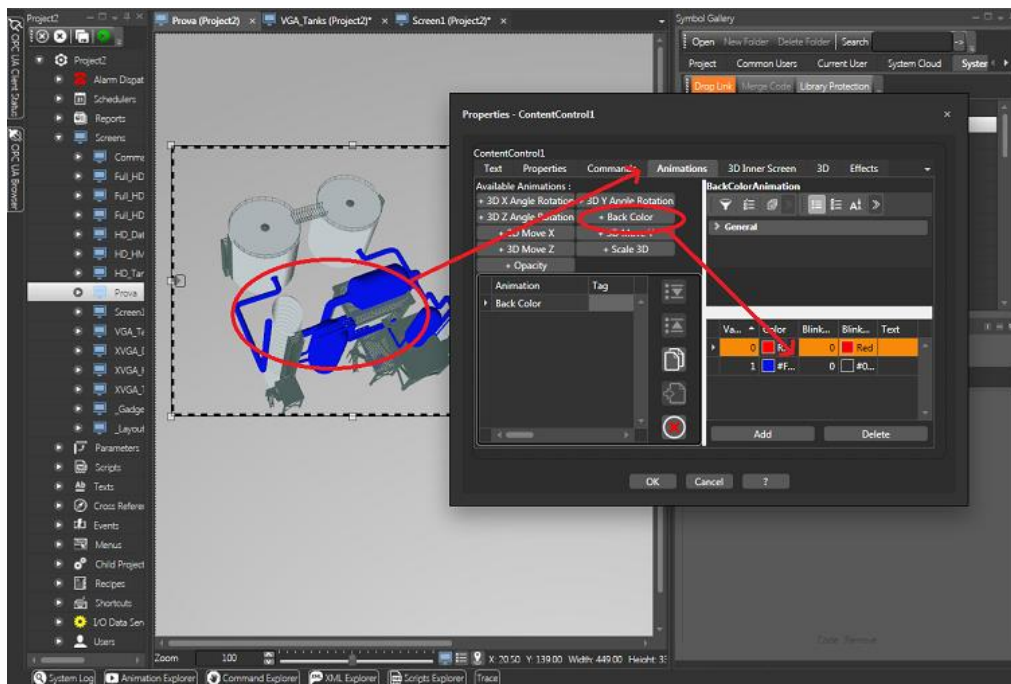




6. After having selected the 3D component model desired for animating, use the object's command menu again and use the "**Common Property Editor**" command.
7. Select the "**Animation**" tab from the dialog window that opens to access the **graphical animation commands** available for the individual components and component groups that compose the the 3D model and which are listed in the table below:

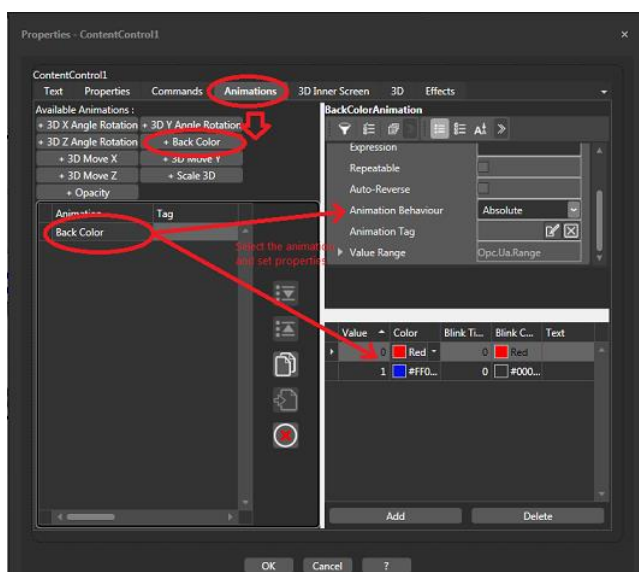
<b>3D X Angle Rotation</b>	Object's angle rotation on X axis in function with tag value
<b>3D Y Angle Rotation</b>	Object's angle rotation on Y axis in function with tag value
<b>3D Z Angle Rotation</b>	Object's angle rotation on Z axis in function with tag value
<b>Back Color</b>	Back Color based on tag threshold values
<b>3D Move X</b>	Object's linear movement on the X axis in function with tag value
<b>3D Move Y</b>	Object's linear movement on the Y axis in function with tag value
<b>3D Move Z</b>	Object's linear movement on the Z axis in function with tag value
<b>Scale 3D</b>	Object's scale sizing in function with tag value
<b>Opacity</b>	Transparency and visibility value in function with tag value

6. In our example we will use 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 will select 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 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 test run this Animation to see if it 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.

## 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.





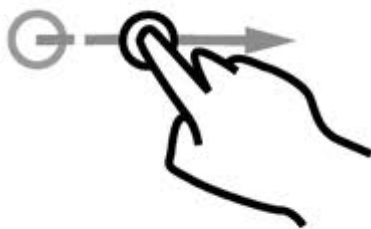
*This example shows how a screen can become an Inner Screen 3D model component. The screen functions as a 3D Inner Screen positioned and fully interactive within the model.*

## 5.5. Using multitouch

The modern user interface systems can be equipped with touch screens with multitouch support, based on experience using smartphone or tablet systems, offering new possibilities of industrial HMI system usages which are more intuitive and a pleasure to use.

The Movicon.NExT user interface and is predisposed natively for Multitouch use. This Tutorial quickly shows you which are the concepts and the possibilities used in design engineering Movicon screens.

### Swipe



From HMI system user interface experience the "swipe" has proven to be one of the most intuitive and natural commands to use. It makes page navigation much more intuitive and smoother than those techniques used in conventional systems.

By placing your fingertip on the screen and sliding it towards the left or right enables you to change pages by scrolling from one page to the next or previous with a simple touch. You can instantly return to the main page by tapping the hidden system bar if not provided otherwise.

The main page supports the navigation concepts based on touch or scrolling according to how the project is set in its main settings (as described in the "Screen Navigation" section).

## Pinch & Zoom, Pan



The Pinch & Zoom command is used with two fingertips on the screen to produce open and close gestures. This command produces Zoom In or Zoom Out effect to enlarge or reduce screen.

The Zoom effect usually enables you to scroll within the panned area or visualize design objects and elements that are not made visible by normal displays considering that all object have display properties which function according to the zoom value given to the screen.

## Simultaneous Multitouch Support

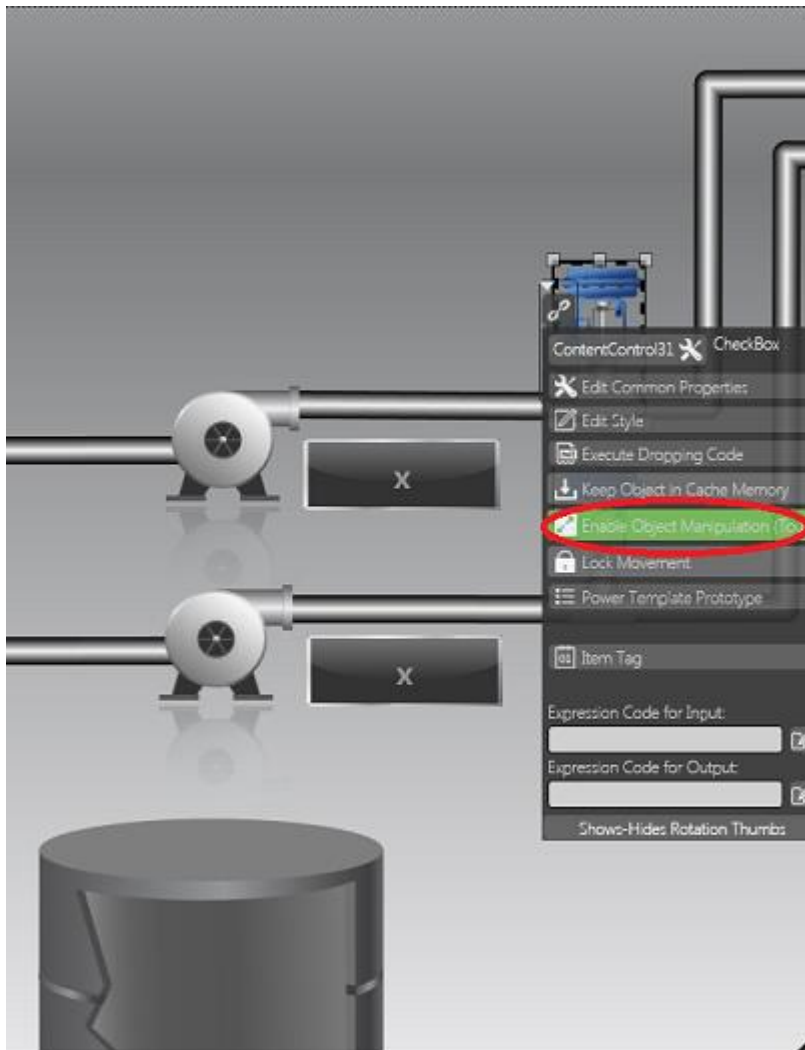


A very useful feature to have in the industrial ambient is the possibility to execute commands conditioned to respond to the touch of two graphical command objects simultaneously. For example, for the operator's safety it might be required that an operating command, used for starting up dangerous machinery movement, be invoked by making the operator use both hands to press two buttons, placed at opposite sides of the screen, simultaneously. on the screen.

### 5.5.1. 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 simple select the object desired, active 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:

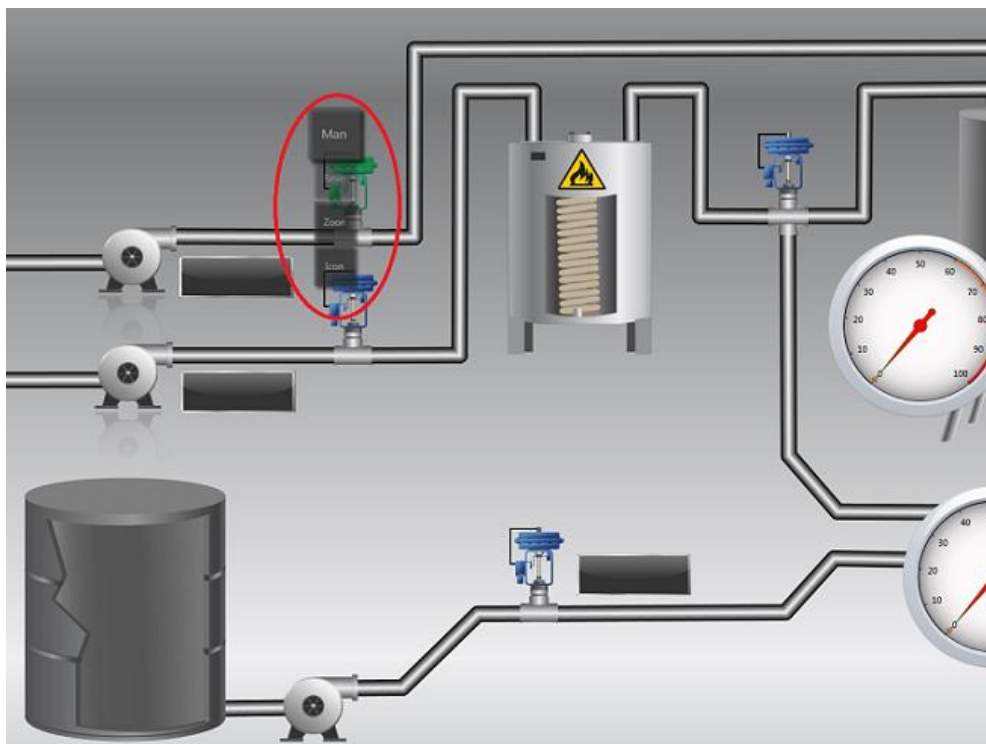


*This example shows how to enable the touch manipulation of graphical Movicon objects in design mode.*

**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 as refers to each single object.



*This image shows an example of a Gauge object enabled with 'touch' manipulation during Movicon runtime using buttons.*

To **restore position**, as defined in design mode, simply 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.*

## 5.6. Using Display Controls

Different objects dedicated to displaying Tag values and their modification can be inserted in the screen user interface. The Toolbox offers a vast range of objects to choose from that have been designed to display or represent dynamic Tag values. All the objects have been pre-configured and their styles, with which values are represented, can be modified as pleased.

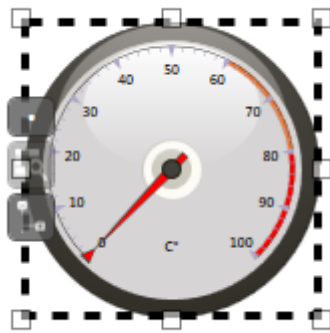
Those objects committed to displaying Tag values are mainly those belonging to the Toolbox's "**Displays**" category.



The different types of data viewer objects in the "Displays" groups have been designed for displaying values in numeric format or other display representations such as graphics with scales or indicators.

For example, to insert a Gauge (indicator or needle) on a screen proceed as follows:

1. Open a project screen.
2. Open the Toolbox and go to the "Displays" group and click to select the "Gauges" group.
3. Select a Gauge type
4. Position the mouse pointer on an area of the screen where you wish to insert the object and click to release selected object.



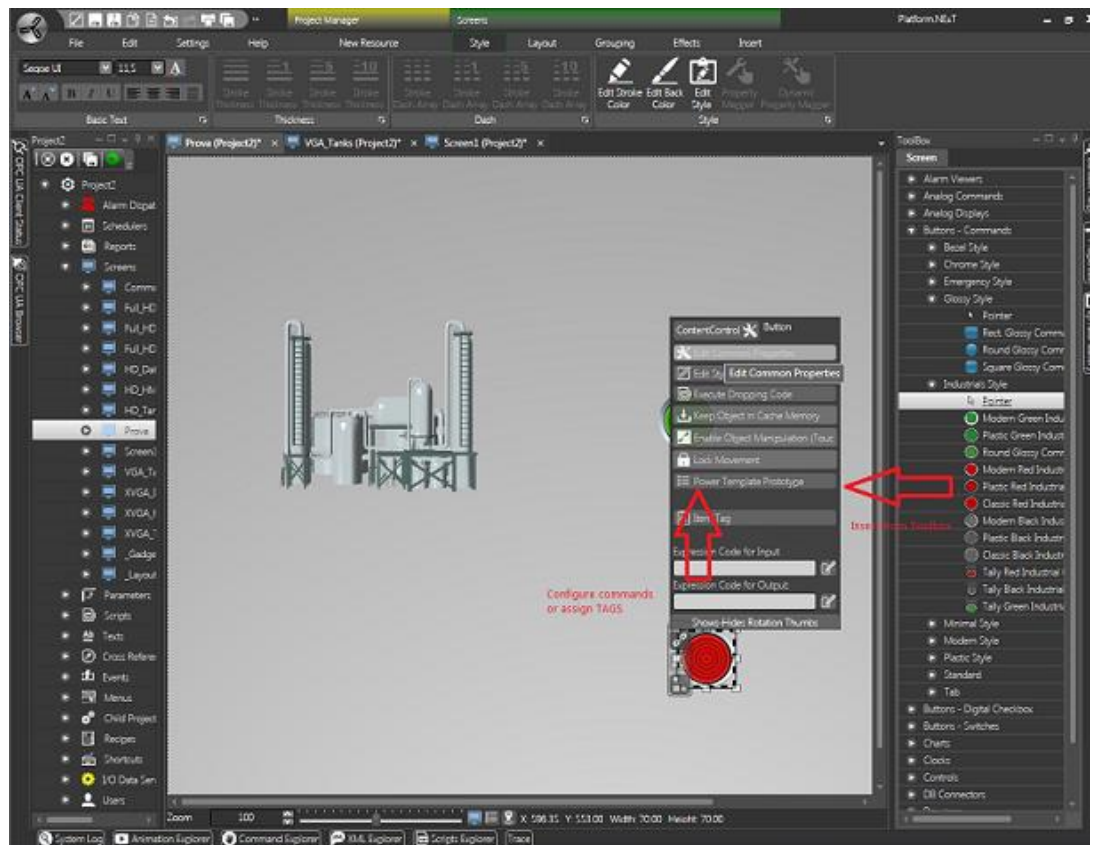
6. After having inserted the object on screen you can now proceed with modifying its styles using the **Properties Window**. Besides the different styles you can also customize colours, gradients, fonts, scales or indicator needle.
7. After having modified the graphical styles, you can start assigning which Tag the object is to display by following the procedures set out in the paragraph on "Assigning Tags To Objects" such as dragging the desired Tag and dropping it on the object or selecting the reference Tag from the object's tooltip menu.
8. After you have completed the necessary procedures launch the project in runtime Starting UP Project in Runtime to verify that if functions correctly by modifying the Tag value to see if the display object's indicator needle points to it.

## 5.7. Using Command Objects

Different object types dedicated to executing commands can be inserted on the screen user interface with the possibility to modify Tag values. The Toolbox provides a vast range of different objects to choose from and which are dedicated to executing command or determining Tag values.

All these objects are pre-configured with styles that can be modified as pleased to represent values.





## The different command objects:

Objects dedicated to executing Commands can be inserted on the screen user interface to use for changing page, setting values in tags, executing scripts or changing project language along with many other possibilities. You will also find objects dedicated to act on tag values, such as the ON-OFF button (checkbox), or meters.

### Objects that operate on Tags

This objects are mainly those belonging to the "Analog Controls" or "Button - Digital Checkbox" Toolbox categories.

These types of objects have been purposely designed to act on tags, therefore after having inserted and customized the object's styles a reference tag must be assigned so that it can act on the tag's value according to user operations performed in runtime.

### Command execution objects

These objects are mainly those belonging to the "Button - Commands" Toolbox category.

Specific commands or lists containing various commands can be associated to these objects using a predisposed configuration window designed for this purpose.



It is important to note that even though each object can be completely configured in style, its mechanics will remain those for the purpose it was designed for.

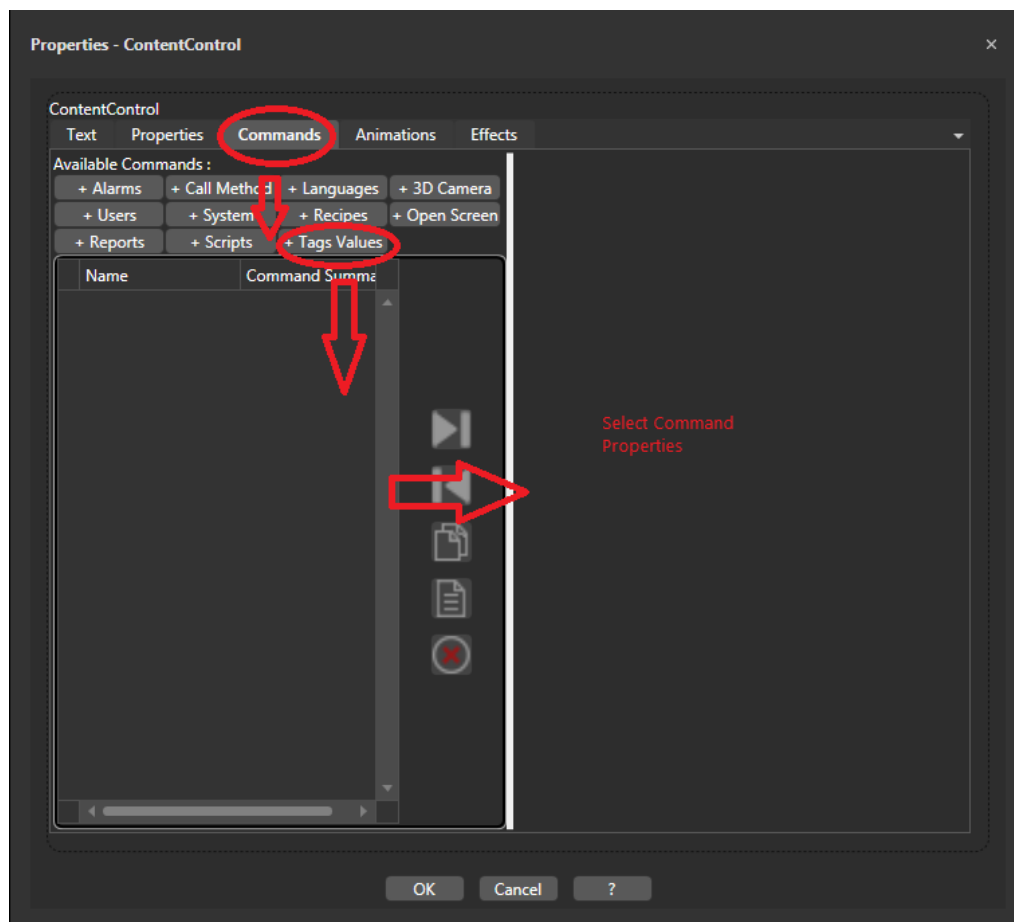
In other words, it is possible to insert a "checkbox", in the form of a button, and then change its style as pleased by associating it with any graphical symbol from the symbol libraries. However, the object strictly functions as a "checkbox" and cannot function as a command object.

## Assigning commands

To configure the assignment of one or more commands to an object you will initially need to select and insert a command object from the Toolbox. Please note that project commands can only be assigned to the category of objects whose mechanics have been purposely designed for command settings. Therefore, select the objects belonging to the Toolbox "**Buttons - Commands**" group.

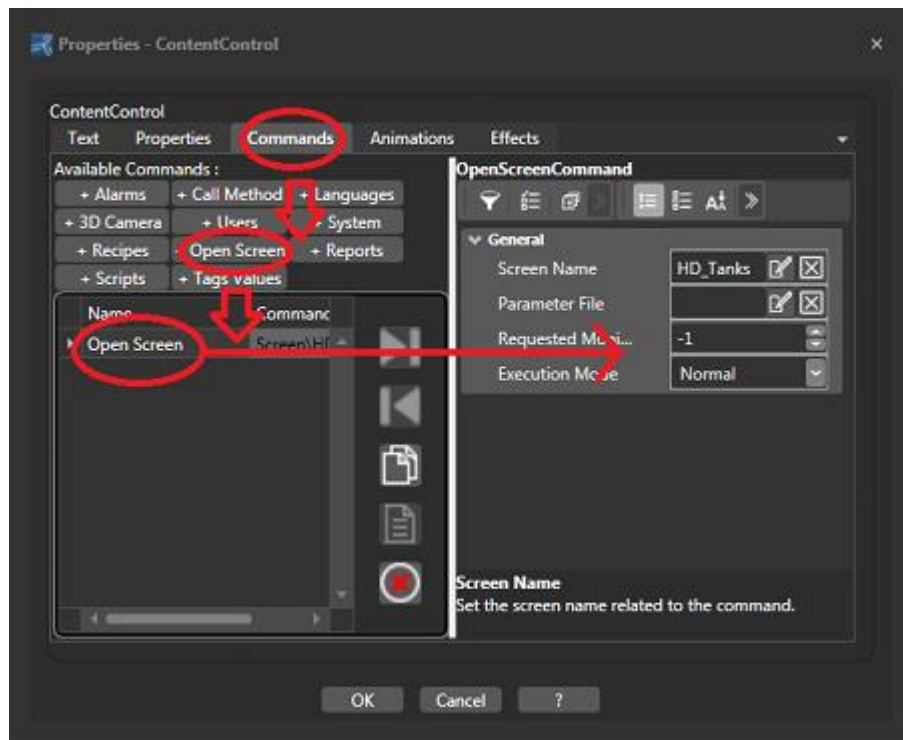
After having inserted one of these command objects, you will be able to activate the "Common Property Editor" settings window which you have already seen for configuring dynamic animations.

Among the tabs shown in the window you will find the Commands tab, as shown below, containing the **Commands** available for inserting in the object's command list.



After having added the desired command using the buttons at the top of the window, you can now proceed with setting the relative parameters by double clicking on a row in the list to update the window with the selected command's parameters.

*For example, if you wish to create a change page command, you will need to select the "Open Screen" command from the list of commands as indicated in the figure below. Please note that the same command can be inserted numerous times whereby a list can be created to execute the commands in sequence.*

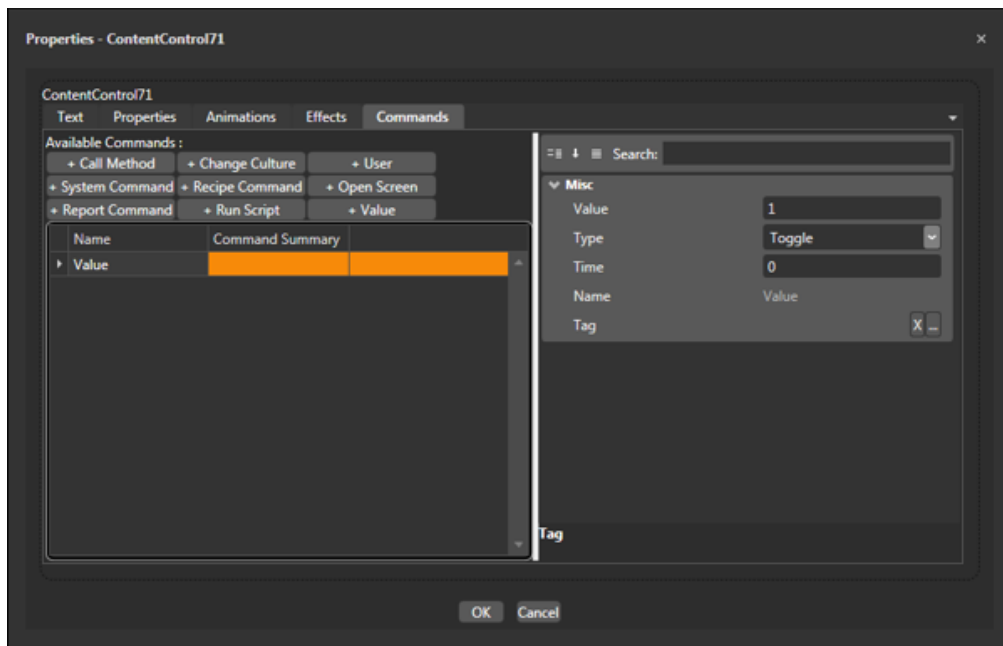


#### Example:

The procedure below explains how to associate two different command to two Buttons. The first command is a "Set" tag command and the second is a "Change Page" command.

#### To insert the "Set" command of a tag proceed as follows:

1. Open a project screen.
2. Open the Toolbox and click to select an object from the "**Buttons - Commands**" group.
3. Position the mouser pointer where desired on the screen and click to release the object in that position.
4. Select the object and open the command Tooltip then click on the "**Common Property Editor**" button.
5. In the dialog window that shows you will see some tabs where different types of settings can be defined. Select the "**Commands**" tab which is the one we need to use.
6. Click on the "**Value**" item in the "Commands" window to add the command we need to work with. The command will then be added to the list ready for configuring.
7. By selecting the added "Value" command from the list, the window will update with the "Value" command's parameters enabling you to define them as desired. For example, if you set the "Value" field with the "1" value and the "Type" field with the "Toggle" action, every time the Button is pressed the tag associated to the command will change from the "0" to the "1" value and viceversa.
8. Go to the Tag item and select the variable with which the button will act on.



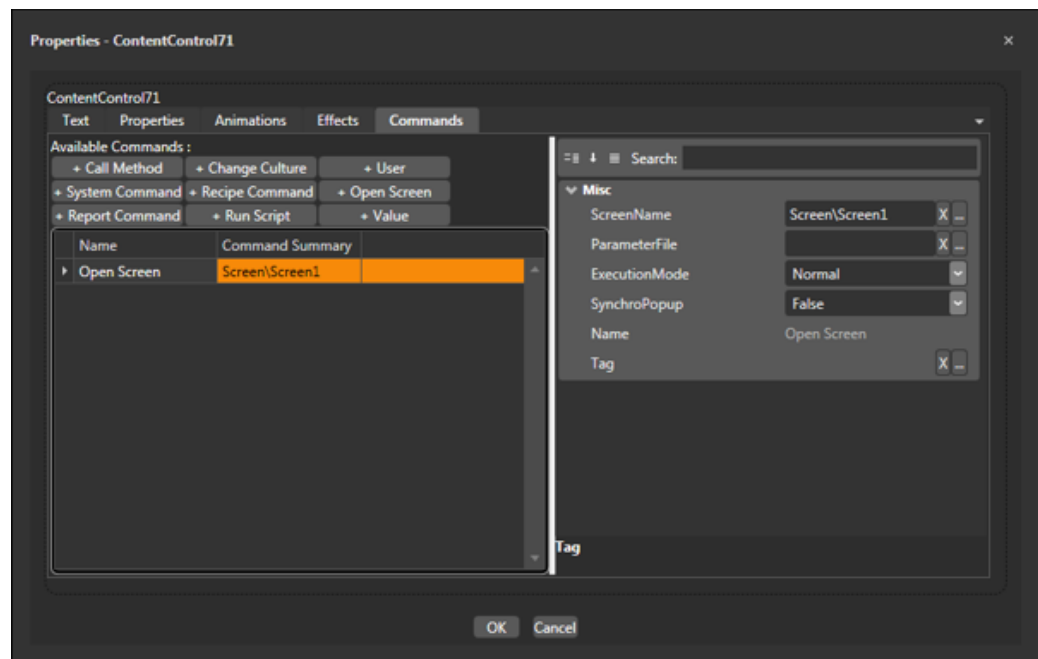
Window containing the Value command's parameters ready for configuring.

9. If no variable is specified in the Command's "Tag" field, Movicon will use the one associated to the object for managing the Command (see paragraph on Assigning Tags to Objects). If no variable has been associated to the object and no variable has been inserted in the "Tag" property, the Command will have no effect in Runtime.
10. Now check whether the Command works by launching the project into runtime Starting UP Project in Runtime and clicking on the button. The "Variable1" value should set to "1". This action can be viewed better if you insert an "EditDisplay" object to show the "Variable1" when set by the Button.

#### To insert the "Change Page" command proceed as follows:

1. Open a project screen or use the one you already opened beforehand.
2. Open the Toolbox and select by clicking an object from the **"Buttons - Commands"** group.
3. Position the mouser pointer where desired on the screen and click to release the object in that position.
4. Select the object and open the command Tooltip then click on the **"Common Property Editor"** button.
5. In the dialog window that shows you will see some tabs where different types of settings can be defined. Select the **"Commands"** tab which is the one we need to use.
6. Click on the **"Open Screen"** item in the "Commands" window to add the command we need to work with. The command will then be added to the list ready for configuring.
7. By selecting on the command added to the list the window will update with the "Open Screen" command's parameters.
8. You can now go ahead and define the settings desired using this window. For example, set the "ScreenName" field by selecting the screen name you wish to open, "Screen1",

and select the "Normal" action in the "ExecutionMode" field. This will determine the opening of "Screen1" in "Normal" mode when pressing the Button.



Command Settings Window with the Open Screen command added to the list.

9. Now check whether the Command works by launching the project into runtime Starting UP Project in Runtime and clicking on the button. The "Screen1" screen should load.

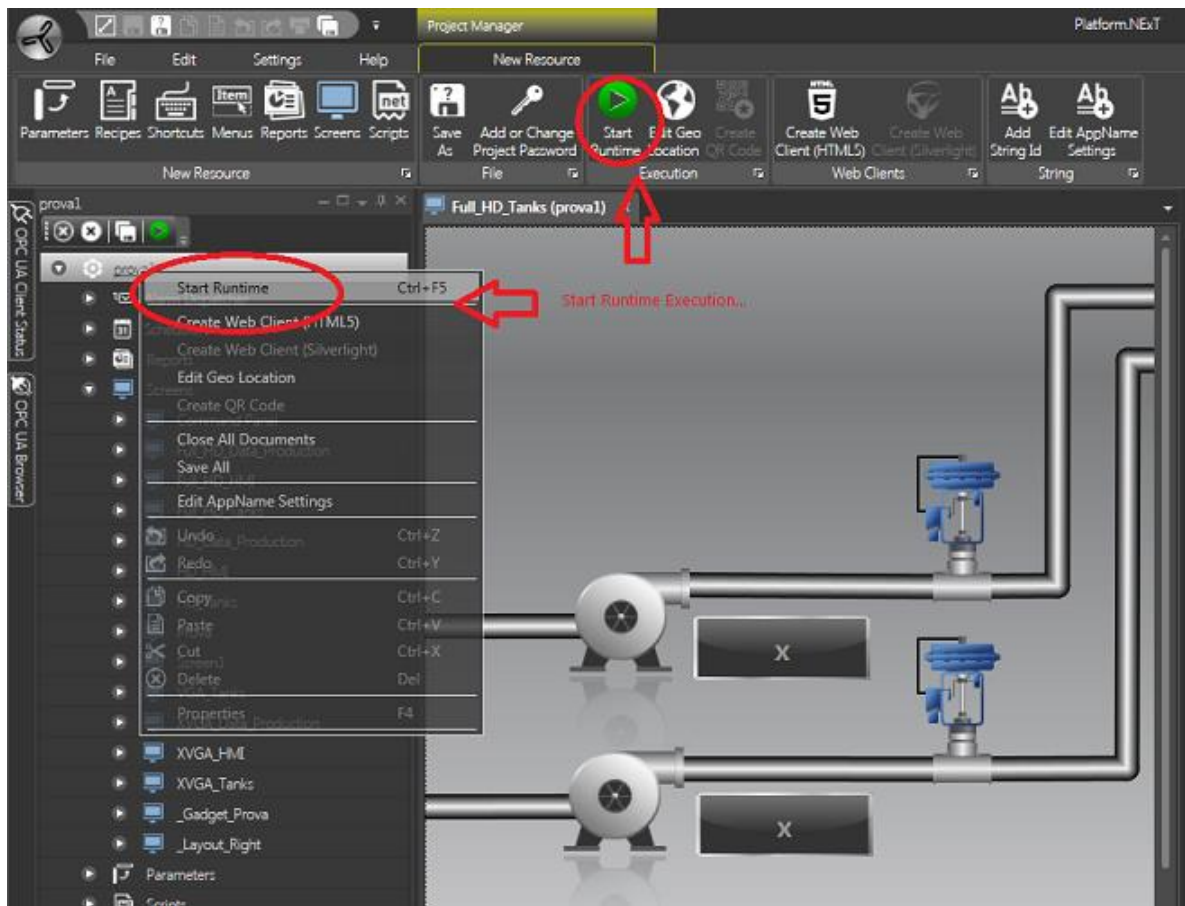
## 5.8. Starting Up Projects in Runtime

The Platform.NExT platform project is fundamentally composed one part Server, designated to perform communication tasks and manage data, and one part Client which is mainly composed of the Movicon.NExT for project's HMI graphical interface.



both the platform's Server and Client components are independent and therefore can be started up independently in runtime.

The whole project can be simply be started up in runtime directly from the platform's workspace using the **"Start Runtime"** command available from the workspace's main "Project Manager - New Resources" ribbon. The startup command can also be made available by right clicking to open the project's contextual menu in the Project Explorer Window.



*The Start Runtime command starts Project up in runtime*

When the "Start Runtime" command is activated Movicon will check whether the Server module has been started up. If this is the case the Client module will then be started up otherwise both Server and Client module will be started up together.

### Start Project from command line

The following command can be used to startup a Project directly in Runtime without opening development/ design mode beforehand:

```
"C:\Program Files\Progea\Movicon.NExT\MoviconNExT.exe"
E:\Projects\Test01\Test01.ufproject -start
```

where:

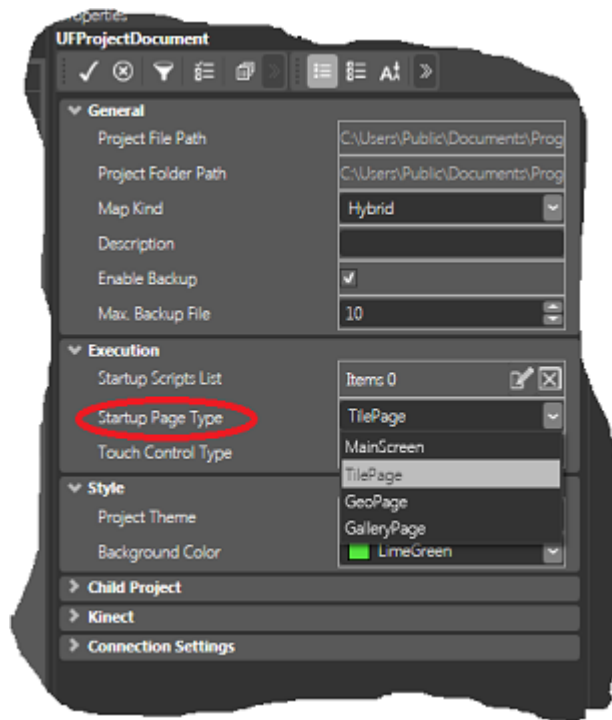
<E:\Projects\Test01\Test01.ufproject -start> represents an example of a PlatformNExT project path and position.

This command line will start up both the Project's Server and Client modules. It can also be created as a shortcut or defined in the Windows automatic startup.

### Project Startup Type

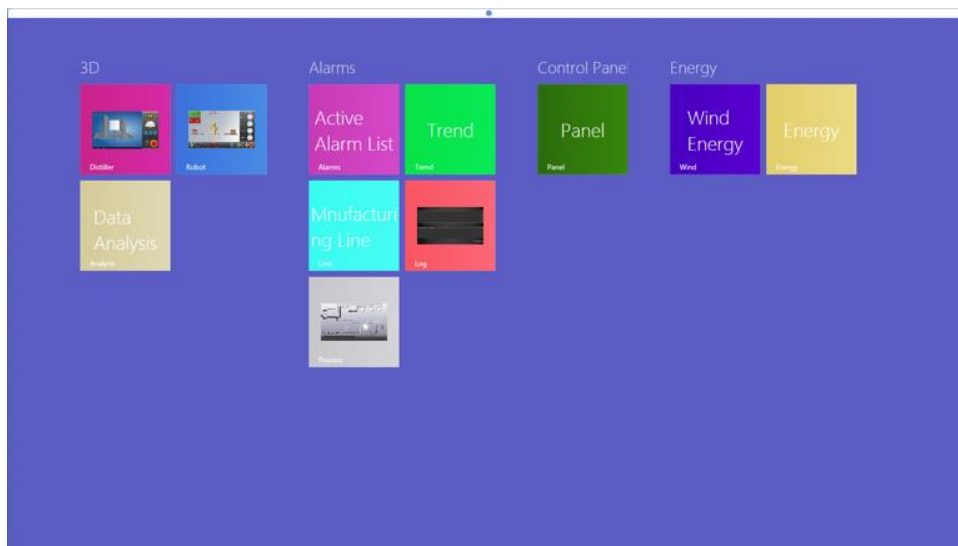
By using the project's **"StartType"** property you can select the project startup window display type which will show when activating Runtime (Start Runtime command). The available modes are:

- **Main Screen:** the startup page will be one of the screens created and inserted in the project. The screen to be used as the startup screen can be selected from the "MainScreen" property. If this property is left empty, Movicon will search for a screen named "Main". If this screen is found it will be started up otherwise the first project screen in alphabetical order will be started up.
- **Tile Page:** The start up page will automatically be shown with the typical Windows 8 tile interface. It will be composed of with same number of tiles as there are project screens. Each tile will show the name of the screen and its preview. A click of the mouse will open the corresponding screen.
- **Gallery Page:** The startup page is automatically built with the Gallery interface with a carousel preview of the project screens. The preview of screens can be scrolled back and forth until the desired screen is found. Once found simply click to open it.
- **Geo Page:** The startup screen will be a geographic map in which the locations of project's resources are highlighted. In this case the project is usually composed of one or more child project located in different geographical areas. The click on one of these hot regions will open the corresponding project.



*The project startup type is defined in its "StartType" property*

Project are created with the "TilePage" setting for default. For example when starting up a project in Runtime in this mode you will get a startup page like the one shown below. The tiles can be positioned and grouped in persistent mode as pleased and also in function with folders in which the project screens have been placed.



*Project start runtime page in "TilePage" mode*

## Stop Project Runtime

The "ALT+F4" system keys can be used for stopping project runtime after project has been started up.

It is also possible to predispose useful command objects or menus in the project to startup and stop project runtime according to the commands provided in the object command list.

You can also predispose a close window command in the title bar of the corresponding window display according to Windows standards. However this is only possible if specified in the screen "WindowState" property.

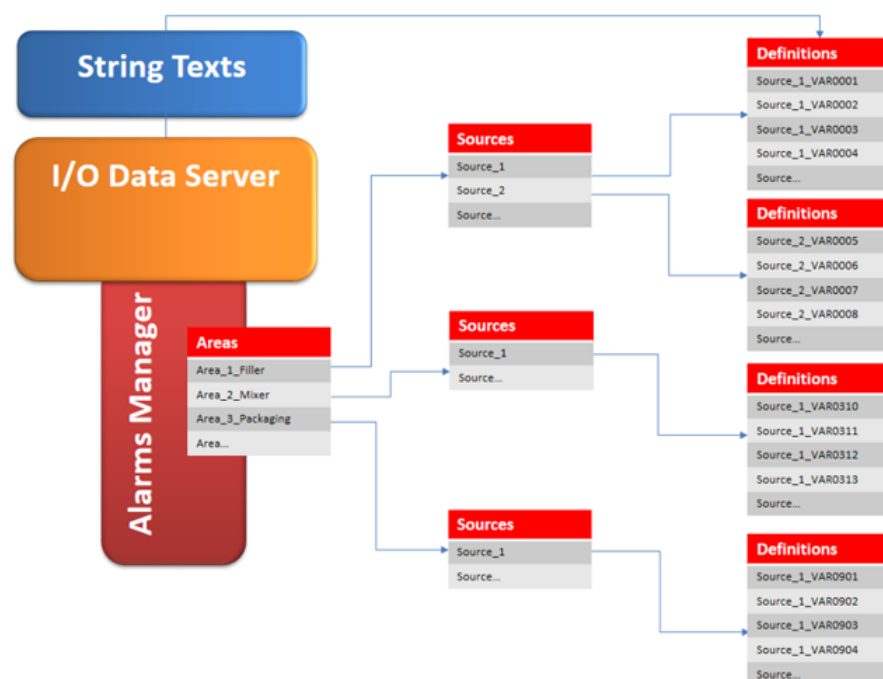


## 6. Alarm Management

### 6.1. Inserting Alarms

The Platform.NExT Alarm management provides the use of the **Alarms Server module**, a component of the **I/O Data Server**.

The Alarm Server module enables you to defined and manage the alarms based on the Prototype concepts that can be defined in the I/O Data Server "**Alarm Prototypes**" form. Look at the diagram below to get a better idea of the Alarm concept in Platform.NExT:

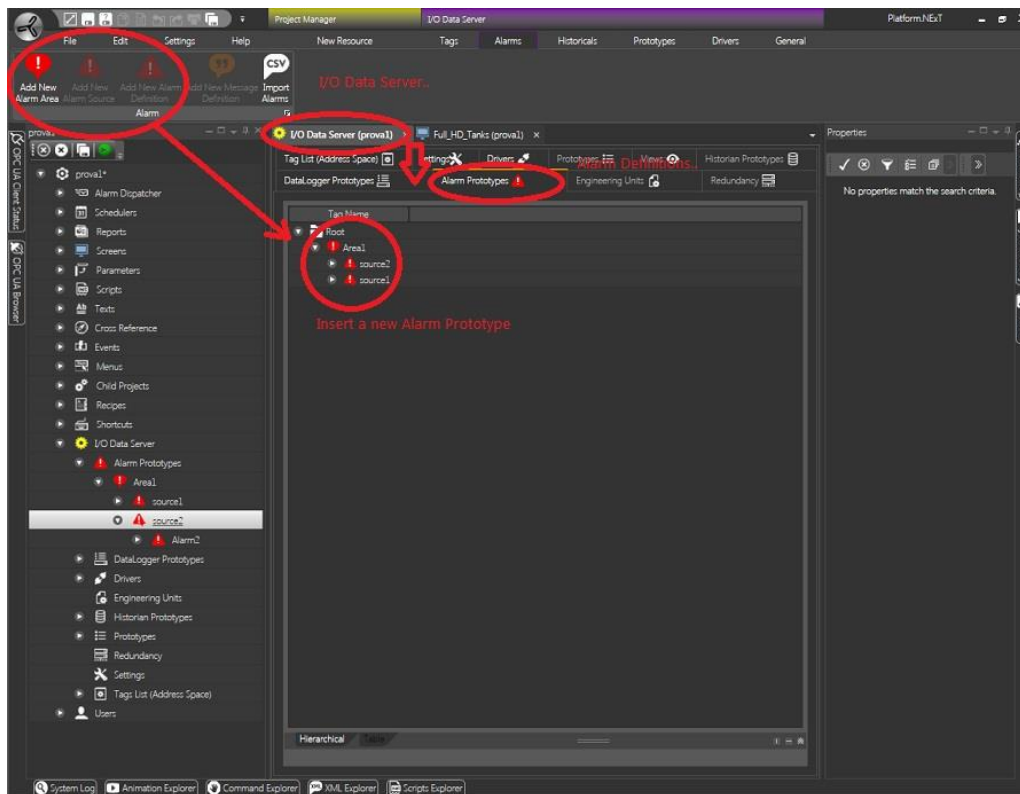


As you can see in the diagram, the Alarm structure provides the possibility to divided alarms in different **Areas**, which can be defined with different **Sources**, each one having its own Alarms.

The alarm is always understood as a "Prototype", and therefore can be associated with different tags. This is very handy for repetitive alarm configurations (the same alarm type, e.g. "Motor Overload", for 300 motors, each one with its own tag).

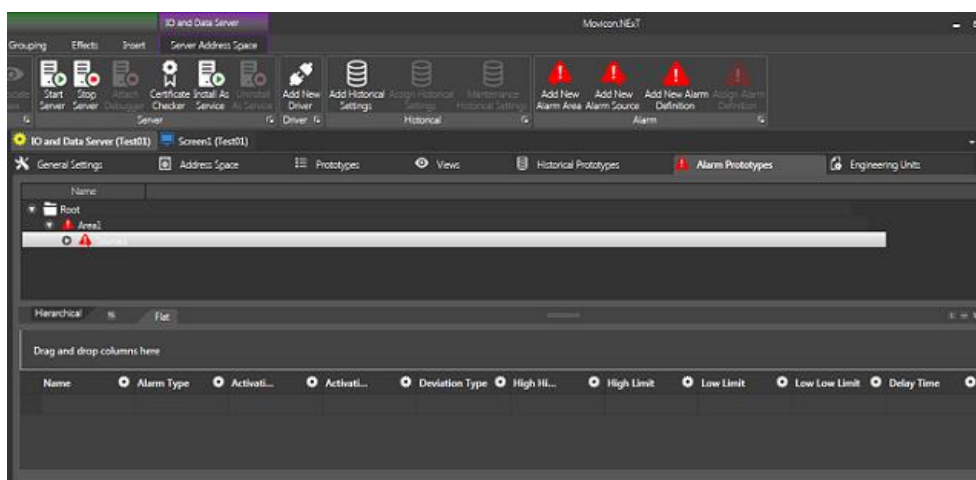
**We will now see how to proceed with inserting alarms in a Platform.NExT project:**

1. Open the "**Alarm Prototypes**" resource from the "I/O Data Server" group. Use the "**Alarm ->Add New Alarm Area**" command from the "I/O Data Server-Address Space" ribbon to add a new **Area** to belong to the alarm we are about to create. Assign Area with the desired name and confirm it with OK.



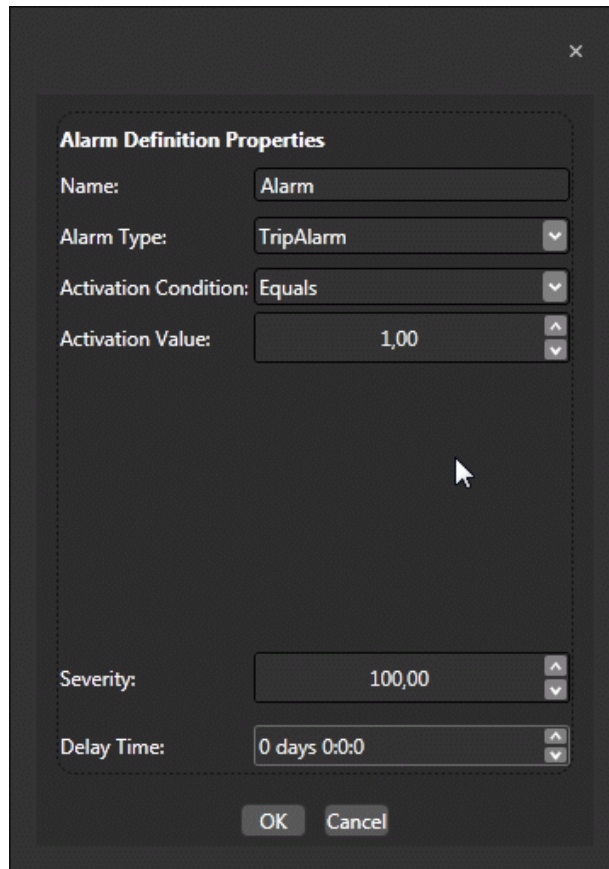
By using the "Alarm Prototypes" resource you can insert and edit project alarms.

- After having added the Area, you now will have to insert the "**Source**" of the Alarms we are about to define. To do this select the name of the Area you have just inserted and by using the "**Add New Alarm Source**" command from the "I/O Data Server-Address Space" Ribbon add a new Source in the Area. Confirm with the name desired and "OK" the window for adding the Alarm Source.
- At this point we can insert a "Alarm Threshold" referred to the Area and the Source. Select the name of the Source and add an Alarm Threshold in the Source using "**Add New Alarm Definition**" command from the "I/O Data Server-Server Address Space" Ribbon.



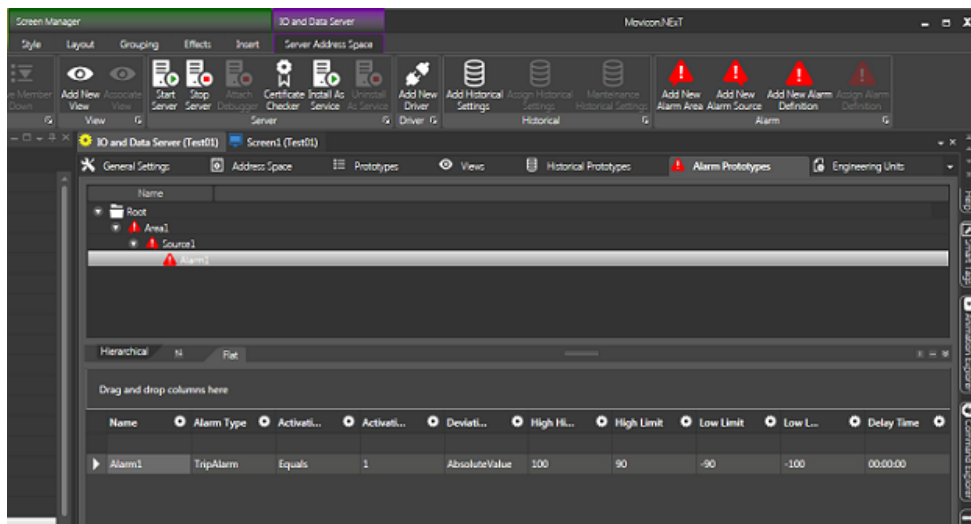
Project Alarm Thresholds can be inserted and edited using the "Alarm Prototypes" resource.

4. Inserting a new "Alarm Definition" will open the dialog window for setting the threshold properties. This window can also be opened by double clicking an already existing Alarm Threshold name on the list to enable you to modify its settings.



*An Alarm Threshold's Properties Window*

5. The default properties that are provided for creating a new Alarm Threshold propose a digital Alarm TripAlarm) with an intervention threshold with the "1" value. This means that when the tag associated to the Alarm assumes the "1" value, the Alarm will activate. Otherwise the Alarm will remain deactivated. However the parameters relating to the "**Alarm Type**", "**Activation Condition**" and "**Activation Value**" can be modified. In our example will leave the those proposed by default therefore a digital alarm with activation "equal" to the value "1".

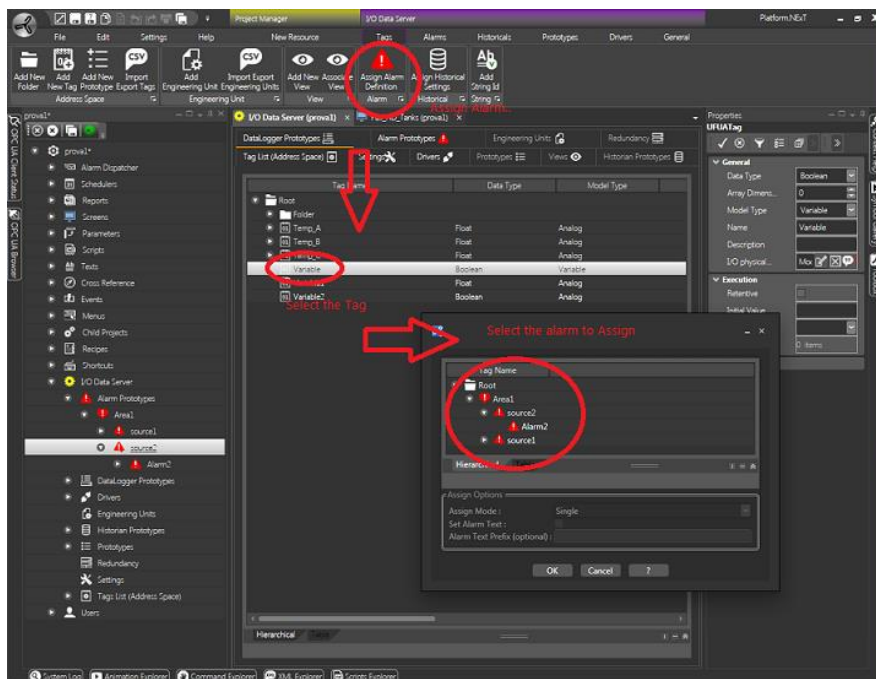


A tree structure display of an Alarm prototype

At this point after having defined and configured the Alarm Prototype object, we will need to assign the tag which will contain the dynamic reference value and therefore descriptive text.

### Assigning the Tag

1. Open the **"Address Space"** Resource from the "I/O Data Server" group so that the list of project tags displays in the workspace. If this list is empty, add the tags for managing the alarms.
2. Select the **Tag desired** from those presented, e.g. "VAR00001", and activate the **"Assign Alarm Definition"** command available from the "I/O Data Server-Address Space" Ribbon. This command will display the window from which the previously inserted alarm threshold (**Alarm Definition**), is selected. Select the alarm and confirm with OK.

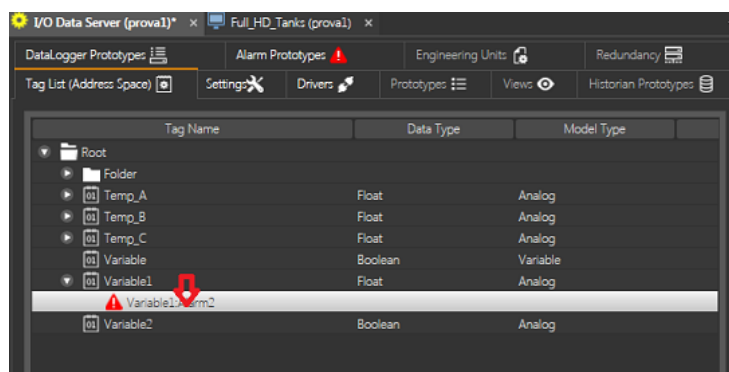


The "Assign Alarm Definition" command links the Tag to the Alarm Prototype.

- Confirm to associate tag to alarm upon which will then be displayed through an icon linked to the tag as illustrated in the image. Repeat these procedure to assign other tags to other alarms. When having repetitive alarms, such as 300 "Motor Overload" Alarms, simple create the one Alarm Prototype then multi-select all the tags desired and assign them altogether with one operation.



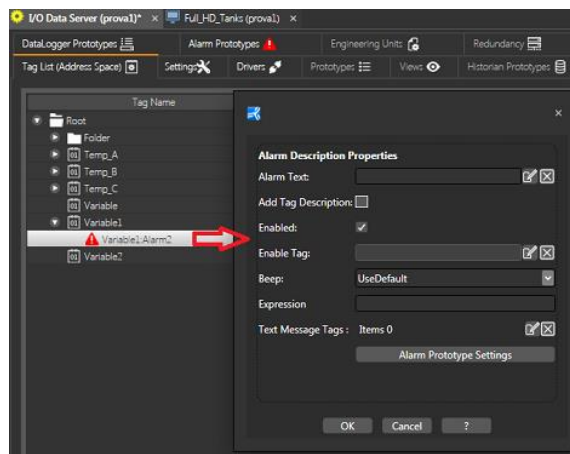
To delete a tag assigned to an alarm simply select the tag from the tag list and then select its assignment to the alarm and press the "CANC" keyboard key.



The Alarm name associated to a tag can be displayed by clicking the "+" at the side of the tag.

## Assigning Text

- To assign a text string to the alarm you will need to select the Alarm assigned to the tag desired, from the tag list. Double click to display the property window through which you will be able to define a text or select a Text String from the String Resource. This will also allow you to change the language of the alarm text as well.



Insert alarm text by double clicking the Alarm-Tag association object.

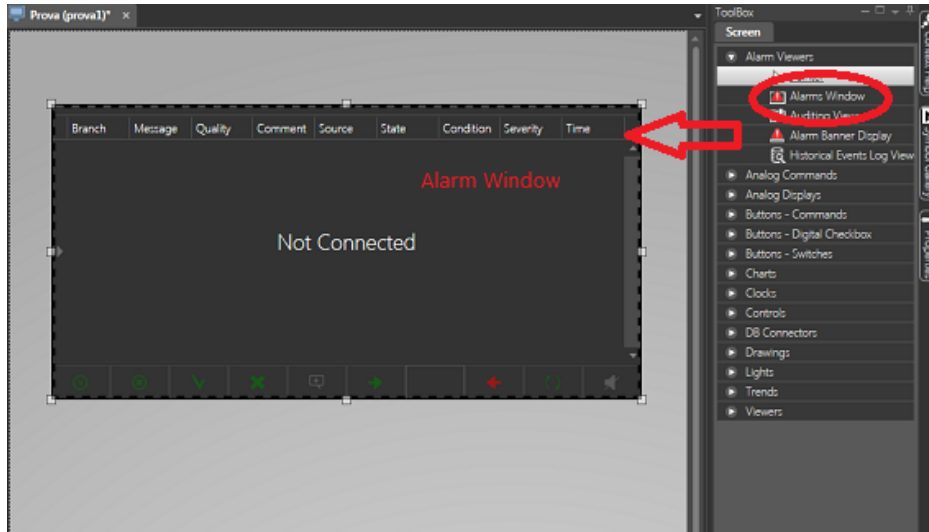
To check whether the alarm management is working correctly or not start project up in Runtime Starting UP Project in Runtime and change the value of the tag which has been associated to the Alarm (also see paragraph "Creating Alarm Simulation"). When the Tag's value is set at "1", the alarm will activate, otherwise the Alarm will deactivate. To see the list of active alarms you will need to use the appropriate screen objects as described in "Displaying Alarm".

## 6.2. Displaying Alarms

Active Alarms can be displayed on screen using the "**Alarm Viewer**" or "**Alarm Banner**" viewer objects during project Runtime.

To insert an alarm viewer object proceed as follows:

1. Open a screen within the Movicon workspace.
2. Activate the Toolbox and select the "**Alarm Viewers**" object group by clicking the icon to expand list below.
3. Select the object you wish to insert, for example the "**AlarmWindowControl**".
4. Drag and click to drop it on the Screen where desired.

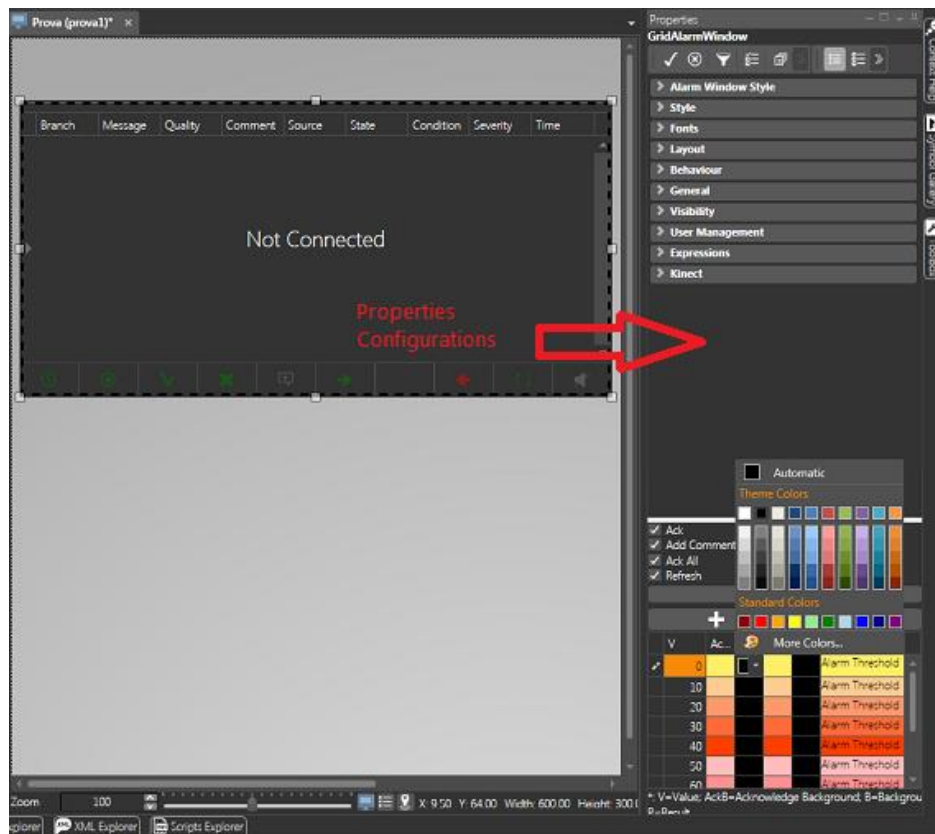


*The "Alarm Viewer" object is designed for viewing active Project alarms.*



The local machine's Server module automatically connects to the Alarm Viewer inserted on screen for default.

After having inserted the Alarm Viewer object, you can then size it as pleased and configure its style using the object's Properties window.



*The "Alarm Viewer" object can be configured as required through its Properties Window.*

At this point, start the project up in runtime Starting UP Project in Runtime and activated the Tags linked to the Alarms to show messages in the Alarm Viewer (also see the paragraph on "Creating Alarm Simulation").



The "Alarm Viewer" can be connected to a Server different from the one used by the Project in runtime, for example to retrieve alarms from another project's Server or third party projects with OPC UA over the net. To achieve this you will need to specify which reference Server the Alarm Viewer is to use. To do this proceed as follows:

1. Open the project's "I/O Data Server" resource in the Workspace.
2. Activate the "Start Server" command from the "I/O Data Server - Address Space" Ribbon.
3. Wait until the project's Server module starts up.
4. Once started the Server will show the Screen in the workspace's foreground.
5. Bring the "OPC UA Browser" window to the foreground if not already done so. and check to see if there are any projects with the same name on the list of OPC UA Servers.
6. Open the OPC UA Server tree with the same project name by clicking on the node's icon. Double click the endpoint you wish to use as transport, e.g. "net.pipe, None".
7. Select the "Server" folder from the OPC UA Server browse window which opens and drag it on to the Alarm Viewer.



## 6.3. Creating Alarm Simulations

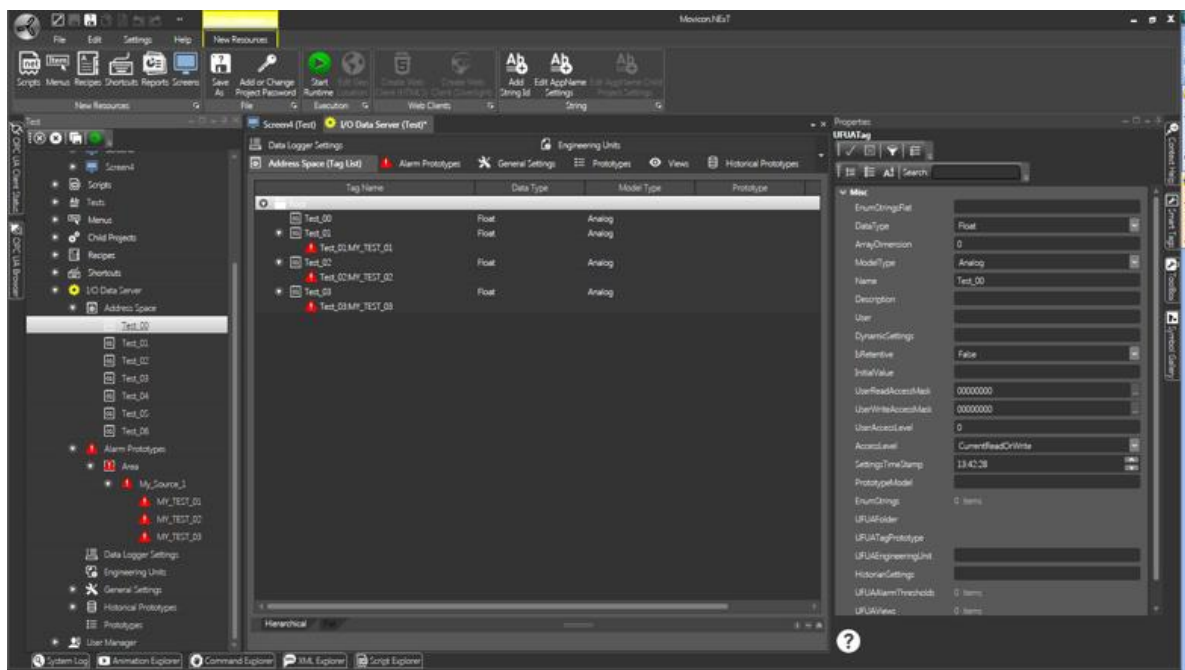
In this section we will create a simple project to represent an alarm simulation of an example project.

In this example we will use 3 tags and 3 alarms assigned to the tags.

To simulate activation in runtime, we will use three "selector" command objects associated to the tags.

Proceed as follows:

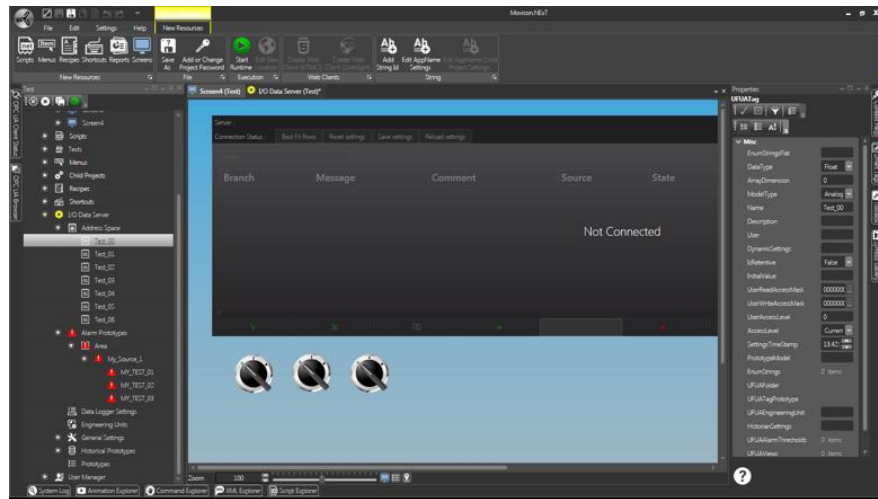
1. Open the Platform.NExT workspace
2. Access the 'I/O Data Server and create three new tags in the Address Space with the names of Test\_01, Test\_02 and Test\_03, leaving their properties set for default.
3. Remain in the 'I/O Data Server and access the Alarm Prototypes resource and crate a new Alarm prototype by creating the Area, Source and three Alarm Definitions called MY\_TEST\_01, MY\_TEST\_02 e MY\_TEST\_03. Leave properties set for default.
4. Return to the Address Space resource to assign the tags to the alarms you have just created. Select the Test\_01 tag and activate the "Assign Alarm Definition" command by selecting the MY\_TEST\_01 alarm. Repeat for the other tags.
5. We will not assign text strings for the time being but leave the alarm text currently displayed as the alarm's name.



*The workspace will look like this after tags and alarms have been created and assigned.*



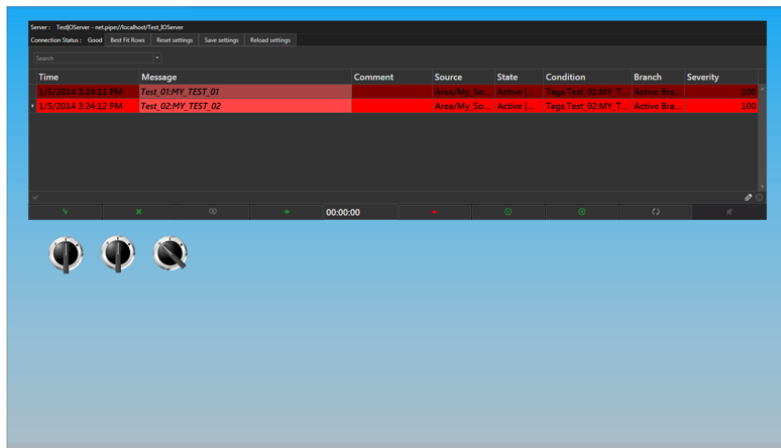
6. After having configured the alarms and Tags, we will continue with the graphical interface: we will create a new screen and insert alarm viewers from the Toolbox then size them as pleased.
7. Drag three graphical Switch button objects in the form of selectors from the Toolbox and drop them onto the screen underneath the Alarm Viewers.
8. Assign the tags to the selector switches simply using the Drag&Drop techniques (or using the object configuration window). For example, Select the "Test\_01" tag from the project's tree structure's Tag, and drag it to the first of the three selectors. Repeat this operation for the other two Tags.



*After having created the screen with the Alarm Viewers and tag command controls, the workspace will look like this.*

At this point we only need to start the project up in Runtime and test run the simulation:

1. Activate the **"Start Runtime"** command available from the "Project Manager - New Resources" ribbon. The project will startup in Runtime.
2. Open the alarm simulation screen we have just edited from the main window.
3. The screen will show with the Alarm Viewers and Tree Seletors. By manipulating the three selectors the tag should change status when passing to value "1" to activate the alarm. A message will appear in the Alarm Viewer reporting the alarm activation. You can then use the **Acknowledge** and **Reset** button provided in the Alarm Viewer.



Alarm simulation screen during runtime.

## 6.4. Event Historian Log

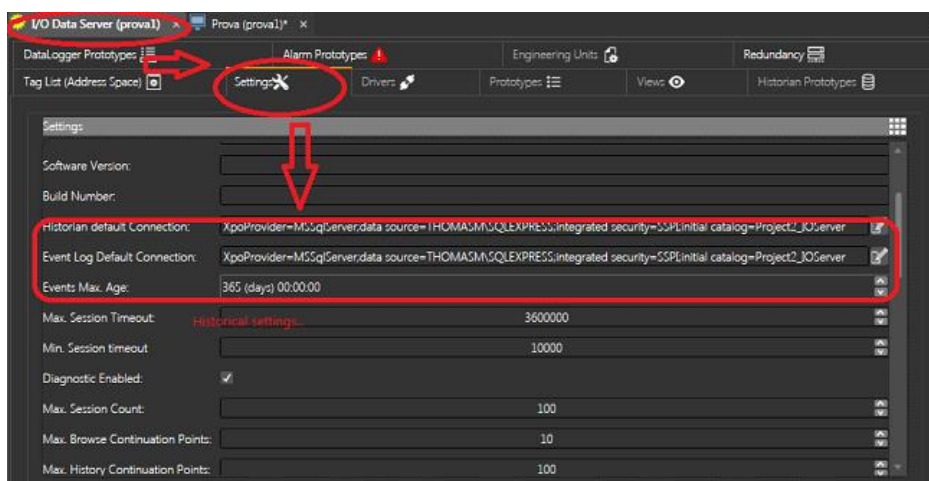
The Platform.NExT I/O Data Server's **Alarm Manager** module automatically records all the event on Historian archived definable on database files in the 'I/O Data Server's **General Settings**.

The events that the Alarm Manager module records in the archive files are:

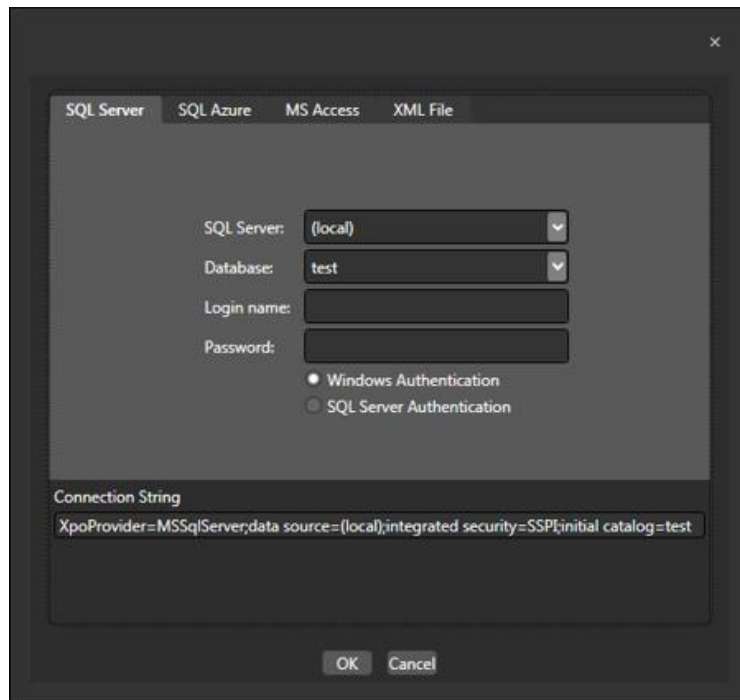
1. Alarms: All the alarms or messages defined as Project Alarms
2. Drivers: All the system events relating to the I/O communication drivers inserted in the project.
3. System: All the system events generated by the platform during project runtime

The data archive is defined using the connection string towards the databases, as indicated below:

1. Start Platform.NExT and open the project's **I/O Data Server** resource.
2. From the Data Server, access "**General Settings**" using the corresponding tab or project tree structure
3. Find the "**Event Default Connection**" property and open the settings widow using the [...] button on the right.



4. Set the Database and access parameter. For example, you can use the installed SQL Server version and keep Windows authentication. The database table can be selected from those already existing if you wish or you can enter the name of the one desired in the "Database" field which will be created by the system.



Confirm the operation to create the **Connection String** towards the Historian Logs archive database. In this way during project runtime, all the event will be recorded on the DB archive.



Please note that in addition to defining connection strings to the DB in the "Settings" and Data Base Settings" section, you can also define the maximum file size in days (365 days for default) in the "Events Max. Age:" property. Once this size has been reached, the system will recycle data by overwriting the oldest data.

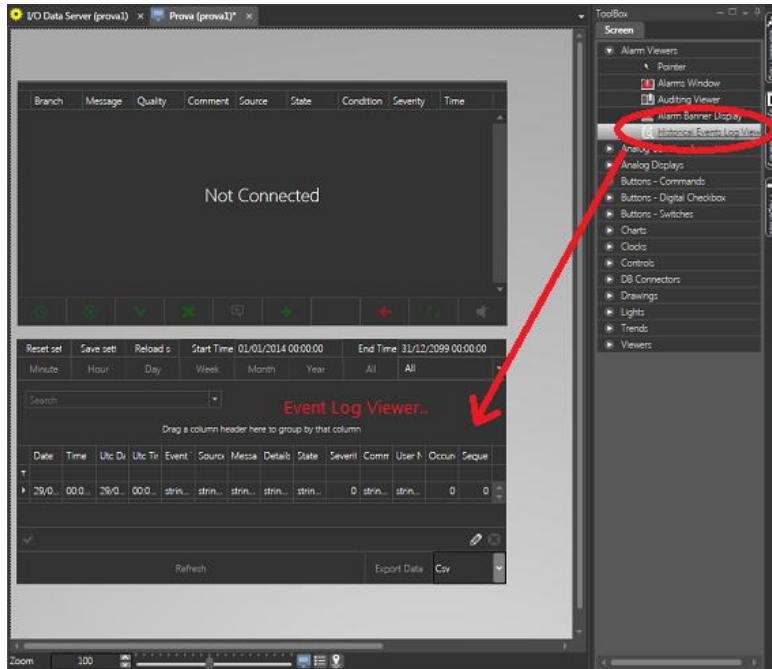
## 6.5. Historian Log Visualization

During the Project Runtime phase, you can view Alarms and Event recorded on Data Base by the Alarm Manager model during runtime. The Historian Log events can be viewed using the "Event Logger Viewer" object that can be taken from the Toolbox and inserted in Screens.

To insert a Event Log viewer object proceed as follows:

1. Open a Screen in the Movicon workspace.

2. Activate the Toolbox and select the "**Alarm Viewers**" object group click on the icon to expand group.
3. Select the "**Event Logger Viewer**" object.
4. Drag and drop the selected object to a position on screen. To simplify things, this viewer will be inserted in the same screen where we previously inserted the Alarm Viewer as indicated below:



*The Event Logger Viewer object displays the Alarms and event recorded during runtime on the project's DB.*

After having inserted the Event Logger Viewer object you can size and configure it style as pleased using its properties windows.



Historian Log viewer provides three tables as previously described for: Alarms, Events, Drivers and System Events. You can view all by allowing the user to apply filters in runtime or force data displays of one of the three tables, for example to view only the Alarm Log, using the appropriate property from the object's properties window.

At this point start the project in runtime Starting UP Project in Runtime and activate the tags linked to the Alarms to display the messages in the Alarm Window (also see "Creating Alarm Simulation"). As a consequence the system will start recording the event on DB to can be viewed using the inserted viewer object.



The Viewer's Refresh command forces access to the DB to refresh the Historian data viewed. In addition you can use the appropriate viewer object command for applying filters and order bys of logged data.

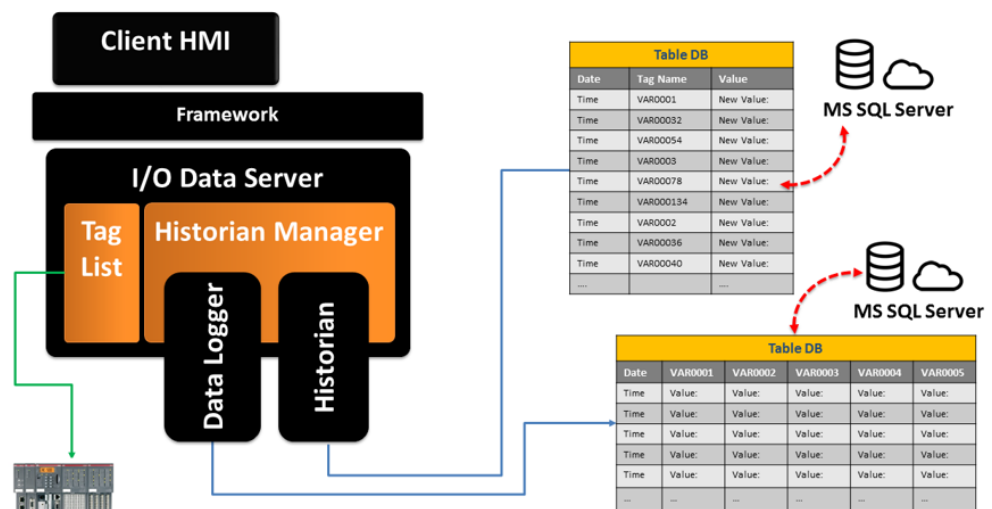
# 7. Historian Management

## 7.1. Historians Management

The I/O Data Server's Historian module records tags managed by the I/O Data Server and defined in the Address Space, using two different recording models:

- **Historians**
- **Data Loggers**

Both recording models record tag values on Database archives, but use different models and criteria for recording data.



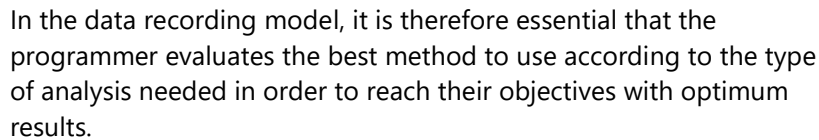
### 7.1.1. Historians

The Historian records data associated to Historian prototypes according to a model which provides DB tables containing records for each even recorded, in one column. This means that tables will have one column only independently from the amount of tags associated, whereby each record represents the changed value recorded only. The DB is very useful being efficient and ideal for analysing and displaying data in the viewer objects such as the Trends or Data Analysis. The Data Logger model is more suitable for the Report management.

### 7.1.2. Data Loggers

The Data Logger records data associated to the Data Logger prototypes according to the a model that provides data tables on DB, with records for each event recorded shown in one column for each tag associated.

This means that one table will be obtained with the same number of columns as tags associated, with the recordings of values of all the tags (also those that do not change) for each record. This occupies more DB memory and therefore less efficient, but data can be organized better to structure the those archives typically used by the Reports.



- Historian: Process Data
- Data Logger: Production Data

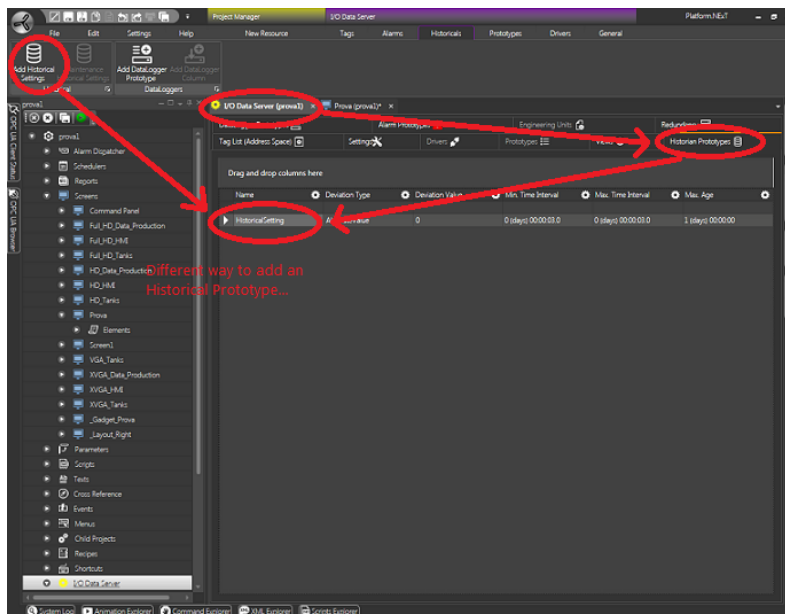
## 7.2. Inserting and Associating Historian Prototypes

Here we will see how to insert a data recording model in a Platform.NExT project. The techniques and the Tags used to do this are identical for both the Historian and Data Logger models.

To simplify things we will explain the techniques used for inserting a Historian Prototype considering that these procedures are the same used for inserting Data Logger prototypes as well.

Follow the step-by-step procedures below to insert a new Historian Prototype in the project:

1. Start up Platform.NExT and open the "I/O and Data Server" module's **"Historian Prototypes"** resource.
2. Use the **"Add Historian Settings"** from the "I/O Data Server-Address Space" Ribbon, to add a new Historian Prototype to the list.



*Project recording engines are inserted and edited using the "Historian Prototypes" resource*

- When inserting a new Historian Prototype the **dialog window used for setting its properties** will open. This window can also be opened by double clicking on a Historian Prototype that already exists on the list to modify its settings.



The fundamental concept of Historian Prototype recording parameters is the criteria with which the record on change or time interval frequency is defined taking into considering:

- Min. Interval
- Max Interval

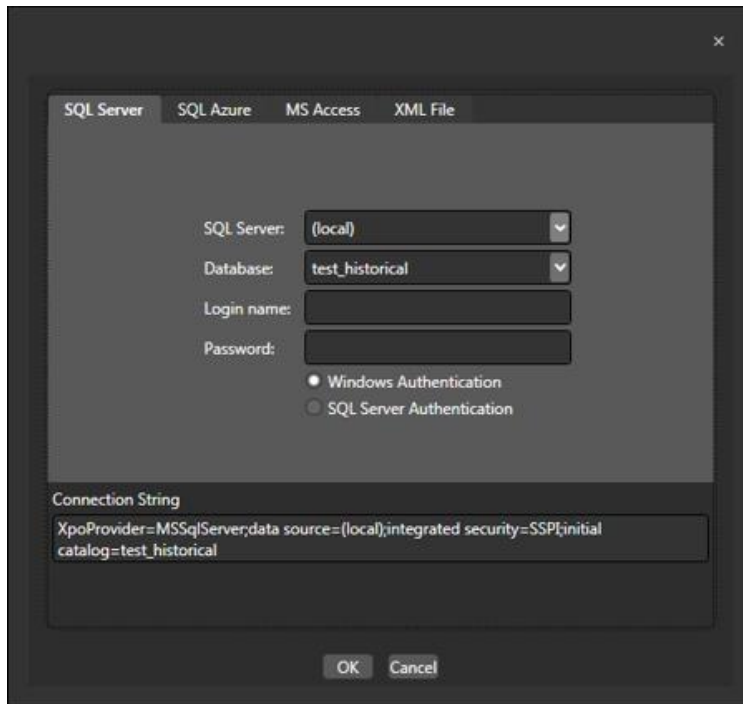
Where the Minimum and Maximum interval define the minimum frequency value when data changes are very frequent, and the maximum frequency value when data changes are not very frequent, hardly ever or never change.

Example:

1. Setting the Minimum and Maximum value to zero, as for default, the Historian will record the value upon each change.
  2. Setting a value different to zero (for example 5 sec.) for both the Min and Max parameters, the Historian will record each time value inserted (eg. 5 sec.) regardless whether data changes or not.
  3. Setting a Min value (eg. 1 sec.) and a Max value (eg. 5 sec.), the Historian will record at least every second (Min value) if data changes are very frequent. If the data does not change by the time the Max value is reached (eg. 5 secs) a recording will be executed nevertheless.
4. We will now continue with assigning some parameters to the Historian Prototype. Suppose we need to record at time intervals of every 5 seconds and keep data 30 days. To achieve this we will need to set the "Min Interval" and "Max Interval" parameters with the 5 value and the "Max Age" parameter to the 30 day value.

*Historian Prototype parameter settings window*

5. In order to record data in archives the Historian Prototype must use a **Connection to the DataBase**. This connection can be configured by inserting the connection string in the "Connection String" parameter or by using the browse button to configure it.
6. The Database connection configuration window enables you to select the database type you wish to connect to. In our case we will select the "SQL Server" and define the connection parameters with SQL Server name, DataBase name and Authentication type.



*DataBase Connection configuration window.*

7. Confirm the window with "OK" upon which the connection string will be composed automatically with the parameters defined and inserted in the "Connection String" field.



If the "Connection String" field is left empty, the Historian Prototype will use the default connection set in the "**Historian Default Connection**" property from the I/O and Data Server's **General Settings** resource.

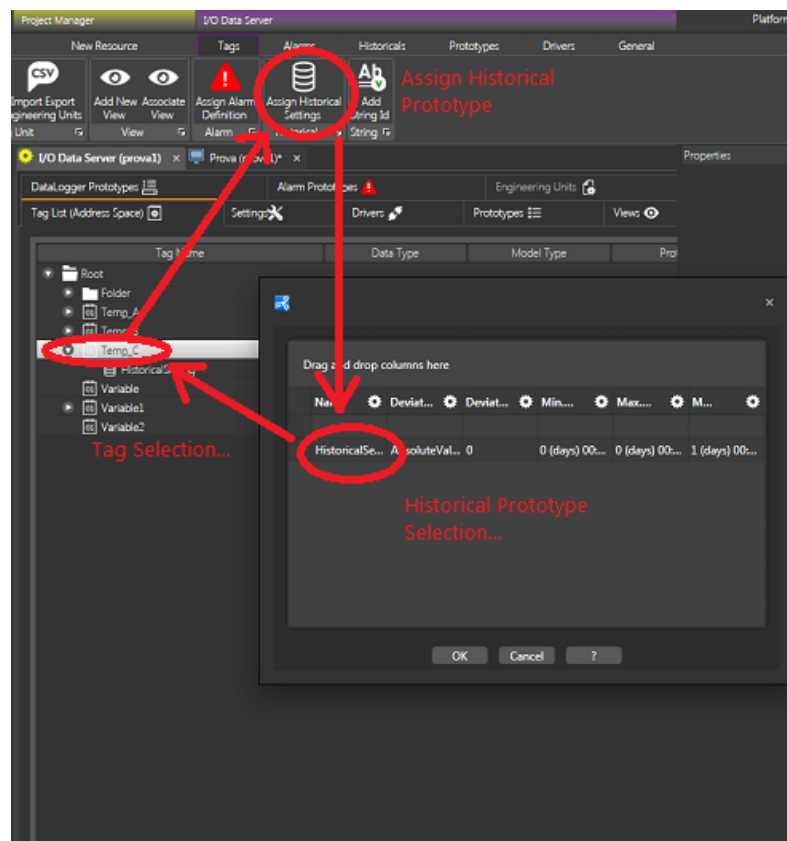
### Assigning Tags to the Historian

At this point, after having defined and configured the Historian Prototype object, we will need to assign the Tags whose values will be recorded on DB during Runtime. This is done as follows:

1. Open the "**Address Space**" resource from the "I/O and Data Server" group so that the list of the Project Tags are shown in the workspace area. If this list is empty add the tags to be recorded with the Historian Prototype.
2. **Select the Tag or Tags desired** (multiple tag selection is also possible) to connect to the Historian Prototype, for example "Test\_01", "Test\_02" etc.



- Execute the **"Assign Historian Settings"** command available in the "I/O Data Server - Address Space" Ribbon.



*The "Assign Historian Settings" command connects the Tag to the Historian Prototype.*

- Upon confirming this operation the Tag will be associated to the Historian Prototype, and it will be displayed using the icon linked to the tag as shown in the image above. These operations can be repeated for assigning other Tags to other Historians. When dealing with repetitive selections, such as 300 tags of values to be recorded with the same Historian Prototype, simply create just one Historian Prototype, then multi-select all the Tags desired and assign them all at one go.



To delete a Tag assignment to the Historian Prototype just select the Tag's assignment and press the keyboard "CANC" key.

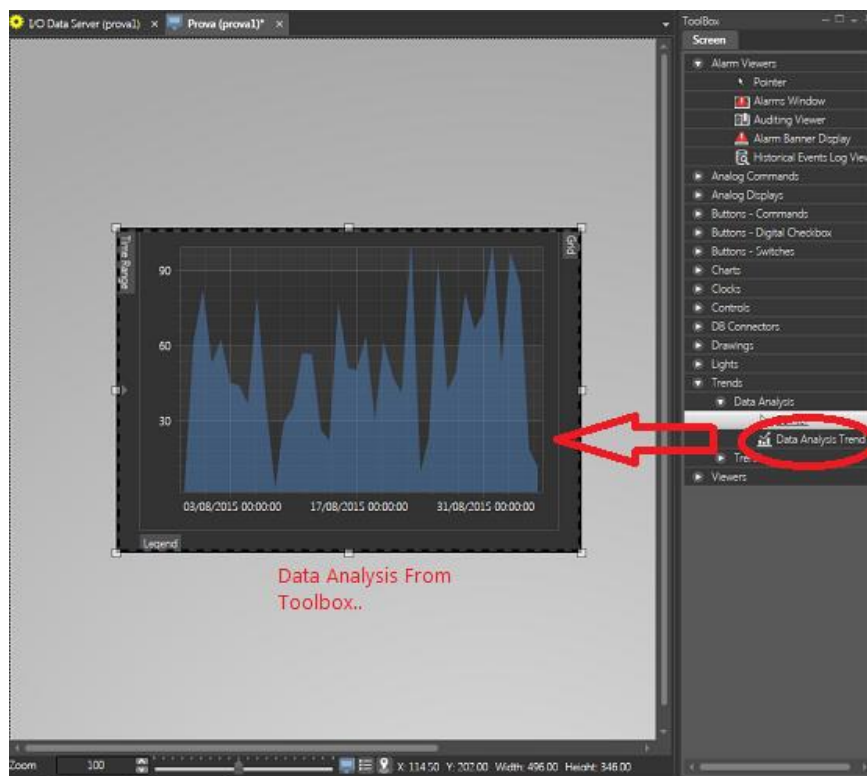
To check whether the Historian Prototype records correctly start the Project in Starting UP Project in Runtime and modify the tag values if need be. The "UFUAAuditDataItem" table will be created in the previously set DataBase in which the Historian Prototype records will be inserted.

## 7.3. Displaying Historian with Data Analysis

Platform.NExT has a large number of tools for displaying Historian data recorded by the I/O Data Server's **Historian Module**. Historian data is displayed using the **Movicon.NExT** graphical interface engine, being Client visualization by Server module. By using the Movicon.NExT Toolbox you can implement and put into use the various viewer tools such as Trends, Data Analysis, Grids and Reports.

To simplify things in this example we will use the **Data Analysis** which is particularly effective for representing trend curve graphics of values recorded on the database by the Historian Prototype described earlier on in this tutorial.

Proceed with inserting the Data Analysis viewer object by dragging or dropping it from the **Toolbox's** Trend group onto any screen of interest.



*Historian Data recorded by the Historian module can be displayed by using a viewer object such as the Movicon.NExT Data Analysis.*

Simply insert on screen and configure the object as pleased through its **Properties Window**. You will need to specify the object's **Connection String** in order to connect to the data archive created by the Historian module during runtime.



